ECE 445 Senior Design Lab Design Document: Smart Curtains

Team 41: Jack Davenport (johndd2), Vinay Konda (konda3), Max Mauschbaugh (maxjm2)

March 25, 2023

1 Introduction

This section will set up the problem we are trying to tackle and the general solution we came up with.

1.1 Problem

Alarm clocks are a shocking way to wake up in the morning, and often times can be ineffective when used on their own for extra early mornings.

1.2 Solution

We want to make LED-assisted smart curtains that automatically open in the morning to complement a user's alarm set on their smart phone. The curtains will open at a certain time in the morning based on user preference, should be able to sync up with a smart phone's alarm by connecting via WiFi and Apple HomeKit, and will activate LEDs that are installed around the user's window to simulate sunlight in case it is not bright enough outside.

1.3 Visual Aid

A visual depiction of our smart curtain system can be found below. The black button in the top right would be used by the user to interface with the processor. Depending on the input this button could assist the user with setting up the WiFi connection, toggle modes of the device to control if it should react to the connected phone's alarm or not, perform a hard restart on the device, or much more.

The blue light on our system represents the photodetector used to read the brightness in the room. This sensor reading will control the LEDs represented by the Christmas lights wrapped around the window.

The green button would be used for the user to open or close the curtains without requiring a connection to WiFi, or without the user's alarm having to be set.

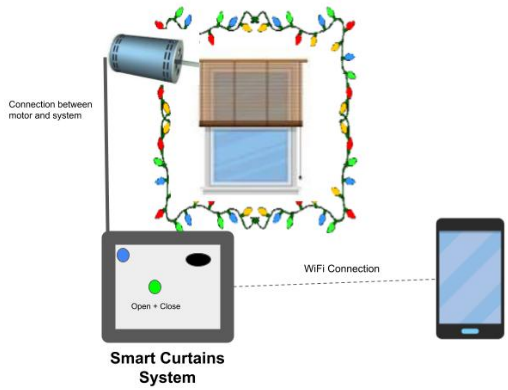


Figure 1: Visual representation of Smart Curtains.

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 project will modify the design of typical two string roller curtains to make them use a motor instead. Instead of pulling one string to raise the curtains and one string to lower the curtains, our motor will turn the gear the strings are attached to, causing the curtains to raise and lower (see figure 2). 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 Curtains (fig 2). We will use a plastic enclosure to house the majority of our electronic components. Furthermore, there will be a photo-diode placed on the enclosure facing the inside of the room to sense light levels (fig 2). Additionally, inside of the enclosure we will house the motor and the PCB will be placed next to it (fig 3).

2.2 Block Diagram

The following is the block diagram of our solution. It contains an LED module, a Control module, a Blind Movement module, a Sensing module, a Power Module, and a User Interface module.

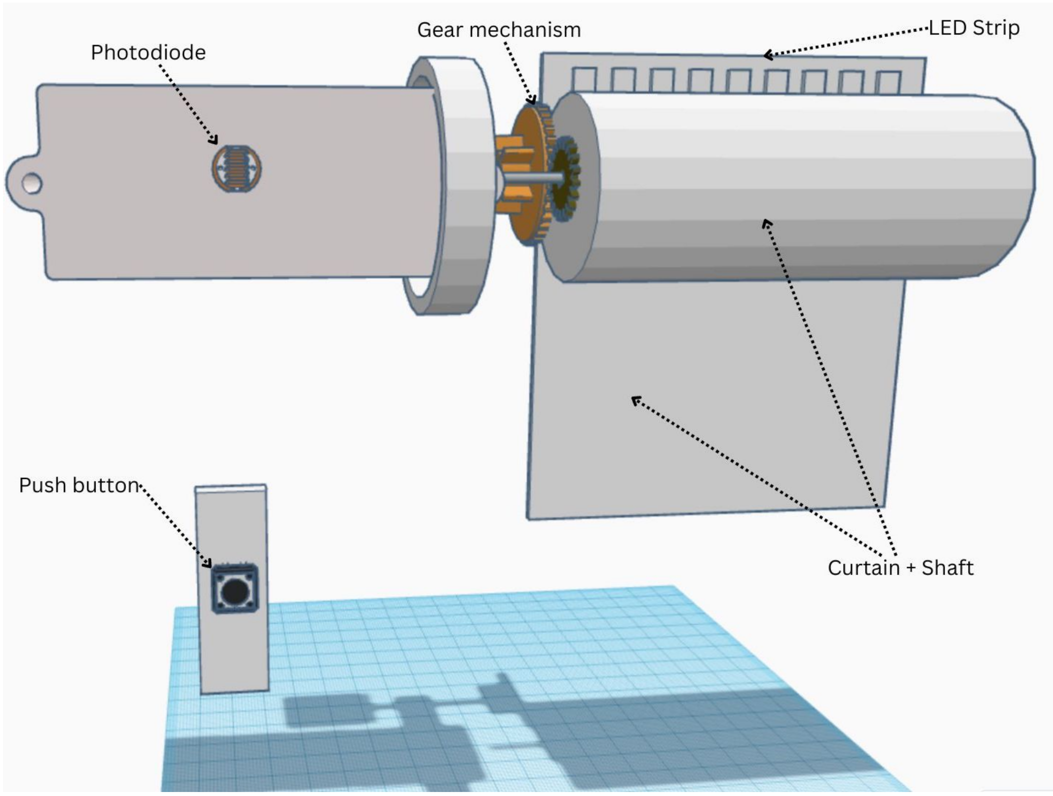


Figure 2: Front view of CAD representation of physical design.

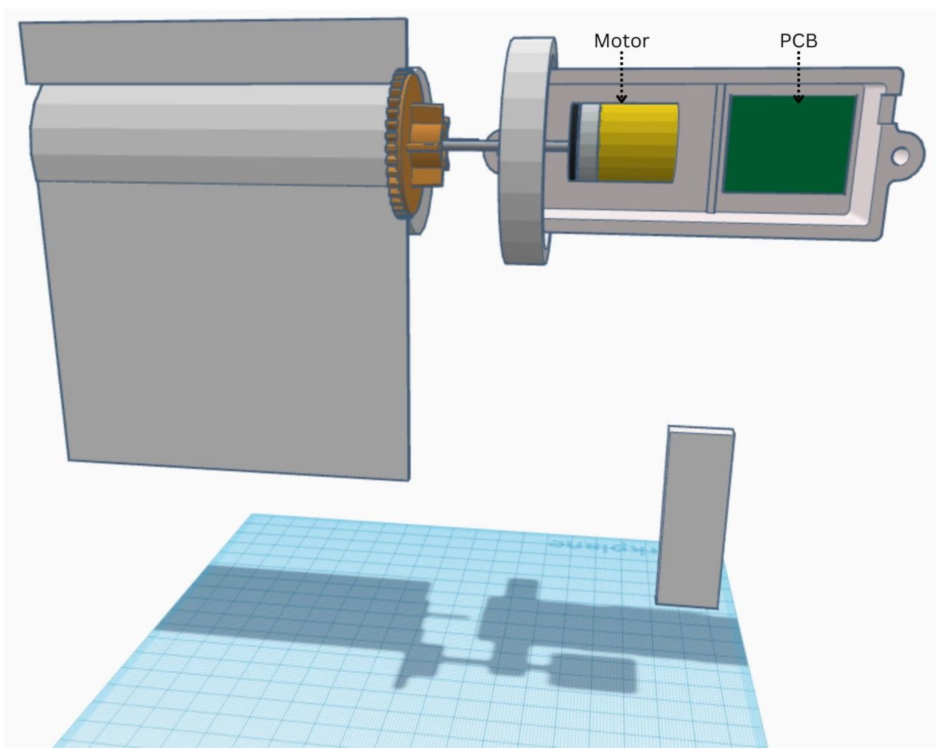


Figure 3: Back view of CAD representation of physical design.

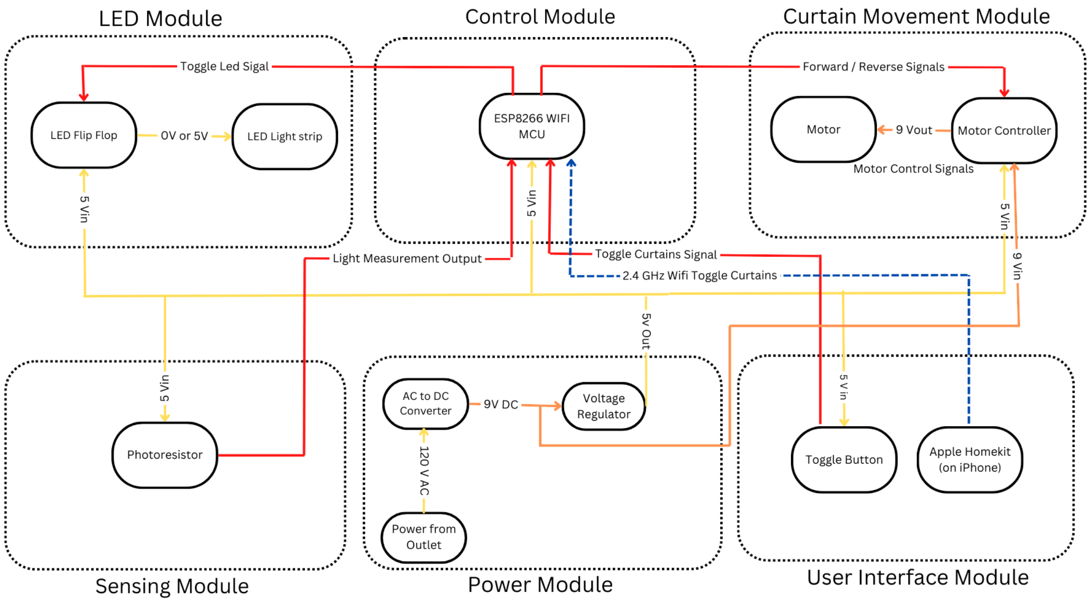


Figure 4: Block diagram representation of Smart Curtains.

2.3 Subsystem Overview

2.3.1 Curtain Movement Subsystem

For the movement of the curtain itself, we will connect a motor to a modified roller curtains mechanism to raise and lower the curtain. The curtains will be modified so that the gear that controls the turning of the curtain rod will be rotated by our motor. A gear will be attached to the motor's output shaft, which rotates as the motor runs, transmitting rotational force to the gear on the curtains. The gear on the motor shaft will be designed and 3D printed with teeth that engage the teeth on a curtains gear, which in turn drives the rod connected to the curtains. This results in the curtains being raised and lowered by the bidirectional rotation of our motor.

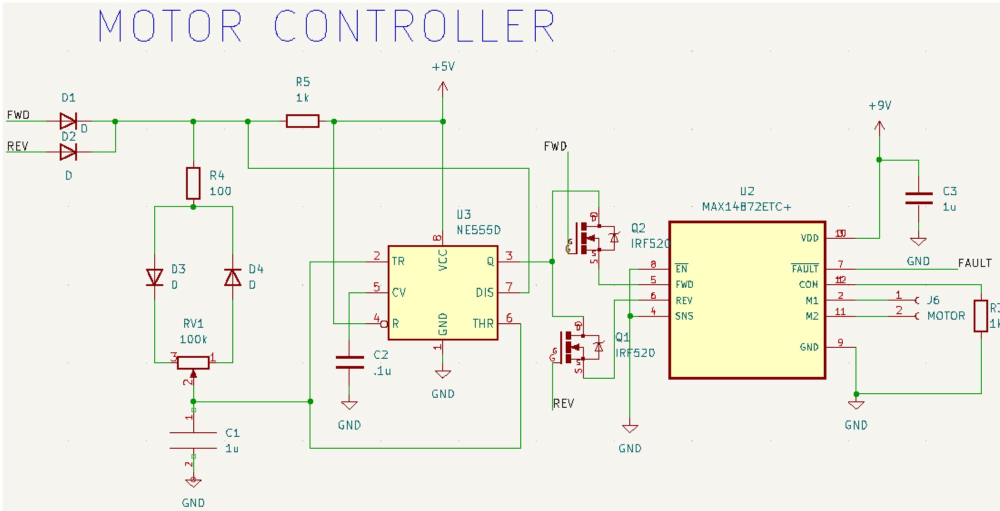


Figure 5: Circuit Diagram of Motor Controller.

Requirements	Verification
• The motor should open and close the curtain in $\sim$ 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 all 72” of the curtain 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 (72”) be rolled/unrolled within $\pm$0.5 inch of its full 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 $\pm$0.5 inch of its full length when unrolled and in the other case check if there is $\pm$0.5 inch of curtain remaining when it is rolled. • Perform 15 iterations of this procedure for robust testing.

2.3.2 Brightness Monitoring Subsystem

In the case that the curtains open up and there isn't a sufficient amount of light entering through the window, we will utilize a photoresistor to measure the ambient light intensity around our device. The intensity reading will be used by the microcontroller to determine whether the room needs additional light by turning on the LEDs. For applying additional light to the room, we will mount an LED strip onto a curtain rod and turn them on only if there isn't enough light entering the room through the window.

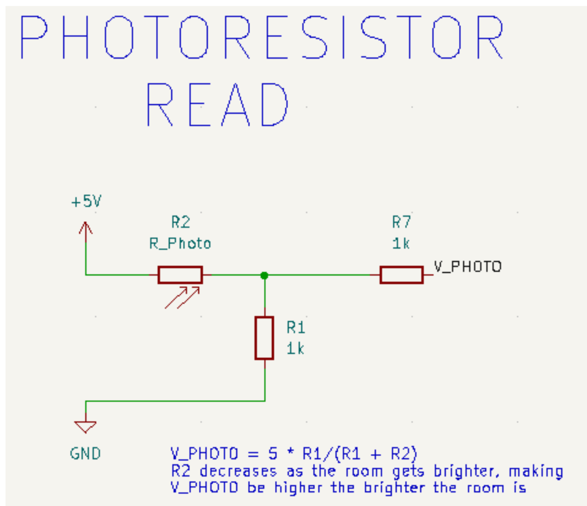


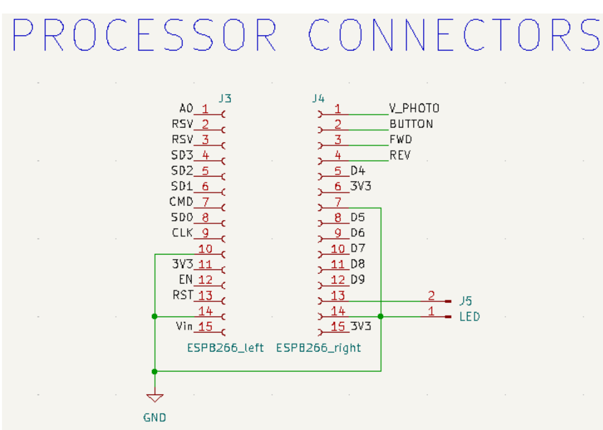
Figure 6: Circuit Diagram of Photoresistor Read.

Requirements	Verification
• The light sensors must be able to read the light intensity in the room and detect when the light threshold is crossed.	• 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 • Verify that LEDs turn on only with very dim light/ no light. • Repeat procedure 10 times for robustness

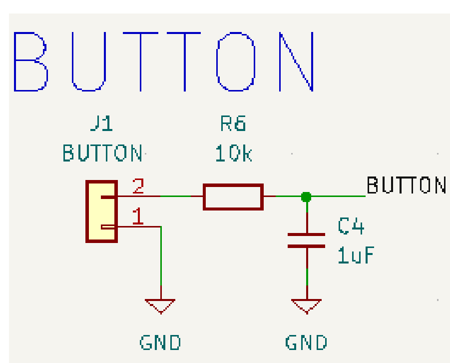
• The LEDs must possess enough healthy lighting to make up for darkness inside the room. This light level should be greater than 650 lumens.	• Connect the device to power and make sure LED components and light sensor components are mounted to the window frame. • Make sure photoresistors indicate darkness 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.

2.3.3 Processing Subsystem

Our microcontroller is connected to our WiFi module (ESP8266) and will receive open and close signals over the WiFi using Apple HomeKit, which the microcontroller will then process. The microcontroller will then send signals to the motor controller to open and close the Curtains. The microcontroller also will receive the measurement of the lighting in the room from the photoresistor, and will potentially send signals to the LED strip to activate it if there is not enough light. In addition, we will have a physical button for manually changing the microcontrollers open and close state of the curtains.



(a) Processor Circuit Schematic



(b) Button Circuit Schematic

Figure 7: Processing Subsystem Components

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 $\sim$ 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 $\sim$ 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.

2.3.4 Power Subsystem

In order to power the Microprocessor, motor, and other components of the system we will draw 120 V AC from the wall, which will be converted to 9 V DC using an AC to DC converter. We will then use a LM2596S-5 to step down the voltage to 5 V DC, the needed voltage for most of our components.

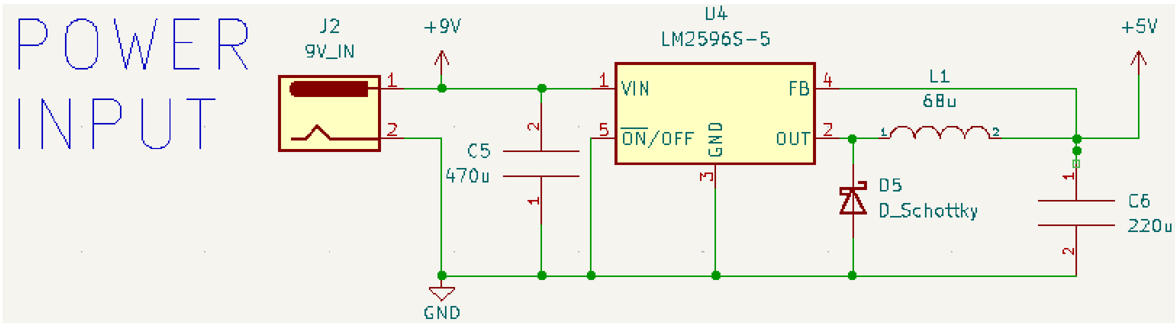


Figure 8: Circuit Diagram of Power Input.

Requirements	Verification
• The system should have protection against overvoltage, overcurrent, undervoltage, 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, overcurrent, 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 ±0.1V and at 5V ±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 ±0.1V and 5V ±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 ±0.1V and 9V ±0.1V.

2.4 Hardware Design Decisions

For our hardware design we had to consider how we wanted to power our circuit, what system we would use to sense the ambient light of the room, how we would control our motor, and how we would communicate over WiFi. When deciding our power subsystem, we debated between using a LiPo battery and using a wall outlet to power our system. We decided on using the wall outlet to simplify our design and reduce the weight of our system. For our light sensing subsystem, we decided to create our own circuit using a photoresistor for simplicity of integration into the rest of our design. When deciding how we would control our motor, we decided to create our own circuit for a motor controller so we could integrate it directly onto our custom pcb for our project. When deciding how we would communicate over WiFi, we chose the ESP8266 WiFi module because it can be used for WiFi communication and as the microcontroller for our system. When making our hardware decisions, we made sure all our components would operate using the same voltage for simplicity of design (except the motor). Lastly, we made sure that we would have enough I/O pins on our microcontroller to be able to receive and send signals to all parts of our system.

2.5 Software Design Decisions

For our design, we are using the ESP8266 chip for our microprocessor and our WiFi connection. We chose this chip because of its compatibility with Apple HomeKit, Apple's home automation software. We will use open sourced Apple HomeKit packages to implement the home automation aspect of our project, and then write the rest of our motor and sensor code in C. The Apple HomeKit code will take care of the user customization of when to open and close the curtains. The additional code we write will control the motor, LED, and sensor inputs, while integrating this with the Apple HomeKit functionality.

2.6 Tolerance Analysis

The part of our system that we are most worried about overheating or causing issues regarding tolerance is our motor control system due to the amount of current required to drive our motor. To fix this issue we had to choose a motor driver that provides extra safety and was designed to support our required current while minimizing heat generation. We chose to use the MAX14872 IC since it can support a wide voltage range from 4.5 - 36V, handle up to 2.5A peak current, as well as provide a variety of other benefits discussed below [2].

The MAX14872 device offers safety features built into the integrated circuit such as overcurrent condition checking, current regulation using a sense resistor, and thermal shutdown [2]. We may monitor these failure cases by observing a FAULT signal generated by the device. These features ensure that our system functions within safe operating points, and disables our motors in the case of a short or improper circuit design to avoid unnecessary damage. We can choose our desired maximum current output by increasing or decreasing the value of a sense resistor connected between the COM and GND pins. According to the documentation [2] for the MAX14872, we can set our maximum motor current using the equation:

$$I_{MAX} = 0.1V/R_{SENSE} \tag{1}$$

The documentation [2] also recommends choosing our sense resistor to have a value such that the voltage at COM relative to ground due to the current flowing through the sense resistor stays within ±250mV. This means we must choose a resistance such that:

$$|I_{SENSE} * R_{SENSE}| < 250mV \tag{2}$$

To control the current supplied to the motor we will use an external pulse-width modulation circuit. This controlled signal will be supplied into the FWD or REV pins of the MAX14872 device. We will experimentally decide what motor current will be ideal for our implementation, and with this current we will directly calculate the corresponding ideal sense resistance.

The schematic for our motor controller implementation is shown in the image below. The MAX14872 IC can be seen on the right side of the circuit. The left side of the circuit is our implementation of a PWM regulator using a precision timer IC, specifically Texas Instrument's NE555D [4], which closely resembles the circuit shown on a website [3]. This can be controlled using the potentiometer to increase or decrease the duty cycle of the signal the MAX14872 receives, thus increasing or decreasing output motor current.

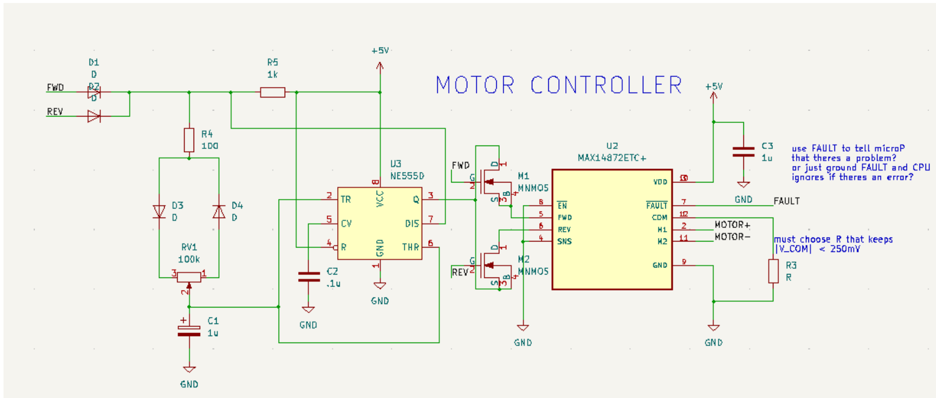


Figure 9: Circuit Schematic of the Motor Controller Labeled for Tolerance Analysis.

2.7 Cost Analysis

Here we will analyze the cost of parts and labor.

2.7.1 Labor Costs

The average ECE grad makes around 40 dollars per hour and we each plan to work on our project for around 10 hours per week for 7 weeks. Resulting in 210 total hours worked at a rate of 40 dollars per hour, resulting in a total labor cost of $ 8,400.

2.7.2 Part Costs

Part	Total Cost	Quantity
Curtains	$14.08	1
ESP8266-12E WiFi Enabled MicroController	$1.98	1
Motor	$1.20	1
LEDs	$12.99	1
Button	$8.41	1
NE55D Timer	$0.91	1
100 ohm Resistor	$0.10	1
1k ohm Resistor	$0.40	4
10k ohm Resistor	$0.59	1
100k ohm Resistor	$4.43	1
Photoresistor	$1.30	1
0.1 mu F Capacitor	$ 0.10	1

1 mu F Capacitor	$ 0.30	3
220 mu F Capacitor	$ 0.90	1
470 mu F Capacitor	$ 0.47	1
Diode	$ 1.88	4
Schottkey Diode	$ 0.27	1
2-Pin Connector	$ 0.36	2
15-Pin Connector	$ 1.48	2
Male 9mm Barrel Connector	$ 6.68	1
Female 9mm Barrel Chord	$ 5.72	1
68 mu H Inductor	$ 1.23	1
NMOS Transistor	$ 2.26	2
Total	$ 66.55	

2.7.3 Grand Total Cost

With our parts totaling to $66.55 and our labor costing $8400, our grand total cost for our project is $8466.55.

2.8 Schedule

Week	Everyone	Jack	Vinay	Max
2/20 - 2/27	https://www.overleaf.com/project/63f7a170ad5d269c5b18319 Prototype with breadboard and order parts	Start pcb board design	Plan physical design of window shade	Test apple home-kit and wifi chip communication
2/27 - 3/6	Prepare 3D prints	Prepare for first round PCB order	Complete initial push button software	Complete initial motor control software
3/6-3/13	Begin window shade assembly	Solder and Test out first PCB and test out power system	Complete initial photoresistor software	Figure out calibration of motors with PCB
3/13-3/20	Spring Break	Spring Break	Spring Break	Spring Break
3/20 - 3/27	Build push button wall panel	Finish photoresistor unit assembly and motor unit assembly	Figure out calibration of photoresistors with PCB	Complete initial LED software

3/27 - 4/3	Build enclosure for electronics	Complete second round of PCB order	Finalize photoresistor software with working photoresistor unit	Finalize motor software with a working motor unit.
4/3 - 4/10	Attach all physical components and mount to window/wall	Solder and test second PCB order.	Test photoresistor/LED in various light environments	Optimize motor timing/speed with curtain mounted
4/10 - 4/17	Debugging	Debugging	Debugging	Debugging
4/17 - 4/24	Mock Demo	Mock Demo	Mock Demo	Mock Demo
4/24 - 5/5	Final Demo + Final Presentation	Final Demo + Final Presentation	Final Demo + Final Presentation	Final Demo + Final Presentation

3 Ethics & Safety

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