Vision-Based Gesture Recognition Smart Furniture Control System

Project Proposal (ECE 445/ME 470 ZJUI)

Team 15

Zihan Xu	zihan19@illinois.edu
Licheng Xu	lx8@illinois.edu
Chongying Yue	yue25@illinois.edu
Mingzhi Gu	mingzhi4@illinois.edu

Instructor: Yushi Chen

March 24, 2026

Contents

1 Introduction

1.1 Problem

Current smart-home systems primarily rely on voice commands or mobile applications for operation. While these interaction methods are widely available, they are not always the most efficient or intuitive for everyday furniture control. In particular, app-based control often requires multiple attention and interaction steps, which can reduce usability and responsiveness for quick and frequent actions.

At the same time, voice-centered interfaces are not equally accessible to all users. Related accessibility studies on voice assistants note that speech impairments can present a significant barrier, and that many systems are designed around clear and intelligible speech [1]. Therefore, a more direct, vision-based interaction method could improve general smart-home interaction efficiency while also providing meaningful accessibility benefits for users with hearing-related communication needs.

1.2 Solution

We propose a **vision-based gesture recognition smart furniture control system** that translates a small set of predefined hand gestures into commands for furniture devices, such as turning lights on or off, adjusting light brightness to predefined levels, and opening or closing a curtain. The core interaction is as follows: a user performs a gesture in view of a camera; the vision processing unit captures the image stream, performs gesture image classification, and outputs a command token; a microcontroller-based main control unit then validates and debounces the command, and finally actuates the target device safely through driver hardware.

For gesture perception, the system will use a real-time image-based classification pipeline built on basic computer vision and machine learning methods. The camera continuously captures user gesture images, and the vision processing unit classifies each input frame into one of several predefined gesture categories. For edge feasibility, we target an embedded vision platform (e.g., Kendryte K230) that supports low-latency image processing and on-device inference [2]. The gesture recognition stage will rely on a lightweight image classification model rather than large models or hand landmark extraction. A basic supervised machine learning approach will be used to distinguish a fixed set of gestures with low computational cost and sufficient real-time performance. The recognized gesture classes will then be mapped to corresponding control commands, including not only discrete device actions but also brightness-control commands for the lighting subsystem, allowing the user to switch among different brightness levels through designated gestures.

1.3 Visual Aid (Context Diagram)

Figure 1 shows how the system is intended to be used. The user performs hand gestures in a defined interaction zone. The camera + vision unit performs recognition. The controller supports control for different interfaces including automatic curtain motor, lighting, etc.

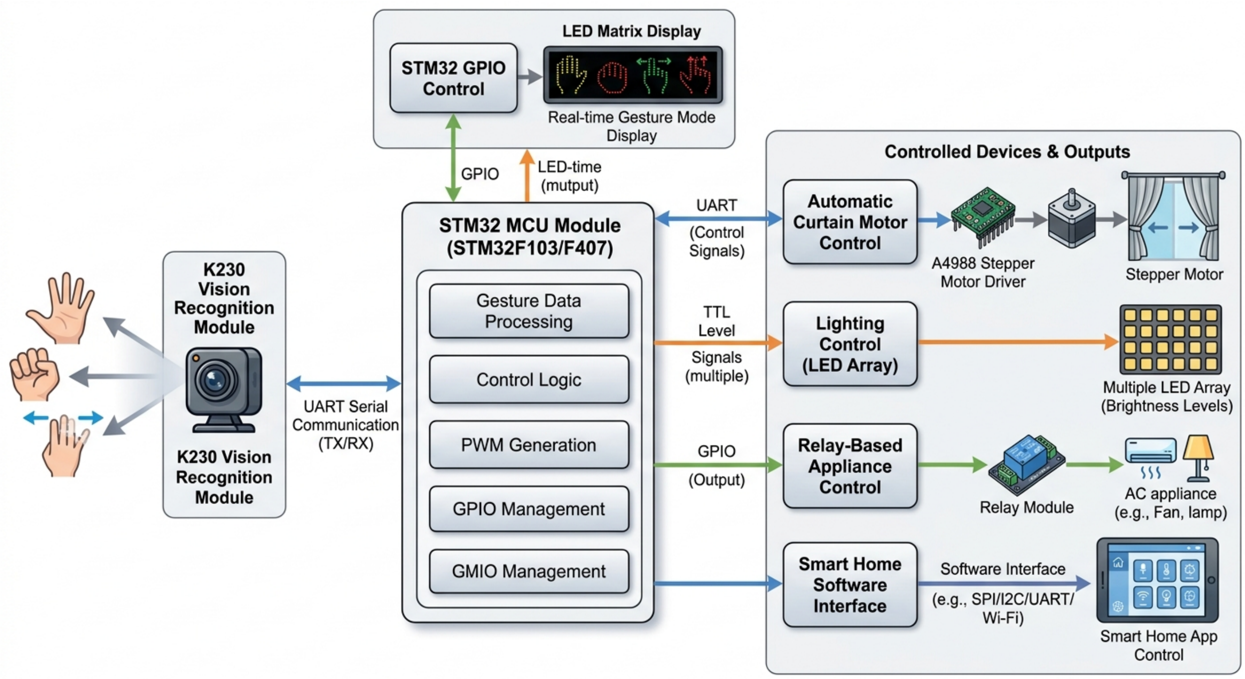
LED Matrix Display
STM32 GPIO Control
Real-time Gesture Mode Display
GPIO
LED-time (mutput)
K230 Vision Recognition Module
K230 Vision Recognition Module
UART Serial Communication (TX/RX)
STM32 MCU Module (STM32F103/F407)
Gesture Data Processing
Control Logic
PWM Generation
GPIO Management
GMIO Management
UART
(Control Signals)
TTL Level
Signals (multiple)
GPIO
(Output)
Controlled Devices & Outputs
Automatic Curtain Motor Control
A4988 Stepper Motor Driver
Stepper Motor
Lighting Control (LED Array)
Multiple LED Array (Brightness Levels)
Relay-Based Appliance Control
Relay Module
AC appliance (e.g., Fan, lamp)
Smart Home Software Interface
Software Interface
(e.g., SPI/I2C/UART/ Wi-Fi)
Smart Home App Control

Figure 1: Overview of Smart Furniture Control System

1.4 High-Level Requirements

We specify three quantitative, system-level characteristics that determine success.

- **HLR 1.** The system shall correctly classify at least four predefined hand gestures, achieving a macro-average classification accuracy more than 60% measured over 50 test trials across 3 users under typical indoor lighting at a user-camera distance of 0.5-2.0 m.
- **HLR 2.** The system shall achieve an average end-to-end control latency ≤ 1.5 s, measured from gesture completion to response , over ≥ 100 command events.
- **HLR 3.** The system shall control at least two distinct furniture device types: (i) a lighting load and (ii) a motorized load.

2 Design

2.1 Block Diagram

Figure 2 decomposes the design into subsystems and labels data interfaces. Each block is designed to be testable independently.

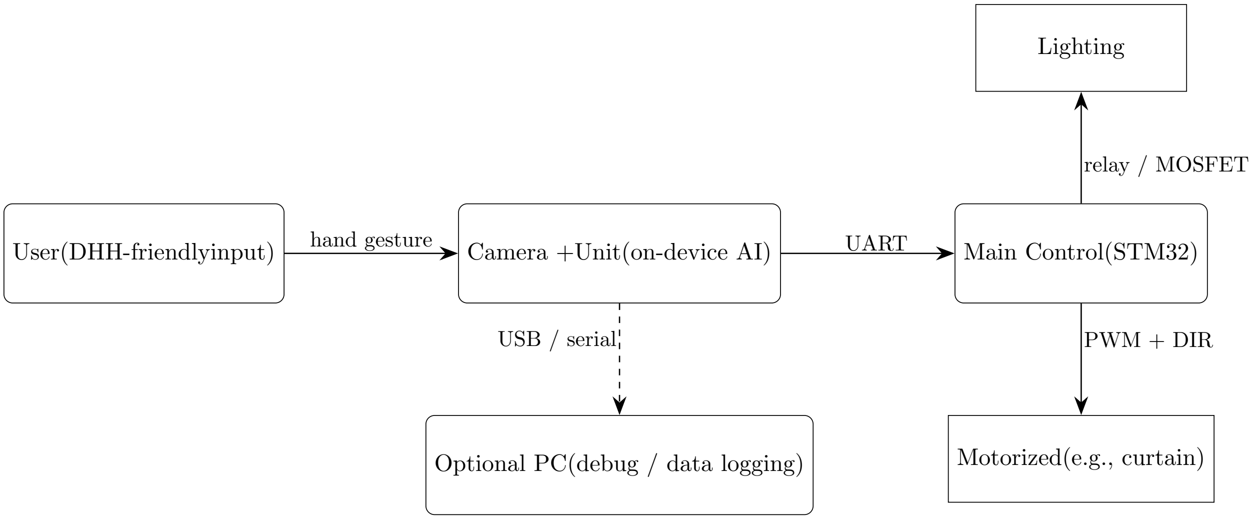

Figure 2: System block diagram with labeled data links and voltages.

2.2 Subsystem Overview

Vision & Recognition Subsystem: Captures frames and computes a gesture command with confidence. The design targets on-device inference: MediaPipe-like landmark extraction (or equivalent) over live video streams and edge compute on K230-class hardware [2]. Output is sent to the control subsystem via UART.

Main Control Subsystem: Validates and debounces commands, maps gestures to actions, and drives actuation blocks (relay and motor driver). It also exports status signals (LED indicators) so users receive non-audio feedback consistent with accessibility needs.

Motor Actuation Subsystem: Drives a DC motor/actuator with PWM speed control and direction control. It implements motion limiting using limit switches or a time/position bound to satisfy the "avoid continued motion after end-of-travel" requirement.

2.3 Subsystem Requirements

Vision & Recognition Subsystem (Camera + K230)

Contribution to overall design: This subsystem determines whether the system can achieve the required command accuracy and latency (HLR 1 and HLR 2).

Interfaces:

- Camera input: K230 integrated camera, resolution $\geq 640 \times 480$, frame rate ≥ 30 **fps**.
- Output: UART command packet; update rate ≥ 1 **Hz**.

Requirements:

- R1: Hand landmark/feature inference shall operate at ≥ 15 **fps** sustained, measured over ≥ 60 **s**, to support low-latency interaction (targeting HLR 2).
- R2: The subsystem shall not store raw video frames to persistent storage during normal operation; if logging is used for development, it shall be opt-in and time-limited (ethics/privacy).

Main Control Subsystem (STM32 parsing + debounce + mapping)

Contribution to overall design: Implements reliable command interpretation and enforces system-level latency and safety (HLR 2, HLR 3).

Interfaces:

- UART RX: receives command packets at ≥ 10 **Hz**.
- GPIO outputs: relay enable, motor direction, PWM (timer output).
- User feedback: status LED(s).

Requirements:

- R1: The controller shall require $N = 3$ **consecutive identical** communication packets.
- R2: controller processing time per command packet (CRC check + debounce update + mapping decision) shall be ≤ 200 **ms** (measured by GPIO toggle timing).

Motor Actuation Subsystem (H-bridge, PWM, limit protection)

Contribution to overall design: Enables a second device class with safe motion control (HLR 3).

Requirements:

- R1: Shall drive a DC motor at **12 V**.
- R2: PWM frequency shall be $400\,\textbf{Hz} \pm 5\%$ to reduce audible noise.

2.4 Tolerance Analysis (Risk Demonstration)

A key project risk is meeting the **end-to-end latency requirement** (HLR 2) in realistic conditions. We demonstrate feasibility by a conservative latency budget:

Assumptions (conservative):

- Camera stream at 30 fps $\Rightarrow$ frame period $T_f \approx 33.3$ ms.
- Debounce requires $N = 3$ consecutive frames with same gesture $\Rightarrow$ worst-case debounce window $\approx 3T_f \approx 100$ ms.
- Vision processing per frame budget: $t_{vis} \leq 400$ ms (targeted by using an embedded AI accelerator platform such as K230 [2] and leveraging hand/gesture-oriented demo workflows [3]).
- UART packet time is negligible: 5-byte packet at 115200 bps is $\approx (5 \times 10)/115200 \approx 0.43$ ms (10 bits/byte with start/stop).
- MCU parse + decision: $t_{mcu} \leq 50$ ms.
- Actuation command propagation: $t_{act} \leq 500$ ms .

Worst-case latency estimate:

$$t_{e2e} \approx (NT_f) + t_{vis} + t_{uart} + t_{mcu} + t_{act}$$

$$t_{e2e} \approx 100 \text{ ms} + 400 \text{ ms} + 1 \text{ ms} + 50 \text{ ms} + 500 \text{ ms} \approx 1000 \text{ ms}$$

Even if t_{vis} increases significantly (e.g., to 300 ms due to difficult scenes), the total remains below the 1.5 s requirement, leaving margin for real-world variability.

3 Ethics and Safety

Next, we will discuss the ethical and safety issues, including those arising during development and from accidental/intentional misuse, with relevant safety/regulatory standards [4].

3.1 Ethics

Privacy and consent: A vision-based system operating in a home-like environment could capture sensitive information. Privacy concerns are documented in smart-home contexts, including worries about being observed or judged based on activity patterns [5]. To mitigate this, our default design processes video **only on-device** and produces only minimal outputs (gesture ID, confidence). We will avoid storing raw video in normal operation and instead log only anonymized landmark vectors during development when necessary (opt-in, time-limited). This aligns with IEEE ethics principles emphasizing public welfare and protecting privacy [4].

Accessibility and inclusion: While the system targets users who benefit from non-audio control modes, we acknowledge DHH communication diversity; not all DHH users use sign language, and voice assistants can be inaccessible for some due to ASR limitations. To reduce exclusionary design, we use a small, configurable gesture vocabulary and a visual feedback mechanism (LED indicators), and we will test across multiple users during verification.

Honest claims and limitations: Our system will not claim full sign-language translation. It recognizes only a predefined command set. Following IEEE/ACM ethical guidance on honest and realistic claims, we will report accuracy/latency limitations and the tested operating conditions transparently [4, 6].

3.2 Safety

Electrical hazards: The main hazard is switching real loads. If low-voltage loads are used (recommended for prototype demos), hazard is reduced but protection (fusing, current limiting, proper wire gauge, strain relief) remains necessary. We will document safe wiring practices and verify that the system fails to a safe state (outputs OFF) on reset and comms loss.

Mechanical hazards: A motorized device can pinch or catch objects. We mitigate by (i) limiting speed/torque, (ii) implementing end-of-travel protection (limit switches or bounded timer).

Development safety: During lab development we will (i) power the system from a current-limited bench supply when possible, (ii) keep high-current paths physically separated from logic circuitry, and (iii) use insulated connectors and enclosures for any exposed contacts.

References

[1] O. Masina *et al.*, "Investigating the accessibility of voice assistants with impaired users: Mixed methods study," *JMIR mHealth and uHealth*, 2020, open-access via PubMed Central, accessed 2026-03-24. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC7547392/

[2] Kendryte / Canaan Technology, "K230 product full datasheet," Developer documentation, accessed 2026-03-24. [Online]. Available: https://www.kendryte.com/k230/zh/dev/00_hardware/K230_datasheet.html

[3] Kendryte, "Introduction to k230 ai demo," Developer documentation, accessed 2026-03-24. [Online]. Available: https://www.kendryte.com/k230/en/dev/02_applications/ai_demos/K230_AI_Demo_Introduction.html

[4] IEEE, "Ieee code of ethics," IEEE website, accessed 2026-03-24. [Online]. Available: https://ieee-cas.org/about/ieee-code-ethics

[5] D. Brand *et al.*, "A survey assessing privacy concerns of smart-home services provided to individuals with disabilities," *Journal of Rehabilitation and Assistive Technologies Engineering*, 2019, open-access via PubMed Central, accessed 2026-03-24. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC7070117/

[6] Association for Computing Machinery, "Acm code of ethics and professional conduct," ACM website, accessed 2026-03-24. [Online]. Available: https://www.acm.org/code-of-ethics