Design of Automated Window Blinds

Matthew Azer
Yousif Fatohi
Entela Gjoni
Peter Younan

<u>Amp It Up</u>

Table of Contents

Executive Summary

Automated Window Blinds fulfill the need for light, atmosphere, and privacy under the correct conditions without human intervention. Designing window blinds to tilt, open, or close under specified parameters such as lux level, temperature, or manual intervention via wireless remote, allows hands-free operation to the consumers and enhances the environment depending on current conditions.

The development focuses on producing an efficient and responsive device that combines sensors, a microcontroller, and wireless connectivity in order to achieve seamless automation. The project aims to design an affordable, simplified system that will be suitable for application in homes as well as offices. By addressing the need for automated control, the proposed design improves indoor environments while promoting energy conservation and ease of use.

Statement of Problem

No longer a luxury or high standard, automation has become the key component towards productivity and efficiency. Automation’s most crucial duty is decreasing manual tasks as much as possible, this way taking care of a high percentage of human error. With a push of a button, this strategy will make sure that exhausted homeowners will not have to worry about getting up to shut blinds off after a stressful day, or having to get up to fix them while binge Netflix.

What's more, alarm clocks are the biggest pain in the morning. Imagine waking up not to a blaring sound but to the gentle rise of natural sunlight as your automated blinds gradually open at your preferred time. This operation offers not only comfort, but also pleasant waking experiences.

The system integrates light sensors to monitor environmental changes and automatically adjust the blinds for optimal lighting and energy efficiency. A very important addition is the idea of adjusting the blinds manually using a wireless remote control, allowing the owner to have the freedom and control over the settings. A servo motor or stepper motor (or two), which will be Arduino-controlled, will actuate the blinds to move in a precise and smooth way. This system has the potential to be satisfactory for both residential and commercial needs.

Another essential benefit covered by the automated blinds is safety. Timing the opening and closing of the blinds, or linking them with a smart alarm, may help to increase the security of the house. For example, the blinds may close automatically when the house is armed to create an effect called "mockupancy". This gives the illusion of someone being home by simulating regular activity, such as lights and blinds opening or closing, thus discouraging potential intruders (*House Mate, 2023*). These and more conveniences are guaranteed by this user-friendly system integrated into the blinds.

Objectives

This document proposes the design and implementation of an “Automated Window Blinds” system with the following objectives:

(1) The use of the Arduino board’s microcontroller to get raw data from a light intensity sensor, while adjusting the tilting and open or closed state of the blinds using two stepper motors (NEMA-17).

(2) The implementation of an IR (infrared) remote control system that helps manually override automatic commands, while being distance-friendly.

(3) Able to be integrated seamlessly with standard horizontal blinds by connecting to existing tilt rods and lift cords without requiring full replacement.

(4) Calculate total torque outputs required for the tilt & lift mechanism to ensure safe & reliable operation.

(5) Ensure the chosen motor can exert enough torque for the system to tilt and lift without breaking down. Using a pulley system to help achieve the desirable output.

Technical Approach

A VEML7700 light sensor will be used to measure ambient light with high precision and good reliability. The sensor detects movement, tracking the motion and position of the sun, offering a 0~120k Lux measuring range. It is compatible with Arduino boards since it runs on 3.3- 5V. The device will control the movement of the shutters, using the two NEMA 17 stepper motors, depending on the light intensity. One motor will be needed for the up/down movement of the blinds, e.g., if the light is too strong, it lowers the blinds; if the room is too dark, it raises the blinds. And the other motor will work on the tilting of the slat, e.g., if the light is too bright, the system will lower the blinds to reduce glare, if the light is too dim, the system will raise the blinds to allow more daylight in and if partial adjustment is needed, only the tilt motor will activate to optimize lighting without fully closing the blinds [*Arduino SensorKit, 2025*].

A very easy and handy way of controlling an Arduino is the use of the IR remote. Besides being cheap and readily available, this piece is not complicated at all. You likely have multiple IR remotes lying around, some from devices you no longer use, or even ones with extra buttons that serve no purpose. Many remotes are designed to control additional equipment you may not own, leaving several buttons unused. It is going to be implemented in this device to hold the duty of convenience. By decoding the outputs of the remote control, using the Arduino, we are going to reverse engineer the buttons to control the movements of the shutters. For this project, the IR remote control and receiver of the Arduino kit will be used [*DroneBot Workshop, 2025*].

In order to design an intelligent and energy-efficient solution, mechanical, electrical, and software subsystems will be integrated into the automated window blinds system. The system will measure the amount of sunlight in the room by processing data from an Ambient Light Sensor

VEML7700 using an Arduino microcontroller. Two NEMA 17 stepper motors will be controlled by the Arduino using the recorded data to automatically adjust the blinds. Below you can see a representation of the design on Figure 1.

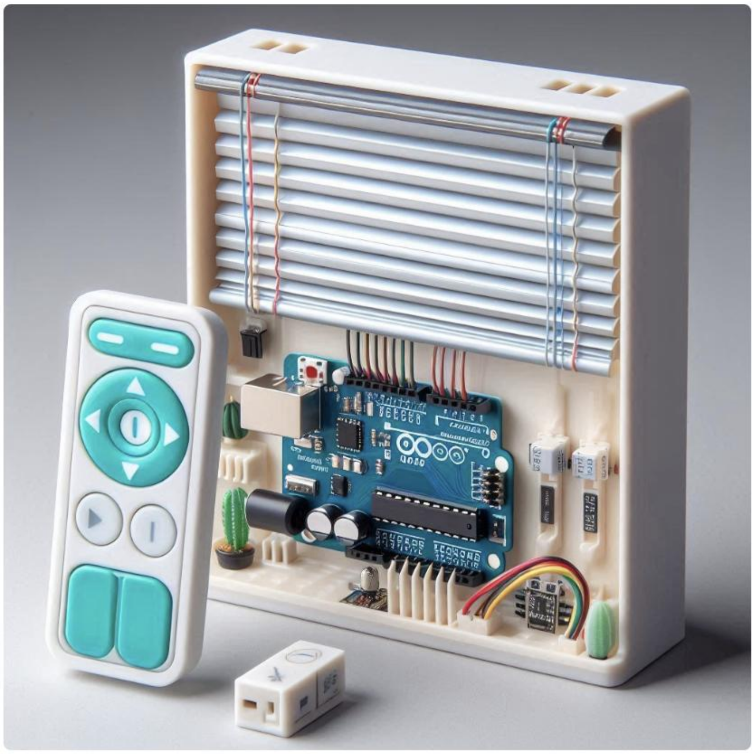


Figure 1: *Simplified Visualization of Design. This model demonstrates integration of all components for automated window blinds.*

Mechanical Subsystem

The mechanism used to lift the blinds will utilize a NEMA 17 stepper motor paired with a spool system. This spool system is responsible for winding and unwinding the pre-installed blind cords, allowing the blinds to move up and down. To determine the necessary motor torque, both static and dynamic force analyses were performed.

$$\tau = r \times F \qquad [1]$$

In this formula, r represents the spool's radius. During our analysis, we considered two possible radii. Initially, we calculated torque using a 10 mm diameter spool. However, we revised our approach and used the radius of the spool when the cord is fully wound, which is 40 mm in diameter. This adjustment reflects a worst-case scenario and ensures the motor selected will be capable of handling all conditions. Our original calculation, which used only the smaller radius, underestimated the required torque and led to a motor not strong enough to fully raise the blinds. This required us to recalculate the torque.

The force F acting on the motor is calculated as:

$$F = m \times a \qquad [2]$$

Here, m is the mass of the blinds, which was found to be about 2.27 kilograms, and a is the gravitational acceleration, 9.81 meters per second squared. Using this, we determined F to be 22.27 newtons. Substituting into equation [1], the calculated torque τ to be approximately 45 newton centimeters.

This torque is applied rotationally to the spool to either wind or unwind the cords, depending on the desired direction of motion. CAD modeling is used to fine-tune component sizing and placement, including the position of the motor mount, the spool diameter, and available space in the housing. A bracket will be attached to the blinds' headrail to secure the motor and spool system and reduce motion loss.

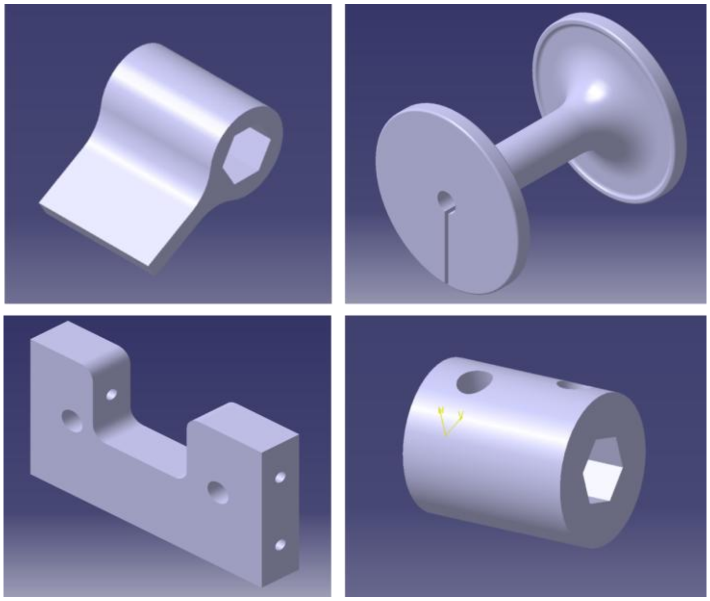


Figure 2: *CAD Components. Figure 2 consists of CAD components intended for mechanical integration between the stepper motors and the window blind assembly. These components enable precise motion transfer required for automation. (Actual components may vary)*

*The slat tilting function uses a 28BYJ48 5V stepper motor, which connects to the tilt rod through a gear-driven linkage. Stepper motors are well-suited for light load and low rotation applications. Also, they are compact enough to fit beside or in line with the headrail. The torque needed to tilt the slats is calculated as:

$$Torque = F \times r \tag{3}$$

In this equation, F is the force exerted by the gear teeth, and r is the radius of the tilt rod or gear arm. Since the slats require less than one full turn to adjust, the tilting system involves minimal travel. 3D printed CAD components are utilized for mechanical integration between the stepper motor and the input rod for seamless integration.

If the current approach for horizontal slat blinds turns out to be too complex or unreliable, a simpler alternative is to automate Zebra Roller Blinds. These blinds only require one motor to rotate the fabric layers for both lifting and adjusting light, eliminating the need for a separate tilting mechanism. A NEMA 17 stepper motor can be either directly connected to the roller or used with a pulley. This significantly reduces mechanical complexity, decreases the number of parts, and minimizes failure points. Additionally, position control is more straightforward, requiring just basic step counting and one limit switch for homing. Zebra blinds offer a space-saving, low-maintenance solution with fewer components while still providing adjustable lighting.

Electrical/Electronic Subsystem

The electrical subsystem is a key component in the operation of the automated blind system, overseeing both power management and control logic. It incorporates two stepper motors, one for lifting the blinds and another for tilting the slats, all supported by driver boards, sensors, and user input modules. The NEMA 17 bipolar stepper motor, chosen for its torque and accuracy, handles the lifting function and is operated through a DRV8825 micro-stepping driver. For slat tilting, a 28BYJ48 5V unipolar stepper motor is used with a ULN2003 driver board. Both motors draw power from a shared 12V 5A wall adapter, ensuring reliable performance across operations.

At the heart of the system is the Arduino Uno, which directs both motors. It sends STEP and DIR signals to the DRV8825 for lifting and four digital outputs to the ULN2003 for tilting. The VEML7700 ambient light sensor supplies brightness data to enable automatic adjustments. At the same time, an IR remote provides manual control via an onboard IR receiver module. Two limit switches support the startup homing process and act as safeguards by detecting when the blinds are fully closed or tilted, preventing mechanical overtravel.

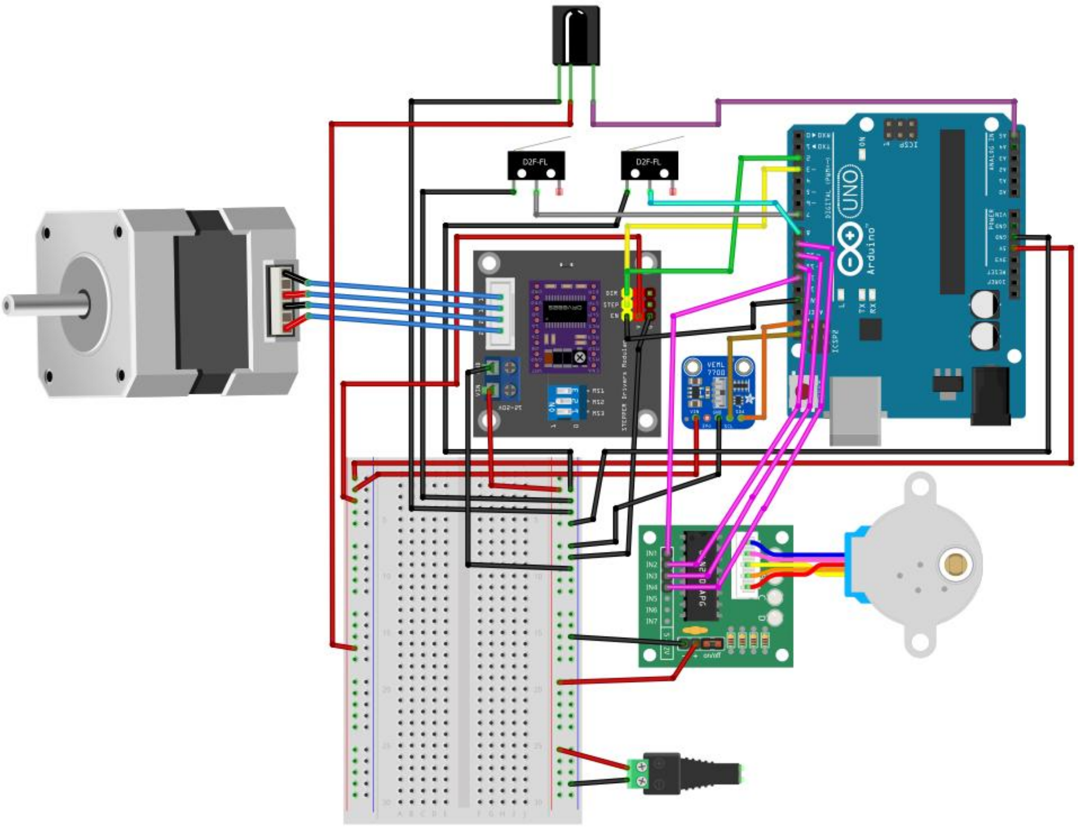


Figure 3: *Complete Wiring Layout. Includes the entire hardware integration of the Arduino Uno controlled automated blind system.*

Figure **3** presents a complete view of the hardware integration. The NEMA 17 motor is connected to the DRV8825, while the 28BYJ48 motor is linked to the ULN2003 board. Both driver boards are powered by a 12V wall adapter through a barrel jack. The Arduino sits at the center of the configuration, handling control commands and reading signals from the light sensor, IR receiver, and limit switches. Color-coded wiring is used to clearly indicate power, ground, and signal lines, streamlining both the debugging process and overall system assembly.

To protect the NEMA 17 motor and configure its current limit correctly, the Vref on the DRV8825 was calculated with the formula:

$$V_{ref} = I_{max} \times 0.5 \qquad [4]$$

For a motor rated at 2.1A, this results in a V_{ref} = 1.05V, which was precisely set using a multimeter to ensure safe and efficient operation without the risk of overheating.

An essential step in system setup is calibrating the motor travel. A test script is run to measure the number of steps required to fully raise or lower the blinds. These step counts are saved in the software, providing reliable and repeatable movement during regular use. Despite the system’s efficient design, some challenges are anticipated, including managing high current without disrupting low voltage signals and dealing with heat dissipation in compact spaces. From a software perspective, properly coordinating inputs from sensors with motor actions is crucial to avoid logical conflicts. Mechanically, any play or misalignment in the tilt motor's gear linkage can lead to positioning inaccuracies.

If necessary, several alternative solutions can be explored. These include switching to DC motors for easier control, upgrading to a more capable ESP32 microcontroller for wireless access and increased interface, or adding rotary encoders for closed-loop control, which would enhance movement precision and increase the system’s robustness.

Computer/Software Subsystem

The computer and software subsystem revolves around the Arduino Uno, which acts as the primary controller that operates the lifting and tilting functions of the automated blinds. The software is developed in C++ within the Arduino IDE and structured as a basic state machine. This state machine manages the system's behavior through various states including Home, Move Up, Move Down, Tilt, and Idle. When the system is powered on, it first enters the Home state, carrying out a homing routine by using a limit switch to identify the fully raised position of the blinds. This procedure ensures future lifting actions are precise and consistent.

The lifting component uses a NEMA 17 stepper motor managed by a DRV8825 driver. The Arduino sends pulse signals to the STEP and DIR pins on the driver to adjust the motor's speed and direction. For tilting, a 28BYJ48 stepper motor is used in combination with a ULN2003 driver board. This motor is controlled using a sequence of electrical signals through its coils, enabling precise angular positioning of the slats. The basic operation of both motors relies on a simple stepping loop that switches the step pin between high and low with brief pauses in between, moving the motor one step at a time. This process repeats to create continuous motion as required for lifting or tilting.

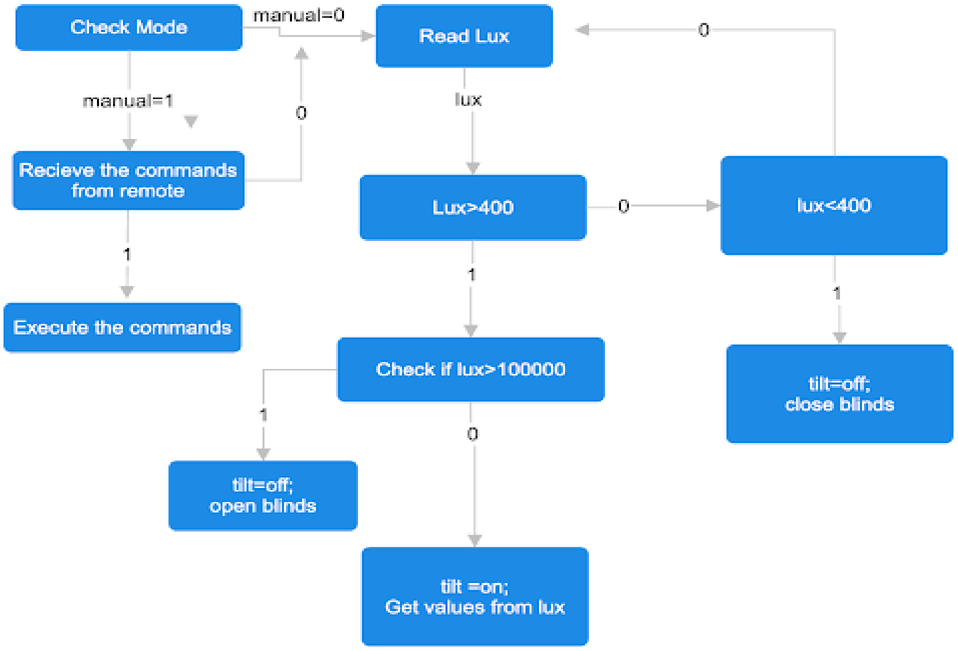


Figure 4: *Presents the flowchart outlining the logical sequence of operations for the automated blinds.*

To illustrate how the software functions, a flowchart has been created and is presented in Figure 4. The diagram maps out how the system interprets various inputs, like ambient light readings or user-selected modes, to determine the correct action, whether that means raising the blinds, adjusting the tilt, or remaining idle. The flowchart offers a visual representation of how software moves between states and reacts to external inputs or user commands.

Altogether, this approach fulfills the system's requirements by supporting both automated and manual control of the blinds, integrating live sensor data, and delivering accurate positioning via motor stepping and limit switch references. Its modular state machine format also paves the way for easy feature expansion in the future.

Potential challenges during implementation include ensuring coordination between the lift and tilt operations, particularly during state transitions. Additional care must be taken when incorporating the IR remote and light sensor, to avoid delays caused by blocking functions in the main loop. There is also a risk that electrical noise from the DRV8825 may interfere with signal integrity, and the limited number of input and output pins on the Arduino Uno could constrain scalability as more components are introduced.

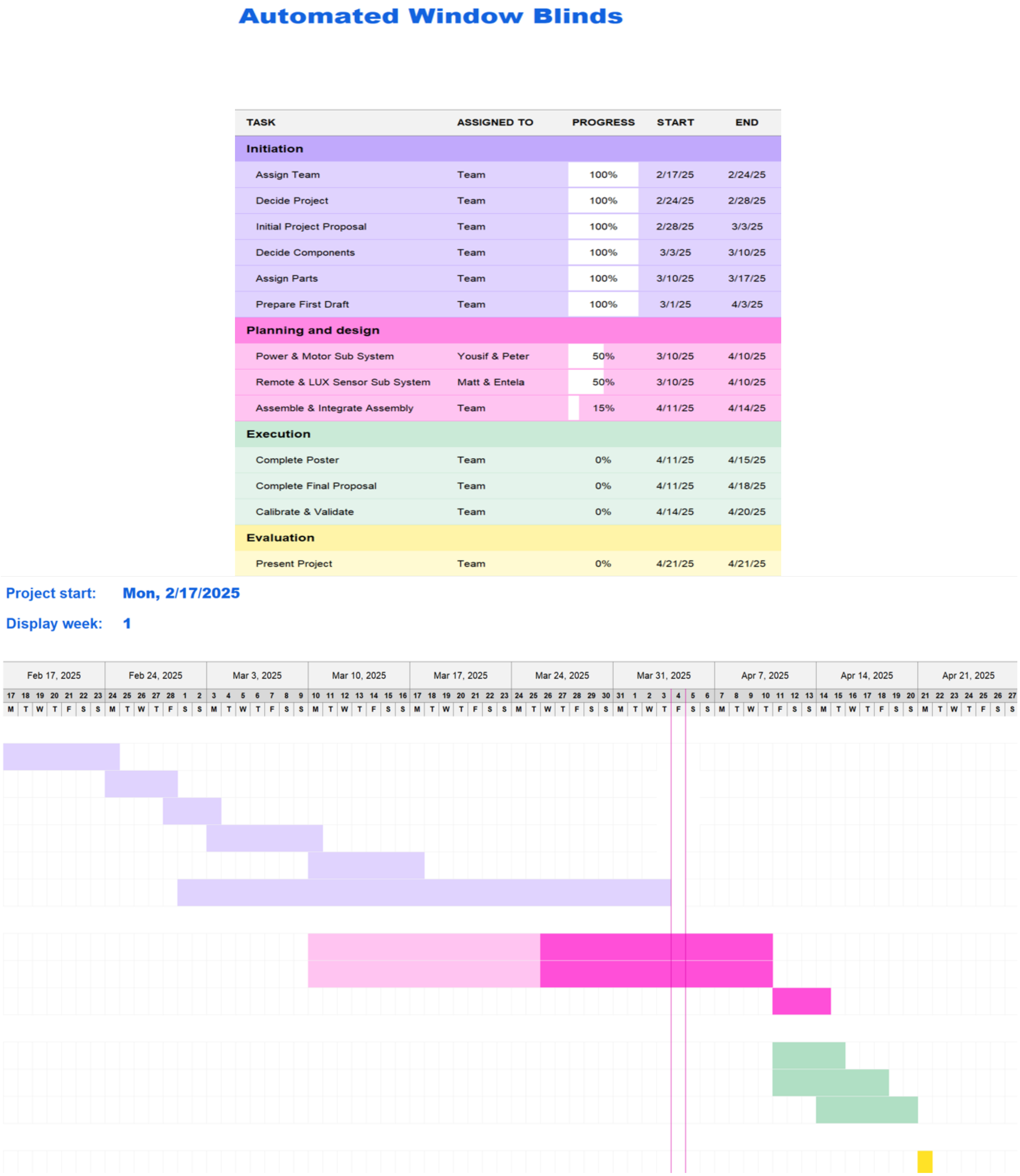
To mitigate these issues and allow for future improvements, several alternatives are being evaluated. One option is upgrading to a more advanced microcontroller like the ESP32 or Teensy, which deliver faster processing speeds and additional input output capabilities. Adding a real time clock module would support scheduled blind operations based on the time of day. Lastly, incorporating interrupt-based code would enhance responsiveness to changes from sensors or IR commands, minimizing delays that result from sequential polling.

.

Bill of Materials

Material	**Qty**	**Cost**
Arduino Uno R3	1	$ 27.60
Nema 17 stepper motor	1	$ 19.98
DRV8825 stepper driver	1	$ 2.00
28BYJ-48 5V stepper motor	1	$ 3.81
DRV8825 Drive Board	1	$ 4.55
ULN2003 driver board	1	$ 5.54
VEML7700 Ambient Light Sensor	1	$ 4.95
IR Receiver	1	$ 1.95
IR Remote Control	1	$ 1.90
12V 5A DC wall adapter	1	$ 5.00
Limit Switches	2	$1.71
Heat sink	1	$ 3.95
Blinds	1	$ 40.00
(2” x 4” x 16”) Wood for Stand	2	$ 6.00
Fritzing Software	1	$ 10.00
	Tax	$ 8.34
	Total	$ 147.28

Schedule/Timeline



Automated Window Blinds

TASK	ASSIGNED TO	PROGRESS	START	END
Initiation				
Assign Team	Team	100%	2/17/25	2/24/25
Decide Project	Team	100%	2/24/25	2/28/25
Initial Project Proposal	Team	100%	2/28/25	3/3/25
Decide Components	Team	100%	3/3/25	3/10/25
Assign Parts	Team	100%	3/10/25	3/17/25
Prepare First Draft	Team	100%	3/1/25	4/3/25
Planning and design				
Power & Motor Sub System	Yousif & Peter	50%	3/10/25	4/10/25
Remote & LUX Sensor Sub System	Matt & Entela	50%	3/10/25	4/10/25
Assemble & Integrate Assembly	Team	15%	4/11/25	4/14/25
Execution				
Complete Poster	Team	0%	4/11/25	4/15/25
Complete Final Proposal	Team	0%	4/11/25	4/18/25
Calibrate & Validate	Team	0%	4/14/25	4/20/25
Evaluation				
Present Project	Team	0%	4/21/25	4/21/25

Project start: Mon, 2/17/2025

Display week: 1

References

House Mate. (2023). *Pros and Cons of Smart Blinds*. Retrieved from https://house-mate.com.au/pros-and-cons-of-smart-blinds.html.

DroneBot Workshop, “Using IR Remote Controls with Arduino,” *DroneBot Workshop*, 2025. [Online]. Available: https://dronebotworkshop.com/using-ir-remote-controls-with-arduino/.

Arduino SensorKit, “The Light Sensor,” Arduino SensorKit, 2025. [Online]. Available: https://sensorkit.arduino.cc/sensorkit/module/lessons/lesson/05-the-light-sensor.