

ECE445

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT

**Vision-Based Gesture
Control System**

Team 15

Mingzhi Gu
Chongying Yue
Licheng Xu
Zihan Xu

Supervisor: Yushi Cheng

March 30, 2026

Summary

This document presents the design of a lighting and curtain control system based on visual gesture recognition. The system uses a Kendryte K230 module running a convolutional neural network (CNN) for gesture recognition and an STM32 microcontroller for control. The recognized gestures are sent through UART to control a 5 V LED strip and a 12V stepper-motor-driven curtain, with an end-stop switch for safety. We define quantitative requirements and verification procedures for each subsystem. The key performance goals include recognition of four swipe gestures with $\geq 60\%$ accuracy and a system response time of $\leq 2s$. All supporting analyses are documented together with quantitative requirements for the major subsystems and a final acceptance plan.

1 Introduction

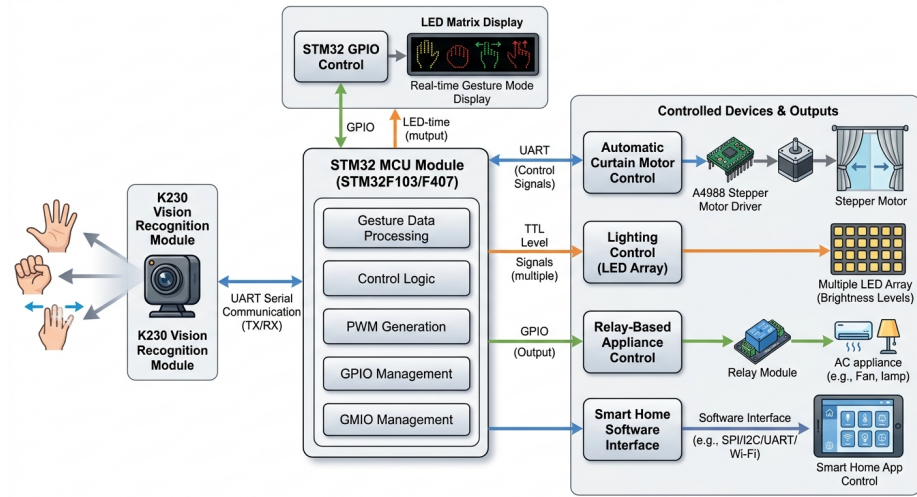


Figure 1: System overview: User performs gestures to control the smart lighting and curtain.

1.1 Problem

The goal of this project is to implement a contactless gesture-based system for controlling lights and motorized curtains. This smart-home control system provides an intuitive and hygienic alternative to physical switches. The system must be able to recognize user gestures in real time, translate them into corresponding commands, and send those control signals to the appropriate smart-home devices to perform the corresponding actions, while ensuring overall system safety and ease of use.

1.2 Proposed Solution

Our solution uses a camera-based gesture recognition pipeline. An image sensor (e.g. a USB camera) feeds pictures to a Kendryte K230-based vision module, which identifies and classifies the gestures in the image. The recognized gesture, with high confidence, is sent via UART to a microcontroller (STM32), which maps the gesture signal to commands for specific lighting and motor control. An A4988 stepper motor driver drives the curtain motor. Safety features regarding this part include NC limit switches (wired fail-safe [2]). The system operates entirely on low-voltage DC power and is designed for clear requirements and verification.

2 High-Level Requirements

These high-level requirements will be refined in each subsystem into specific error requirements and verification procedures for that subsystem's functionality. The key requirements are:

- Recognize 4 distinct swipe gestures with at least 60% accuracy across multiple users and perform the corresponding gesture to the smart-home response within 2 s.
- Control at least one light (on/off and brightness) and one motor (on/off).
- The system shall support around-the-clock operation (24h) with low power consumption (30W) and stable performance, and shall recover safely after a power cycle (press the start button again). In case of failure or communication loss, the system must shut down the outputs and produce no dangerous output.

3 Design

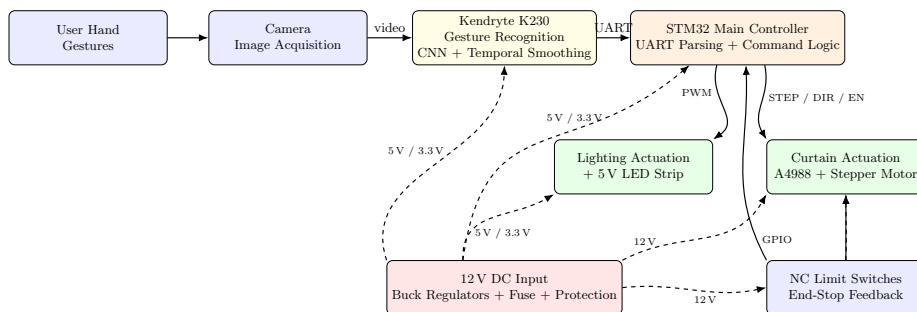


Figure 2: High-level block diagram of the project.

3.1 Vision & Recognition

Description: The function of the vision subsystem is to capture images from the camera and execute a CNN-based gesture recognition pipeline on the Kendryte K230 development board. This pipeline combines frame-wise convolutional image classification with short-window temporal smoothing techniques to enhance the robustness of dynamic sliding gestures. Its output consists of a gesture ID and a confidence score, which are then transmitted to the main controller.

Interfaces: *Input:* Camera video frames (30 fps, 640×480). *Output:* UART packet containing information on header, gesture_ID, confidence, seq#, CRC.

Hardware: Kendryte K230 AI dev board with camera sensor attached.

Software: A lightweight CNN model, e.g. MobileNet, for the vision recognition task running on the K230. The K230 packages results into the UART format.

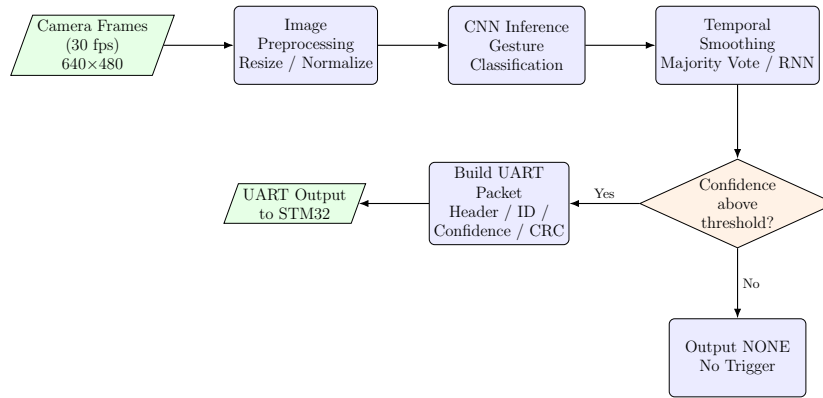


Figure 3: Flow diagram of the Vision & Recognition subsystem.

Requirements	Verifications
1. Correctly classify at least 4 pre-defined gestures with an accuracy of over 60%.	1. Conduct 50 tests across 3 different users and compute the average accuracy based on system logs to ensure it exceeds 60%. Under typical indoor lighting conditions, the distance between the user and the camera should be maintained between 0.5-2.0 meters.
2. The visual processing time per frame (t_{vis}) shall not exceed 100 ms.	2. Calculate the average time interval for 100 consecutive command events—based on K230 firmware timestamps recorded at the start of frame capture and after the completion of UART transmission to verify that the processing latency is ≤ 100 ms.
3. The hand feature inference shall operate at a sustained rate of ≥ 15 fps for at least 60 seconds, and the UART output rate during an active gesture shall be ≥ 10 Hz.	3. Connect the UART interface to a monitor and run the system continuously for over 60 seconds. Verify via timestamps that the K230 outputs valid command packets at ≥ 10 Hz during sustained gestures and maintains an internal inference log with a processing rate of ≥ 15 fps.
4. The subsystem should not store any raw video frames to any persistent storage during normal operation to protect user privacy.	4. Perform a code review of the K230 firmware to ensure no file-write operations are executed for raw image data. Also, inspect the K230 storage after thorough operation to confirm no video files or images have been saved.

Table 1: Requirements and Verifications for the Vision & Recognition Subsystem

3.2 Main Control (MCU)

Description: The STM32 microcontroller acts as the central controller. It receives gesture signals from UART, validates and debounces these signals to prevent false triggers, applies a gesture-to-action mapping, and drives PWM outputs.

Interfaces: *Input:* UART signal from vision & recognition subsystem; switch inputs. *Output:* PWM for LED, STEP/DIR/EN for the motor driver, status LED.



Figure 4: Vision & Recognition Module: K230

Hardware: STM32 MCU board STM32F407 [4]. UART pins, 2 timers for PWM, digital I/Os.

Software: The firmware parses incoming packets and sets outputs accordingly. It implements a fixed packet format (start byte, ID, CRC16). Upon receiving a valid `gesture_ID`, it updates the LED PWM output or the stepper motor command sequence. A timeout ($T_{\text{timeout}} = 1 \text{ s}$) stops all outputs if error occurs.

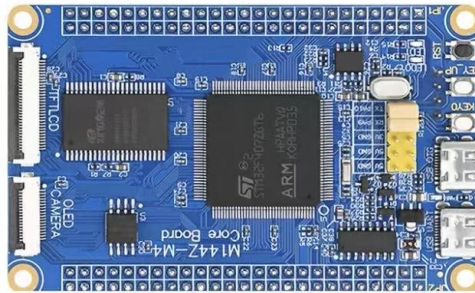


Figure 5: STM32F407 Microcontroller Board

Requirements	Verifications
1. The controller shall implement a debounce mechanism requiring exactly $N = 3$ consecutive identical UART command packets to register a valid gesture command.	1. Connect the MCU to a PC via a USB-to-TTL module. Send 2 identical packets followed by a different packet; verify via oscilloscope that output does not change. Send 3 identical packets and verify the corresponding GPIO/PWM output updates correctly.
2. The controller processing time per command event, including CRC check, debounce update, and mapping decision, shall be no greater than 200 ms with the MCU parsing step taking no more than 50 ms.	2. Add code to the firmware to raise the level of a debug GPIO pin upon receiving the first byte of a data packet, and lower it at the execution signal. Use an oscilloscope to measure the pulse width for 100 consecutive events to verify that it consistently remains no more than 200 ms.
3. For safety, all hardware outputs must be set to an OFF state at default or upon system reset, or if UART communication is lost for $T_{\text{timeout}} \geq 1$ s.	3. During normal operation with outputs active, disconnect the UART RX wire. Then use a stopwatch and logic analyzer to confirm that all motor and relay control signals transition to logic LOW within $1.0 \text{ s} \pm 0.1 \text{ s}$.

Table 2: Requirements and Verifications for the Main Control Subsystem

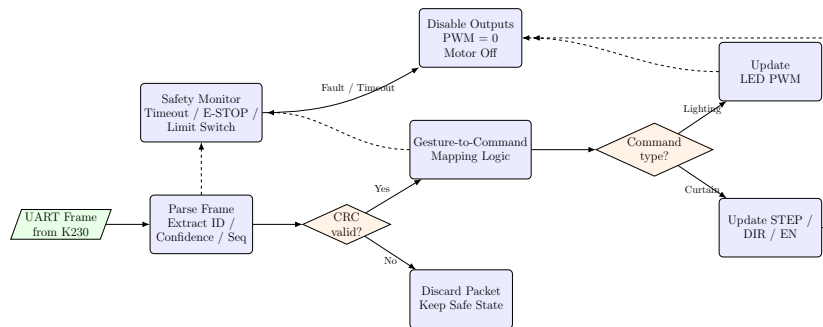


Figure 6: Flow diagram of the Main Control subsystem.

3.3 Lighting Actuation

Description: This module uses a 5 V LED strip to provide controllable lighting, where the PWM duty cycle to modulate LED brightness comes from the MCU.

Interfaces: *Input:* PWM duty command from MCU. *Output:* LED current.
Hardware: 5 V LED strip and relevant equipments for circuit robustness.

Requirements	Verifications
1. The PWM frequency driving shall be securely maintained at ≥ 2000 Hz to eliminate perceptible visual flicker.	1. Probe the MCU PWM output pin with an oscilloscope. Measure the signal frequency over a 1-minute interval and verify it remains ≥ 2000 Hz under all duty cycle conditions.
2. The subsystem shall support at least five user-selectable and distinct brightness levels, each corresponding to a predefined gesture.	2. The command system shall cycle sequentially through these five brightness levels. Measurements shall be taken in a dark room using an illuminometer positioned at a fixed distance of 0.5 meters. Verify that each level produces a distinct and repeatable illuminance reading, and that the variation in readings across three test cycles is $\leq 5\%$.
3. The wiring infrastructure must safely sustain a continuous maximum current draw of up to $250 \text{ mA} \pm 10\%$ without failure.	3. Set the PWM output to 100% duty cycle for a continuous 30-minute stress test, and measure the steady-state current using a multimeter to verify ($\leq 500 \text{ mA}$).
4. The lighting subsystem must exhibit safe electrical characteristics, remaining in an off state during initial power-up, system reset, or MCU shutdown.	4. Connect an ammeter in series with the power line of the LED strip. Apply main power to the system without providing any gesture inputs. Prior to receiving any valid commands, the measured current remains at 0 mA.

Table 3: Requirements and Verifications for the Lighting Actuation Subsystem

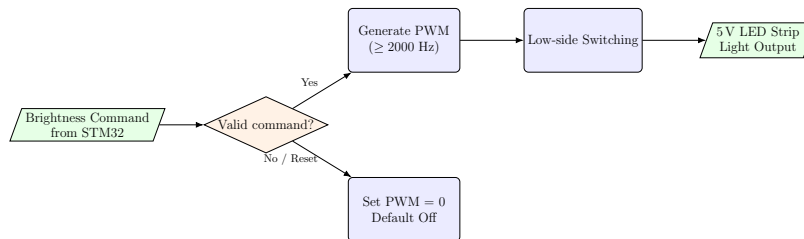


Figure 7: Flow diagram of the Lighting Actuation subsystem.

3.4 Motor/Actuator

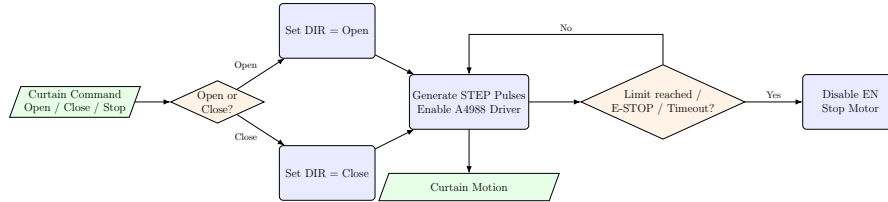


Figure 8: Flow diagram of the Motor / Actuator subsystem.

Description: A 12 V stepper motor drives the curtain via an A4988 stepper motor driver [1]. Two limit switches at the ends of the travel path would cut the drive when hit (NC wiring for failsafe [2]).

Interfaces: *Input:* STEP and DIR signals from MCU; limit switches. *Output:* Motor movement.

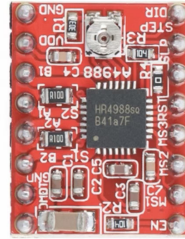
Hardware: 12 V stepper motor; A4988 stepper motor driver; two NC limit switches; other required decoupling and protection components.

Requirements	Verifications
1. Full travel ≤ 5 s	1. Send full open/close command, time with stopwatch over 10 trials, and verify each trial completes within 5 s.
2. Limit switch stops ≤ 0.1 s	2. Hit the limit switch during motion, measure time from switch actuation to STEP/EN disable with an oscilloscope or logic analyzer, and verify the time ≤ 0.1 s.
3. Stop on no-signal	3. Remove the command signal during motion, verify the controller disables motor drive and the curtain stops safely within 1.0 s.
4. Sufficient torque to drive curtain load	4. Set the actual curtains or a counterweight with an equivalent rated load, then issue full open/close commands to run for 10 consecutive cycles. Verify that the motor starts reliably and completes the full range of motion each time, without stalling, losing steps, or failing to actuate.

Table 4: Requirements and Verifications for Motor / Actuator Subsystem



(a) Stepping Motor



(b) A4988 Motor Drive

Figure 9: Stepping Motor and Driver

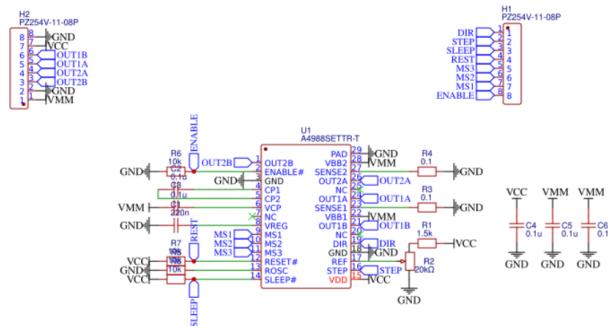


Figure 10: A4988 Circuit Schematics. Source: adapted from the open-hardware project "A4988 stepping motor drive" by akatople on Oshwhub

3.5 Power and Protection

Description: A single 12 V DC adapter powers the system. The 12 V rail feeds the stepper motor driver and is stepped down to 5 V/3.3 V for logic and LED lighting. Other protective elements include a fuse, reverse-polarity diode, and TVS transient suppressor.

Interfaces: *Input:* 12 V DC source. *Output:* 12 V rail to motor driver; regulated 5 V/3.3 V rails. Limit switches break the motor driver enable path.

Hardware: 12 V/3 A DC adapter; 5 V/3.3 V regulators; Schottky diode for reverse protection; TVS diode for spikes.

Requirements	Verifications
1. Under maximum combined load, the primary 12 V rail shall maintain a voltage of $12\text{ V} \pm 10\%$.	1. Connect a programmable DC electronic load or operate the system at full capacity. Probe the 12 V rail with a multimeter and oscilloscope. Verify the steady-state voltage remains between in the range over a 5-minute continuous test, with ripple noise $\leq 200\text{ mV}$.
2. Regulated logic power rails should be maintained within $\pm 5\%$ of their nominal values (4.75 V to 5.25 V for the 5V rail, and 3.135 V to 3.465 V for the 3.3V rail) to prevent undervoltage conditions at the MCU.	2. When the system is rapidly and frequently starting and stopping the LED array and motors, use a multimeter to measure the 5V and 3.3V power rails. Verify that the voltage readings consistently remain no lower than 4.75 V and 3.135 V respectively.
3. Activating a limit switch must immediately interrupt the motor driver's travel path, thereby halting motion regardless of the MCU's current software state.	3. Command the motor to run at full speed. Manually trigger the limit switch. Verify both visually and using a Multimeter that the motor stops instantaneously and that the voltage at the motor driver's enable pin drops to the disabled state.

Table 5: Requirements and Verifications for the Power and Protection Subsystem

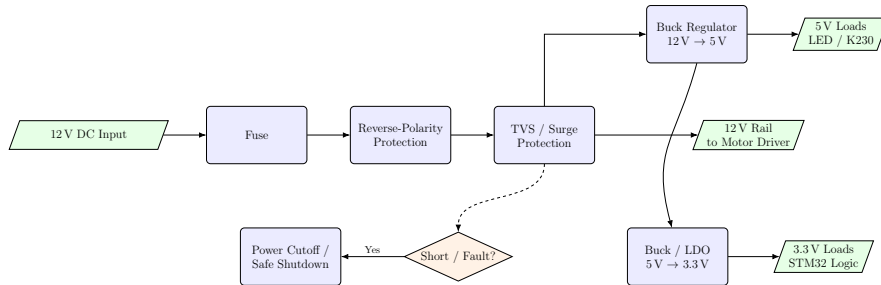


Figure 11: Flow diagram of the Power and Protection subsystem.

3.6 Supporting Material

UART Packet Format: We use a fixed 7-byte frame with a header and CRC to avoid ambiguity. The format is shown in Table 6.

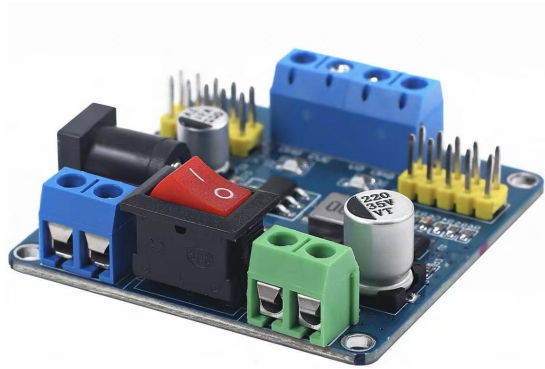


Figure 12: Buck Circuit

Field	Bytes / Description
Start Delimiter	1 byte (0xAA)
Version	1 byte (protocol)
Gesture ID	1 byte (0=None, 1=Palm, 2=Fist, 3=SwipeRight, 4=SwipeLeft, ...)
Confidence	1 byte (0-255)
Sequence #	1 byte (increments)
CRC-16	2 bytes (CRC of prior fields)

Table 6: UART Packet Format (Microcontroller Input)

Gesture-to-Command Mapping: Mapping of gestures to actions are shown in Table 7.

Gesture	Command Action
Swipe Up	Increase LED brightness
Swipe Down	Decrease LED brightness
Swipe Right	Open curtain
Swipe Left	Close curtain

Table 7: Gesture Vocabulary and Command Mapping

4 Tolerance Analysis

The most safety-critical function of this system is the end-stop behavior of the motorized curtain. When the curtain reaches either travel limit, the system must stop it quickly and predictably so that mechanical impact and pinch hazards are minimized.

In this design, the STM32 uses a timer-generated pulse train to drive the STEP input of the A4988 driver. Therefore, the curtain motion is determined primarily by the STEP pulse frequency rather than by an analog motor voltage. Based on this control method, the overshoot after an end-stop event can be analyzed from two parts: (1) the distance traveled during signal cut-off latency, and (2) the distance traveled after the STEP pulses stop but before the mechanism comes fully to rest.

4.1 Pulse-to-motion relationship

Let f_{step} denote the STEP pulse frequency applied to the A4988, and let N_{rev} denote the number of microsteps required for one full revolution of the motor shaft. If the curtain is driven by a spool or pulley of effective radius r , then the linear distance produced by one microstep is

$$\Delta x_{\text{step}} = \frac{2\pi r}{N_{\text{rev}}}. \quad (1)$$

Equation (1) gives the linear displacement corresponding to a single STEP pulse. Here, r is the effective spool radius, and N_{rev} depends on the motor step angle and the selected microstepping mode of the A4988.

The corresponding angular speed of the motor shaft is

$$\omega = \frac{2\pi}{N_{\text{rev}}} f_{\text{step}}. \quad (2)$$

Equation (2) relates the STEP pulse frequency to the shaft angular speed. This shows that, for a given microstepping configuration, increasing f_{step} directly increases motor speed.

The linear curtain speed is then

$$v = r\omega = \frac{2\pi r}{N_{\text{rev}}} f_{\text{step}}. \quad (3)$$

Equation (3) is the key speed relation for this system. It states that the curtain speed is proportional to the STEP pulse frequency. This is why the end-stop safety analysis can be written directly in terms of f_{step} .

4.2 Overshoot during cut-off delay

When the limit switch is triggered, the system does not stop instantaneously. Let t_{cut} denote the total delay from limit-switch actuation to the moment when STEP output is stopped or the driver is disabled.

During this delay, the number of additional STEP pulses is

$$n_{\text{delay}} = f_{\text{step}} t_{\text{cut}}. \quad (4)$$

Equation (4) gives the number of extra commanded microsteps generated during the cut-off interval.

The corresponding linear overshoot caused by this delay is

$$d_{\text{delay}} = n_{\text{delay}} \Delta x_{\text{step}} = \frac{2\pi r}{N_{\text{rev}}} f_{\text{step}} t_{\text{cut}} = v t_{\text{cut}}. \quad (5)$$

Equation (5) shows that the delay-related overshoot is proportional to both the curtain speed and the total cut-off latency. Therefore, reducing either the maximum STEP frequency or the stop latency directly reduces this part of the overshoot.

4.3 Overshoot after pulse output stops

Even after the STEP pulses stop, the curtain may continue moving because of inertia and compliance in the mechanism. Let a denote the magnitude of the passive deceleration after command cut-off.

Assuming a constant deceleration model, the additional coasting distance is

$$d_{\text{inertia}} = \frac{v^2}{2a}. \quad (6)$$

Equation (6) is the standard kinematic stopping-distance expression. It describes the extra distance traveled after active stepping stops but before the curtain fully comes to rest. A larger deceleration a reduces this term, while a larger speed v increases it quadratically.

4.4 Total end-stop overshoot

The total overshoot is the sum of the delay term and the inertia term:

$$d_{\text{total}} = d_{\text{delay}} + d_{\text{inertia}}. \quad (7)$$

Substituting Equations (5) and (6) into Equation (7) gives

$$d_{\text{total}} = v t_{\text{cut}} + \frac{v^2}{2a}. \quad (8)$$

Using Equation (3), the same expression can also be written directly in terms of STEP pulse frequency:

$$d_{\text{total}} = \left(\frac{2\pi r}{N_{\text{rev}}} f_{\text{step}} \right) t_{\text{cut}} + \frac{1}{2a} \left(\frac{2\pi r}{N_{\text{rev}}} f_{\text{step}} \right)^2. \quad (9)$$

Equation (9) is the main design equation for this subsystem. It connects the control variable f_{step} with the safety-related quantity d_{total} . Once the mechanical parameters and stop latency are measured experimentally, this equation can be used to determine whether a chosen maximum STEP frequency is safe.

4.5 Design constraint from overshoot requirement

Let d_{\max} denote the maximum allowable overshoot at the curtain end stop. Then the design must satisfy

$$d_{\text{total}} \leq d_{\max}. \quad (10)$$

Combining Equation (8) with Equation (10) yields the admissible speed condition

$$vt_{\text{cut}} + \frac{v^2}{2a} \leq d_{\max}. \quad (11)$$

Solving Equation (11) for the positive allowable speed gives

$$v \leq -at_{\text{cut}} + \sqrt{a^2t_{\text{cut}}^2 + 2ad_{\max}}. \quad (12)$$

Equation (12) gives the maximum allowable curtain speed that still satisfies the end-stop overshoot requirement. It shows explicitly how the safety margin depends on the stop latency t_{cut} , the passive deceleration a , and the allowable overshoot d_{\max} .

Finally, converting Equation (12) back to STEP pulse frequency using Equation (3) gives

$$f_{\text{step}} \leq \frac{N_{\text{rev}}}{2\pi r} \left(-at_{\text{cut}} + \sqrt{a^2t_{\text{cut}}^2 + 2ad_{\max}} \right). \quad (13)$$

Equation (13) is the final control-oriented constraint. It gives the maximum allowable STEP pulse frequency for the A4988-based curtain drive. In practice, this formula can be used to set the upper speed limit in firmware, or to define a lower approach speed near the end stops for additional safety margin.

No numerical assumptions are inserted here. Instead, the required parameters r , N_{rev} , t_{cut} , a , and d_{\max} should be obtained from the final mechanical design and from subsystem verification measurements.

5 Cost (BOM)

Table 8 lists major components with estimated costs.

Table 8: Bill of Materials Estimate

Item	Qty	Unit Cost (RMB)	Source / Notes
K230 AI Dev Board	1	350	Kendryte board [3]
USB Camera (640x480)	1	50	Standard webcam
STM32F407 MCU Board	1	150	STM32F407 dev board [4]
5V LED Strip (1m)	1	20	
A4988 Stepper Motor Driver	2	10	Driver module [1]
Stepper Motor (12V)	1	80	
Limit Switches (NC)	2	5	
12V 3A Adapter	1	60	
Buck Regulator (12→5 V)	1	20	Must support ≥ 2 A continuous
Protection components	1 set	20	
Shipping	—	50	Covers shipping
Wires / Fuse / Misc.	—	50	
PCB Manufacture	4	30	some components may be broken
Spare items	any	500	some components may be broken
Total		1500	

6 Schedule

Table 9 shows a tentative 6-week plan and task assignments.

Table 9: Tentative 6-Week Schedule and Assignments

Week	Tasks	Members
1	(1) Determine the overall system requirements of the project and the interface specifications between the vision module and the smart-home devices; (2) complete the hardware selection and related circuit planning for the vision recognition and motor drive subsystems; (3) set up the STM32 development environment to prepare for future development; (4) formulate the annotation and specification plan for gesture dataset collection according to the task requirements.	Mingzhi Gu (1), Chongying Yue (2), Licheng Xu (3), Zihan Xu (4)
2	(1) Collect and label the corresponding gesture samples, and use the data to fine-tune and train the vision model; (2) design and implement the STM32 control framework in hardware; (3) draw the initial motor drive and power schematics; (4) coordinate the development of each module, align the interfaces between modules, and review the progress.	Zihan Xu (1), Licheng Xu (2), Chongying Yue (3), Mingzhi Gu (4)
3	(1) Optimize the gesture model and test the accuracy of different gestures at different heights under different lighting conditions; (2) realize the communication between different modules using the UART protocol and design the packet parser; (3) build the prototype circuit according to the design drawings and carry out the power-on test; (4) provide support for the overall integration planning and debugging.	Zihan Xu (1), Licheng Xu (2), Chongying Yue (3), Mingzhi Gu (4)
4	(1) Generate and verify gesture commands for the vision module; (2) implement PWM/GPIO control and command parsing; (3) carry out integrated debugging of the furniture control interface; (4) provide hardware support for system integration.	Zihan Xu (1), Licheng Xu (2), Mingzhi Gu (3), Chongying Yue (4)

Week	Tasks	Members
5	(1) Integrate the motor drive circuit, optimize power management, and implement the related safety protections; (2) realize the control of lighting and the corresponding limit-switch protection; (3) coordinate the overall system debugging and stability testing to ensure that the accuracy and timing error meet our requirements; (4) assist other parts in verifying the command response to ensure the integrity of the whole system.	Chongying Yue (1), Licheng Xu (2), Mingzhi Gu (3), Zihan Xu (4)
6	(1) Complete the overall system integration and stability testing to ensure the safety performance of the power supply subsystem; (2) evaluate the final gesture recognition accuracy to ensure that the whole response chain operates correctly; (3) complete STM32 control optimization and debugging; (4) check overall hardware reliability and try to improve power performance.	Mingzhi Gu (1), Zihan Xu (2), Licheng Xu (3), Chongying Yue (4)

7 Ethics and Safety

This project includes a camera-based user interface and electromechanical actuators (motorized curtains). Our design goals are guided by engineering ethics guidelines and relevant safety/regulatory standards.

Referenced Ethical Guidelines. We are committed to following the IEEE Code of Ethics and the ACM Code of Ethics and Professional Conduct. These guidelines emphasize prioritizing public safety and well-being, honestly explaining system limitations, and protecting user privacy [6, 7].

Electrical Safety. The system is powered entirely by low-voltage DC ($\leq 12\text{V}$) to reduce the risk of electric shock. We follow OSHA electrical safety guidelines, implementing safe wiring, protection, and operating procedures during testing, such as ensuring secure wiring and proper insulation; and avoiding live work [8].

Overcurrent, Fire, and Thermal Risk Mitigation. In our circuit design, we use a series input fuse and incorporate reverse connection protection to reduce the risk of continuous fault current and wire overheating. In laboratory testing, we will use a current-limiting benchtop power supply, where conditions permit. We will validate each power rail under maximum combined load conditions to reduce the risk of unsafe behavior due to undervoltage. The validation plan also includes thermal stress testing of the LED and motor drivers to ensure

that surface temperatures are kept within reasonable ranges during long-term operation.

Mechanical and Pinch Risk Mitigation. Moving curtains may pinch fingers or snag objects. Our mitigation measures include: (i) using normally closed (NC) limit switches with fail-safe characteristics [2]; (ii) setting up a hardware path to immediately disable the motor driver when the limit switch is open; and (iii) shutting down all outputs in the event of communication loss via a firmware timeout mechanism. We have set a conservative maximum step frequency and verified through analysis (as shown in *Tolerance Analysis*) and bench testing (see Motor subsystem R&V) that the overshoot after the limit does not exceed 10 mm.

Visual Privacy and Data Processing. To mitigate privacy risks, raw video frames are processed locally within the vision module and are not written to persistent storage during normal operation. Only high-level gesture IDs and confidence metrics are transmitted via UART. This prototype does not include a network interface, limiting the possibility of unauthorized remote access.

Honest Declarations, Limitations, and Inclusivity. Gesture recognition accuracy is affected by lighting, background clutter, and user posture. Based on ethical requirements, we will truthfully report measured accuracy and failure cases, and will not claim that the system possesses sign language translation or personal identification capabilities. Testing will cover multiple users; when the confidence level falls below a threshold, the system defaults to a safe "no action" state.

References

- [1] Allegro MicroSystems, "A4988 DMOS Microstepping Driver with Translator and Overcurrent Protection," Datasheet, 2016.
- [2] Banner Engineering, *Safety Limit Switches Guide*, 2018.
- [3] Canaan Inc., "Kendryte K230 Data Sheet," 2020.
- [4] STMicroelectronics, "STM32F407xx Datasheet," 2018.
- [5] akatople, "A4988 stepping motor drive," Oshwhub, May 30, 2024. [Online]. Available: <https://oshwhub.com/akatople/a4988-bu-jin-dian-ji-qu-dong>. [Accessed: Mar. 31, 2026].
- [6] Institute of Electrical and Electronics Engineers (IEEE), "IEEE Code of Ethics," PDF. [Online]. Available: https://ewh.ieee.org/cmte/substations/posted_documents/ieee_codeofethics.pdf. [Accessed: Apr. 01, 2026].

- [7] Association for Computing Machinery (ACM), “ACM Code of Ethics and Professional Conduct,”. [Online]. Available: <https://www.acm.org/diversity-inclusion/code-of-ethics>. [Accessed: Apr. 01, 2026].
- [8] Occupational Safety and Health Administration (OSHA), “29 CFR Part 1910 Subpart S – Electrical,”. [Online]. Available: <https://www.osha.gov/laws-regs/regulations/standardnumber/1910/1910SubpartS>. [Accessed: Apr. 01, 2026].

A Component Technical Tables

A.1 A4988 Stepper Motor Driver

Parameter	Value
Device type	Bipolar stepper motor driver
Motor supply voltage	8–35 V
Logic level range	3.0–5.5 V compatible
Microstepping	Full, 1/2, 1/4, 1/8, 1/16
Peak current	Up to about ± 2 A at chip level
Typical continuous current	About 1 A per phase on common carrier boards without extra cooling
Control interface	STEP, DIR, ENABLE, RESET, SLEEP, MS1–MS3

A.2 WHEELTEC High-Torque Stepper Motor

Parameter	Value
Product type	Integrated stepper motor, often 42-series and optionally geared
Input voltage	10–25 V
Max operating current	1.2 A
Speed range	About 9.5–420 RPM
Main controller	STM32F407C8T6 (per public documentation)
Driver IC	A4988
Control modes	Speed closed-loop, position closed-loop, torque control, absolute-angle closed-loop
Interfaces	Type-C, CAN, TTL serial, NPN external control
Operating temperature	0–60°C

A.3 ALIENTEK STM32F407 Development Board

Parameter	Value
Board	ALIENTEK Explorer STM32F407
MCU	STM32F407ZGT6
CPU core	Arm Cortex-M4 with FPU
Max clock	168 MHz
On-chip Flash	1 MB
On-chip SRAM	192 KB
External SRAM	1 MB
External Flash	16 MB SPI Flash
On-board interfaces	USB-UART, SD card, Ethernet, LCD, USB Host/Device, JTAG/SWD

A.4 Yahboom K230 Board

Parameter	Value
Board type	Edge AI vision development board
Main SoC	Kendryte K230
CPU architecture	Dual-core RISC-V C908
CPU clock	About 1.6 GHz + 0.8 GHz
AI engine	Built-in KPU with INT8 / INT16 support
Imaging and multimedia	Supports ISP and coordinated image, video, audio, and AI acceleration
Common interfaces	UART, I2C, PWM, USB OTG, SD/eMMC, MIPI CSI/DSI (subject to board documentation)
Typical power input	5 V, commonly via Type-C or module interface
AI demo current	Around 0.6 A for the standard version and 0.8 A for the deluxe version

A.5 6–24V to 3.3V/5V Buck DC-DC Converter Module

Parameter	Value
Module type	Buck DC-DC switching step-down module
Input voltage	6–24 V
Output voltage	Fixed 3.3 V or fixed 5 V
Topology	Switching step-down
Typical efficiency	Usually about 85%–95%, depending on the IC and the load
Output current	Strongly model-dependent, commonly from a few hundred mA to 4 A+
Output ripple	Depends on controller choice, layout, load, and thermal conditions
Protection	Some modules have overload, thermal, or short-circuit protection