ECE 445

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT

---

# Design Document for ECE 445

---

**Team #28**

CHENG ZHENG
(cz77@illinois.edu)
YUXUAN WU
(yuxuan59@illinois.edu)
ZHEWEI ZHANG
(zheweiz3@illinois.edu)
ZIYANG JIN
(ziyang3@illinois.edu)

Sponsor: Liangjing Yang

March 24, 2026

# Contents

# 1 Introduction

## 1.1 Problem Statement

Portable robotic assistants have strong potential in everyday settings, but their compact form factor often leads to a highly constrained user interface. Most palm-size robots depend on a mobile app or a small set of physical buttons, which makes interaction indirect and less intuitive. In practice, users must frequently switch between the phone and the robot to understand system status and trigger actions, increasing both learning cost and operational friction. Some solutions add external displays to improve usability, but this adds bulk and setup complexity, undermining portability.

To address these limitations, we propose a portable robotic assistant that projects an interactive interface onto any flat surface (e.g., a desk or a wall) and supports both projected "tap-like" interaction and gesture-based commands. By combining dynamic projection with real-time vision-based gesture recognition and feedback, the system aims to deliver a more natural, direct, and engaging human–robot interaction experience while preserving the core advantage of portability.

## 1.2 Solution Overview and Visual Aid

Figure 1: Design Conceptual Graph.

- **Projection Subsystem:** Uses a DLP2000 projector to render interactive UI elements (Weather, Clock, Exit, etc.) onto a desk or wall.
- **Perception & Gesture Recognition Subsystem:** Uses an OV5640 camera and MediaPipe-based pipeline to detect predefined gestures (tap, thumbs-up, thumbs-down, heart)

and interpret user commands.

- **Motion Control Subsystem:** Drives four mecanum wheels for omnidirectional movement under user command.
- **Power Management Subsystem:** Supplies and regulates power to all modules.
- **Main Control & Communication Subsystem:** Raspberry Pi Zero 2 W coordinates all subsystems, runs gesture recognition and UI rendering, and provides audio feedback via a speaker.

The system operates as follows: The robot projects a main interface onto a flat surface. The user selects an icon by pointing/tapping within the projected area, or issues a quick command via hand gesture. The camera captures the interaction, the recognition pipeline identifies the command, and the system responds by updating the projected interface, moving the robot, or playing an audio/animation feedback.

## 1.3 High-Level Requirements List

The Portable Projection Robot system shall satisfy the following high-level requirements:

1. The robot shall project an interactive user interface onto flat surfaces (e.g., a desk or wall) without relying on any external display device, while maintaining overall portability (overall dimensions $\leq 25\ \mathrm{cm} \cdot 25\ \mathrm{cm} \cdot 25\ \mathrm{cm}$, mass $\leq 2.5\ \mathrm{kg}$).
2. The system shall support real-time interaction through both projected point-and-click selection and gesture-based commands, enabling users to reliably operate the interface.
3. The main interface shall provide core navigation and information-query functions, including at minimum Weather, Clock, and Exit, and shall display current weather and time information upon user selection with a consistent return-to-main mechanism.
4. The system shall provide affective feedback for predefined gestures (thumbs-up, thumbs-down, and heart) by displaying corresponding animations and playing distinct sound cues.

# 2 Design and Requirements

## 2.1 Physical Diagram and Block Diagram:

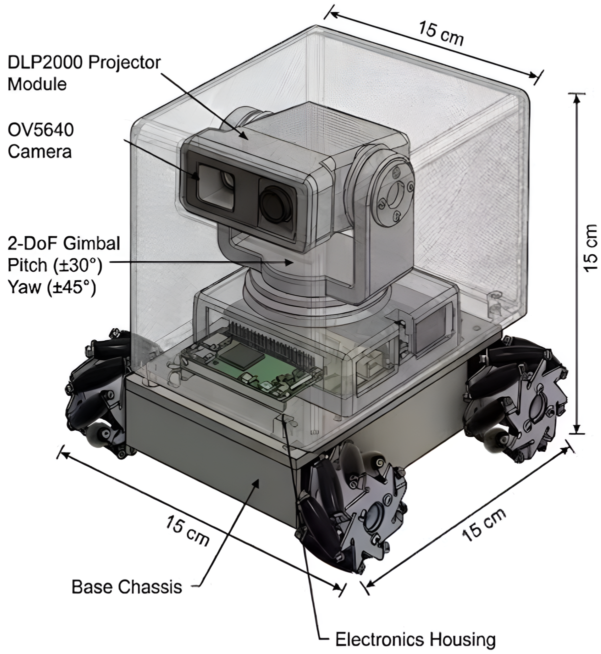

Figure 2: Physical Overview of the Palm-Size Robotic Assistant.

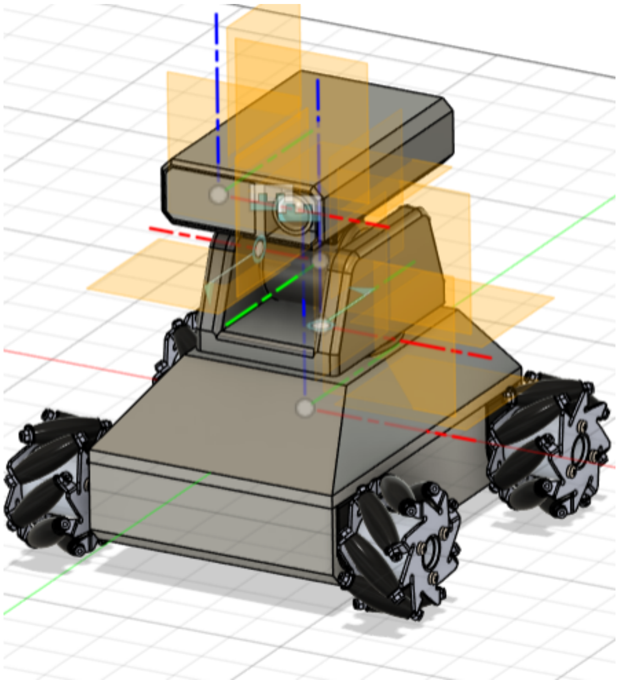

Figure 3: Physical Layout and Major Components.

Table 1: Bill of Materials (BOM)

| No. | Part Name | Qty | Description |
|---|---|---|---|
| 1 | Rectangular Base Block | 1 | – |
| 2 | Mecanum Wheel Assembly | 4 | Wheel diameter: 5 cm |
| 3 | 2-DoF Gimbal | 1 | – |
| 4 | DLP2000 Projector Module | 1 | – |
| 5 | OV5640 Camera Module | 1 | – |
| 6 | Raspberry Pi Zero 2 W | 1 | – |
| 7 | Power Management Module | 1 | – |
| 8 | Speaker Module | 1 | – |
| 9 | 3D-Printed Enclosure Assembly | 1 | Overall size: 25 cm × 25 cm × 25 cm |
| 10 | Misc. Connectors / Screws | 1 set | Fasteners, wiring, and adapters |

The Portable Projection Robot consists of four subsystems. The main control unit (Raspberry Pi Zero 2 W) coordinates all modules. The projector receives rendered frames via

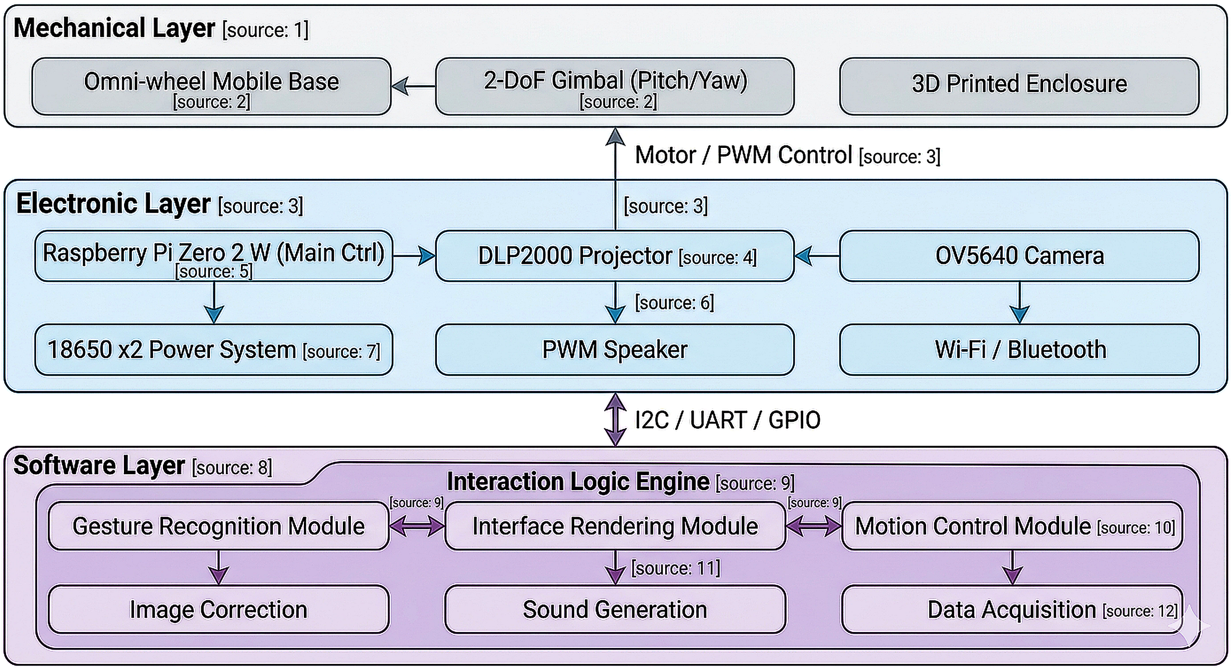

Figure 4: System Block Diagram.

HDMI. The camera captures gestures, which are processed by the gesture recognition pipeline. Motion commands are sent to the motor drivers for omnidirectional movement. A speaker provides audio feedback. Power is distributed from a battery through a power management module.

## 2.2 Projection Subsystem

The projection subsystem is responsible for rendering the interactive user interface onto a flat surface. It uses a DLP2000 projector module driven by the Raspberry Pi Zero 2 W via HDMI.

### 2.2.1 Hardware Support

Table 2: Key Hardware Components

| **Component** | **Specification** | **Interface** |
|---|---|---|
| DLP2000 Projector | $854 \times 480$ resolution, throw ratio $\sim 0.67$, 20–50 cm focus range | HDMI |
| Raspberry Pi Zero 2 W | 1 GHz quad-core CPU, 512 MB RAM | GPIO/HDMI |

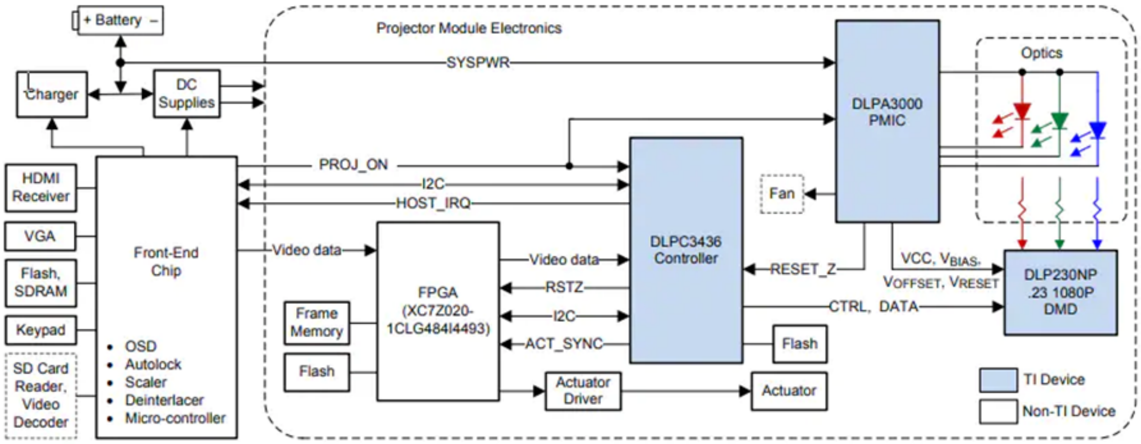

Figure 5: Projection Subsystem Block Diagram.

Table 3: Projection Requirements

| ID | Requirement | Verification Method |
|---|---|---|
| P1 | Projection resolution $\geq 854 \times 480$ | Visual inspection plus resolution test pattern |
| P2 | Maintain clear focus within 20–50 cm distance | Visual inspection at multiple distances |
| P3 | Icon size $\geq 3$ cm $\times$ 3 cm at 40 cm distance | Measure projected UI using a ruler |

### 2.2.2 Interface Rendering

The UI is rendered using a lightweight graphics library (e.g., Pygame or OpenGL ES). The main interface contains three icons (Weather, Clock, Exit). Upon selection, the corresponding information page is displayed. A ”back” button returns to the main interface.

Table 4: Interface Function Requirements

| ID | Requirement | Verification Method |
|---|---|---|
| P4 | UI page switching response time $\leq 2$ s | Frame-by-frame video analysis |
| P5 | Consistent return-to-main mechanism | Functional test |

## 2.3 Perception & Gesture Recognition Subsystem

The perception subsystem captures user gestures via an OV5640 camera and processes them using Media Pipe to recognize predefined commands.

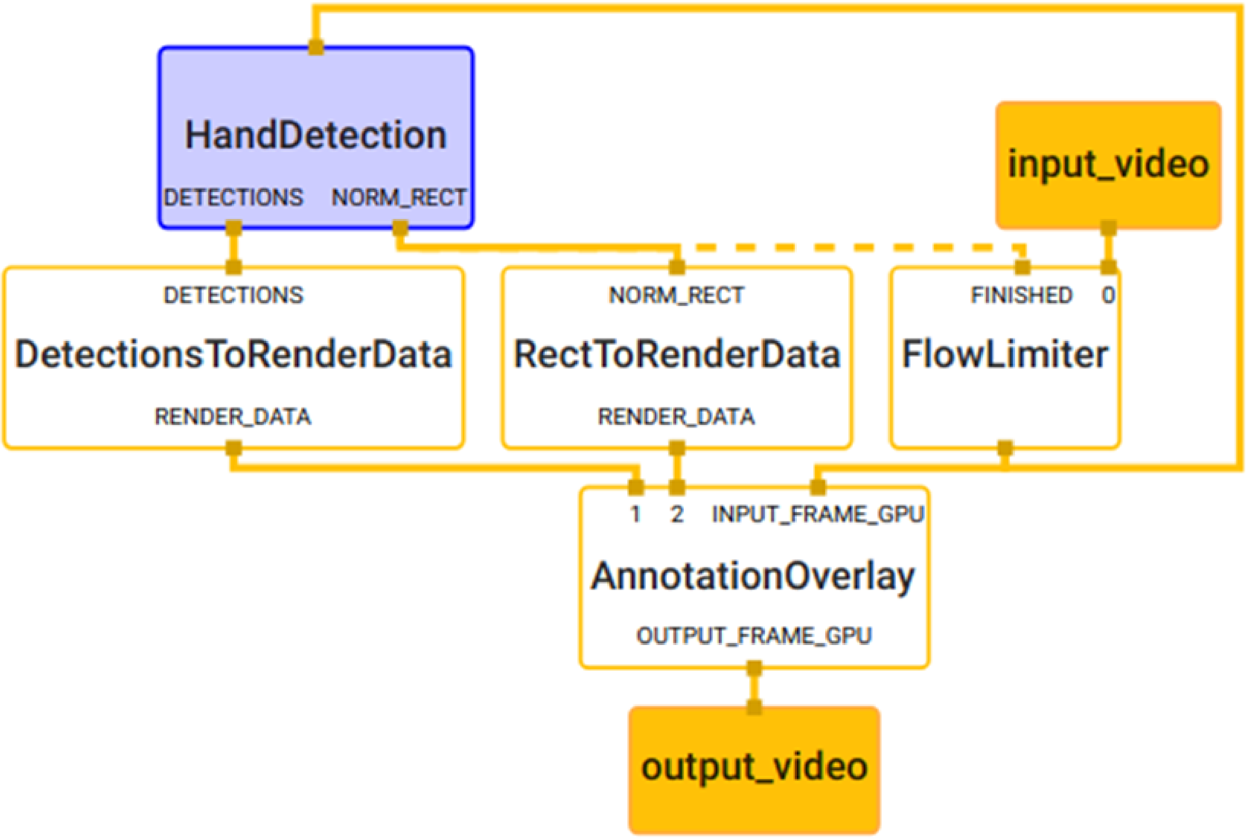

Figure 6: Perception & Gesture Recognition Subsystem Block Diagram.

### 2.3.1 Hardware Support

Table 5: Gesture Recognition Hardware Components

| Component | Specification | Interface |
|---|---|---|
| OV5640 Camera | 5 MP, $2592 \times 1944$, supports 30 fps at lower resolutions | USB/CSI |
| Raspberry Pi Zero 2 W | Vision processing | Onboard |

Table 6: Gesture Recognition Requirements

| ID | Requirement | Verification Method |
|---|---|---|
| G1 | Gesture recognition processing frame rate $\geq$ 15 fps | Log and report frame rate in software |
| G2 | Reliable gesture recognition within 30–60 cm distance | Test recognition success rate at multiple distances |
| G3 | Supports four gesture types: point-and-click, thumbs-up, thumbs-down, heart | Functional test case coverage |

### 2.3.2 Gesture Recognition Pipeline

The system uses MediaPipe Hands for hand landmark detection and a lightweight classifier (e.g., logistic regression or rule-based) to map hand landmarks to gestures.

- Point-and-click: Index finger extended, tip position relative to projected icon coordinates.
- Thumbs-up: Thumb extended upward, other fingers curled.
- Thumbs-down: Thumb extended downward.
- Heart: Both hands or fingers forming a heart shape (or index/middle fingers crossed).

The camera coordinate system is calibrated to the projection coordinate system via a homography transformation (see Section 2.8.1).

### 2.3.3 Projection-Camera Alignment

To ensure accurate mapping between the projected UI and the camera's view, a homography calibration is performed. A calibration pattern (e.g., chessboard) is projected, and the camera captures it. OpenCV computes the homography matrix $\mathbf{H}$ that maps projection coordinates to camera pixel coordinates.

After calibration, the reprojection error $e_{\text{rep}}$ is measured. The design tolerance requires:

$$e_{\text{rep}} \leq \alpha \cdot n_{\text{icon}} \tag{1}$$

where $n_{\text{icon}}$ is the icon width in pixels and $\alpha = 0.2$ (20% of icon width). With a minimum icon size of $3\,\text{cm} \times 3\,\text{cm}$ corresponding to approximately $42 \times 42$ pixels at 40 cm distance, a 5-pixel reprojection error constitutes roughly 12% of the icon width, which falls within the design tolerance.

## 2.4 Motion Control Subsystem

The motion control subsystem enables the robot to move under user commands. It consists of four mecanum wheels, each driven by a DC motor with an encoder and a motor driver.

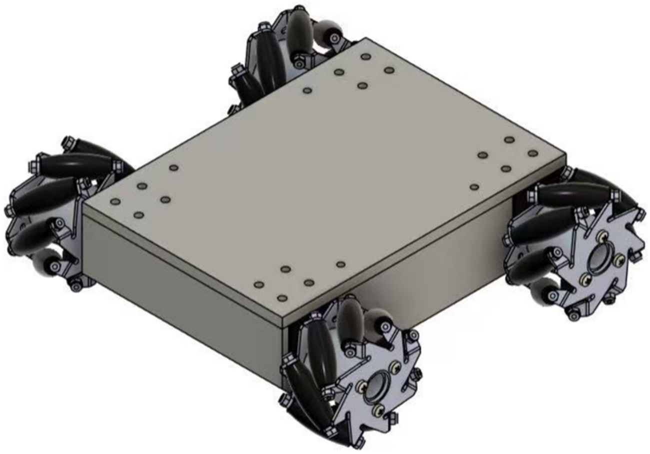

Figure 7: Mecanum Wheel Configuration Diagram.

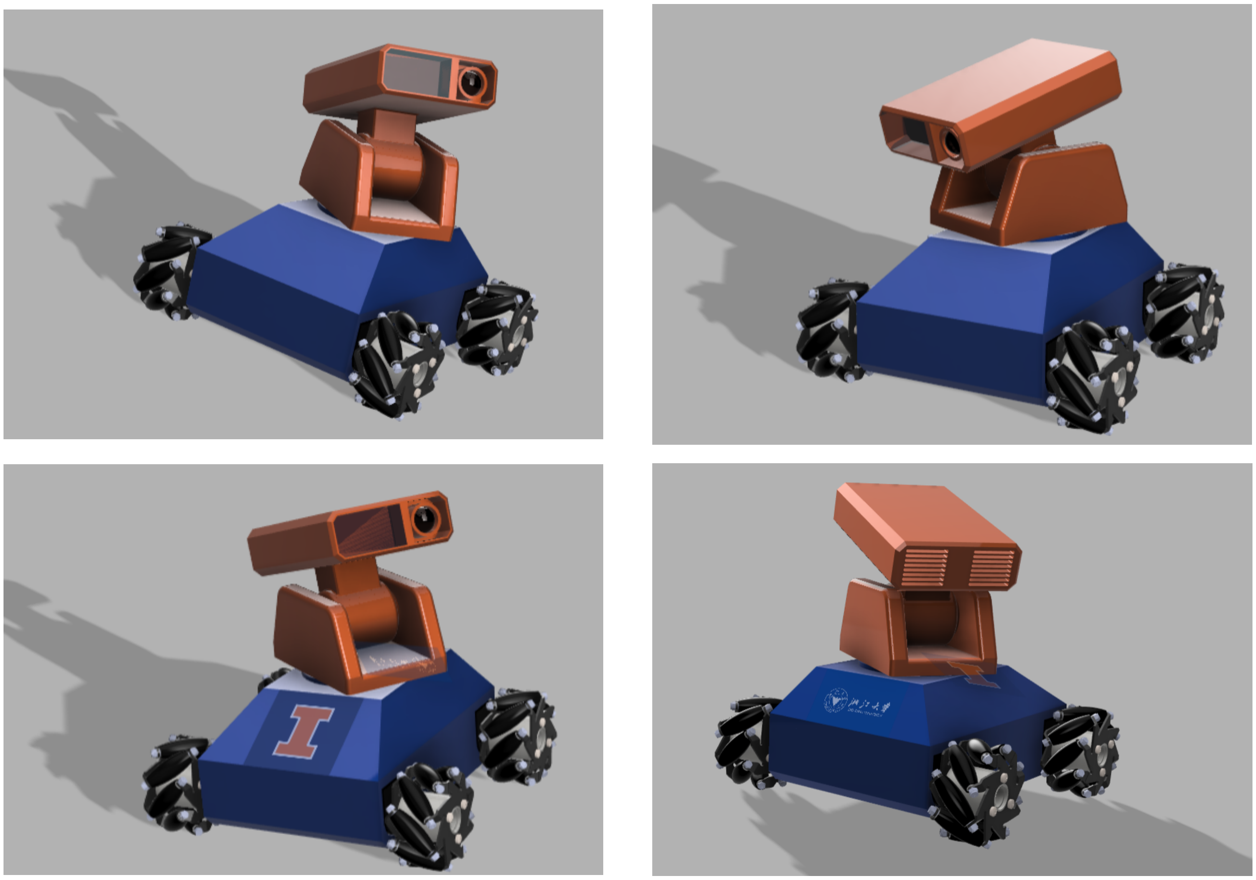

Figure 8: 3D CAD Model of the Portable Interactive Robot

Table 7: Mobility Requirements

| ID | Requirement | Verification Method |
|---|---|---|
| M1 | Maximum linear speed $\geq 0.1$ m/s | Mark a known distance on desk and measure travel time |
| M2 | Positioning accuracy $\leq$ $\pm 2$ cm (open-loop control) | Command fixed travel distance and measure actual displacement |
| M3 | Motion command response latency $\leq$ $500$ ms (from trigger to base movement start) | Frame-by-frame video analysis |

## 2.5 Power Management Subsystem

The power management subsystem supplies stable voltage to all modules. A rechargeable Li-ion battery (7.4V, 3000 mAh) is used. Voltage regulation provides 5V for Raspberry Pi, camera, and speaker, and 3.3V for logic.

Total estimated peak current: $\sim 4.0$ A. Battery capacity: 3000 mAh → estimated runtime $\approx 45$ min under peak load.

Table 8: Power Budget of Major Components

| Component | Voltage | Current (max) |
|---|---|---|
| Raspberry Pi Zero 2 W | 5V | 1.2A |
| DLP2000 Projector | 5V | 1.5A |
| OV5640 Camera | 5V | 0.2A |
| Motors (4×) | 5V | 0.5A each |
| Speaker | 5V | 0.2A |

## 2.6 Main Control & Communication Subsystem

The Raspberry Pi Zero 2 W serves as the central processing unit. It runs:

- Gesture recognition pipeline (MediaPipe)
- UI rendering engine
- Motor control logic (via GPIO/PWM)
- Audio feedback (via GPIO or USB audio)

Table 9: Main Control and Communication Requirements

| ID | Requirement | Verification Method |
|---|---|---|
| C1 | System boot time $\leq 30$ s to ready state | Measure from power-on to UI projection |
| C2 | Concurrent execution of vision and rendering without blocking | Stress test with maximum frame rate |

## 2.7 Schematics

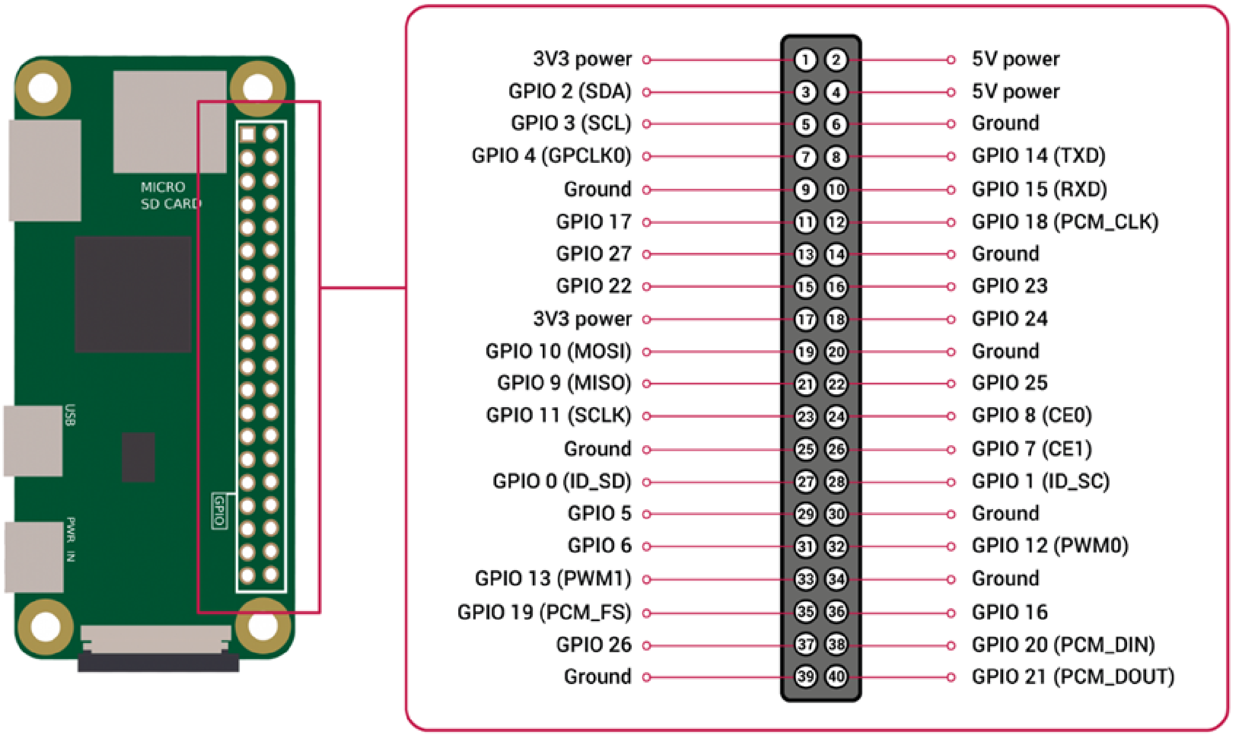

Figure 9: Raspberry Pi Zero 2 W GPIO Pinout Diagram.

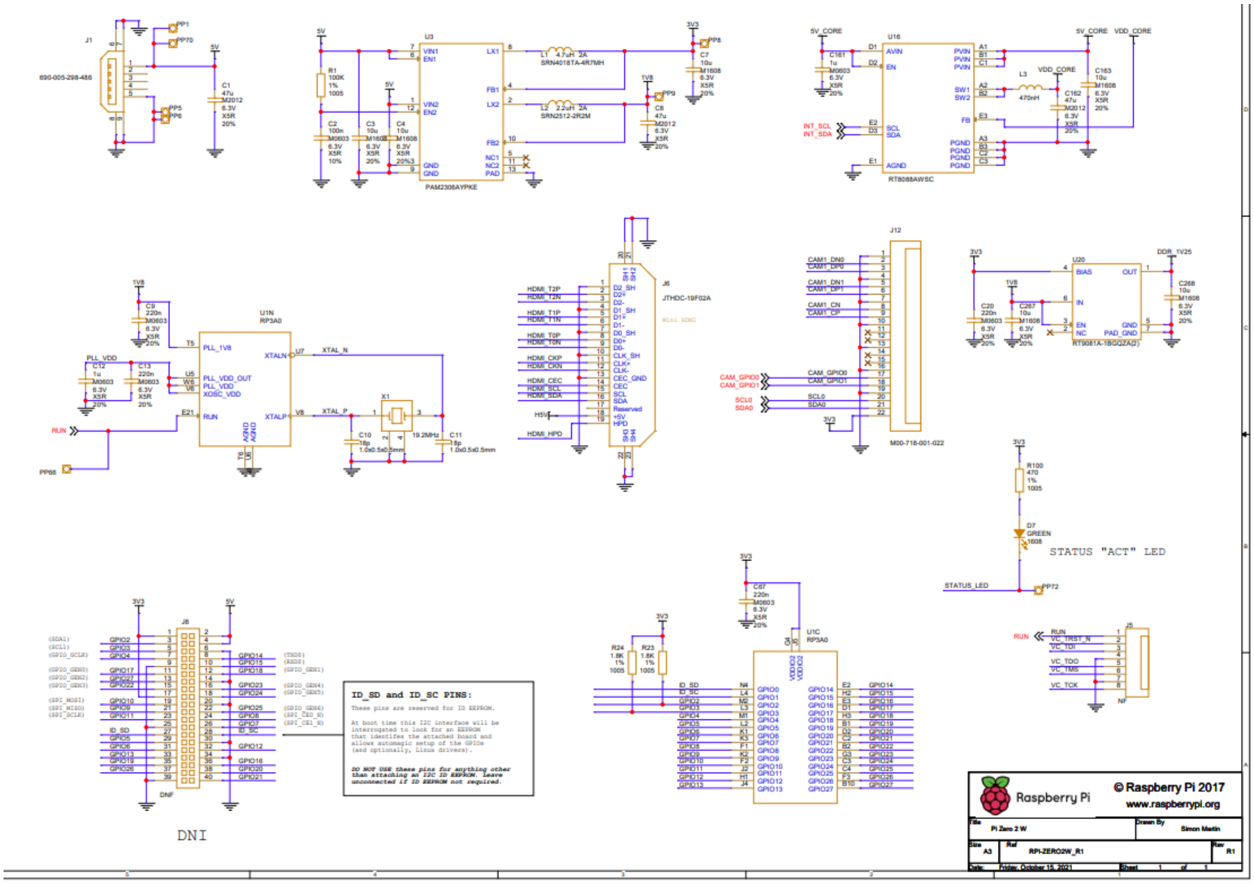

Figure 10: Power Distribution Schematic

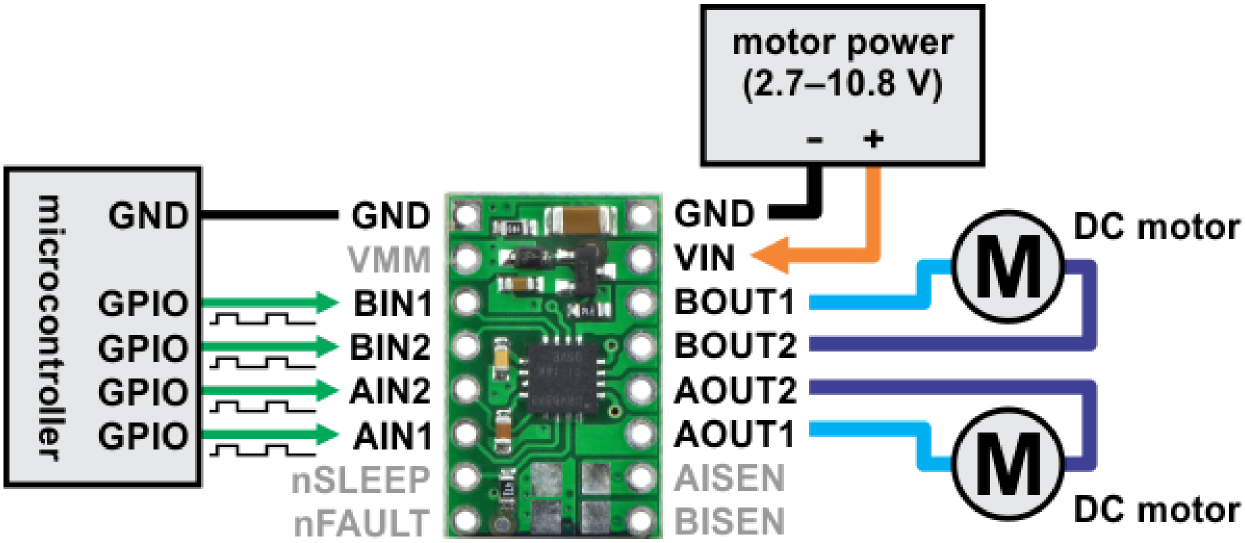

Figure 11: Motor Driver Connection Diagram.

## 2.8 Tolerance Analysis

The most critical subsystem in this project is the projection-vision closed loop, which encompasses camera capture, gesture recognition, and projected interface update. This closed loop directly determines the responsiveness and reliability of the user interaction and poses the greatest technical risk.

### 2.8.1 Projection-Camera Alignment Tolerance

A critical design parameter is the alignment between the projector's output and the camera's field of view. The projector and camera are mounted side by side with an optical center separation distance $d \leq 5$ cm. Under typical operation with projection distance $L = 40$ cm, the angular offset is:

$$\theta = \arctan\left(\frac{d}{L}\right) = \arctan\left(\frac{5}{40}\right) \approx 7.1^\circ \tag{2}$$

This offset is fully compensable through homography calibration. After calibration, the reprojection error between the projector coordinate system and the camera image coordinate system shall not exceed 5 pixels in the camera frame. Given that the minimum icon size of $3$ cm $\times$ $3$ cm corresponds to approximately $42 \times 42$ pixels in the projected image (based on 854 pixels spanning 60 cm at 40 cm distance), a 5-pixel reprojection error constitutes roughly 12% of the icon width, which falls within the design tolerance of 20% of the icon size.

### 2.8.2 End-to-End Latency Budget

To satisfy the success criterion that all gesture-triggered responses occur within 2 seconds, a detailed latency budget is established:

Table 10: End-to-End Latency Budget

| **Stage** | **Component** | **Estimated Latency** |
|---|---|---|
| $T_{\text{cap}}$ | Camera frame acquisition | 50 ms (30 fps capture + overhead) |
| $T_{\text{vis}}$ | Gesture recognition (MediaPipe + classifier) | 100 ms |
| $T_{\text{logic}}$ | Interaction logic & state machine | 50 ms |
| $T_{\text{render}}$ | Interface rendering & HDMI transmission | 100 ms |
| Total | End-to-end latency | 300 ms |

$$T_{\text{e2e}} = T_{\text{cap}} + T_{\text{vis}} + T_{\text{logic}} + T_{\text{render}} = 300 \text{ ms} \tag{3}$$

This is substantially below the 2-second success threshold, providing a comfortable design margin.

Mitigation Strategy: If computational limitations on the Raspberry Pi Zero 2 W cause excessive latency, the gesture recognition frame rate can be reduced from 30 fps to 15 fps as a controlled trade-off.

# 3 Cost

Table 11: Estimated Project Cost

| **Component** | **Quantity** | **Unit Price (CNY)** |
|---|---|---|
| Rectangular Base Block | 1 | 35.00 |
| Mecanum Wheel Assembly | 4 | 18.90 |
| 2-DoF Gimbal | 1 | 109.50 |
| DLP2000 Projector Module | 1 | 1162.69 |
| OV5640 Camera Module | 1 | 106.05 |
| Raspberry Pi Zero 2 W | 1 | 285.83 |
| Power Management Module | 1 | 25.00 |
| Speaker Module | 1 | 15.00 |
| 3D-Printed Enclosure Assembly | 1 | 120.00 |
| Misc. Connectors / Screws | 1 set | 25.00 |
| **Total** | | **1959.67** |

# 4 Schedule

Table 12: Project Schedule and Task Assignment

| **Week** | **Zhewei Zhang** | **Yuxuan Wu** | **Cheng Zheng** | **Ziyang Jin** |
|---|---|---|---|---|
| 3/25 | Define mechanical layout and component placement | Define motion-control framework for the mobile base | Analyze module power needs and overall electrical architecture | Design software framework for gesture recognition and UI logic |
| 4/1 | Start 3D CAD modeling of enclosure and mounts | Develop basic motion control for the mecanum base | Design circuit connections among major hardware modules | Develop the first version of the projected UI |
| 4/8 | Continue enclosure design and refine dimensions | Continue motion-control development and wheel-drive testing | Continue circuit implementation and power verification | Continue UI development and implement page switching |
| 4/15 | Finalize mechanical model and support prototype assembly | Integrate motion control with software commands | Complete hardware wiring and debug electrical interfaces | Develop and test gesture recognition for target gestures |
| 4/22 | Adjust camera-projector mounting and physical alignment | Continue motion debugging and latency testing | Continue hardware integration and troubleshoot communication issues | Continue gesture integration with visual and audio feedback |
| 4/29 | Refine enclosure and support full prototype packaging | Conduct integrated motion testing and control tuning | Measure projector, camera, and power subsystem performance | Optimize recognition robustness and software speed |
| 5/6 | Verify mechanical requirements and assembly quality | Verify mobility requirements and positioning performance | Verify electronic requirements and summarize test results | Verify software requirements and interaction performance |
| 5/12 | Prepare final presentation on mechanical design | Prepare final presentation on control and motion results | Prepare final presentation on circuit and hardware integration | Prepare final presentation on software and gesture interaction |

# 5 Ethics and Safety

## 5.1 Ethics

This project involves human-computer interaction, gesture recognition, and projection-based interface technology, thus requiring consideration of several ethical issues in accordance with the ACM Code of Ethics and Professional Conduct and the IEEE Code of Ethics, especially the principles concerning social well-being, avoiding harm, fairness, honesty, privacy, and confidentiality [1].

**Privacy:** The camera is used only for real-time gesture recognition. No image data is stored, transmitted, or shared without user consent. Users are clearly informed when the system is performing gesture recognition. This is consistent with ACM principles on respecting privacy, using personal information only for legitimate ends, minimizing unnecessary data collection, and establishing transparent policies for data use and informed consent, as well as the IEEE requirement to protect the privacy of others [1].

**Fairness and Accessibility:** The system adopts simple and intuitive interaction methods and avoids relying on overly complex gestures, helping users with varying action habits and physical conditions use it conveniently. This aligns with the ACM principle of being fair, taking action not to discriminate, and designing technologies to be inclusive and accessible, as well as the IEEE principle of treating all persons fairly and with respect and avoiding discrimination [1].

**Transparency:** The system does not exaggerate its intelligence level. Descriptions of system functions remain truthful and clear. This is consistent with the ACM requirement that computing professionals be honest and trustworthy, and provide transparent disclosure of system capabilities, limitations, and potential problems, as well as the IEEE requirement to be honest and realistic in stating claims or estimates based on available data [1][2].

**Social Impact:** The device may be used in shared spaces such as classrooms or offices. Attention is paid to avoiding invasion of others' privacy or unnecessary interference, and to considering the broader social effects of the system in real deployment contexts. This is in line with ACM principles emphasizing contribution to human well-being, avoidance of harm, and the public good as a central concern in computing work, and with the IEEE principle of improving public understanding of the capabilities and societal implications of technologies [1], [2].

This project therefore follows the ethical guidance advocated by the ACM Code of Ethics and Professional Conduct and the IEEE Code of Ethics [1], [2].

## 5.2 Safety

This project also considers safety issues during design, implementation, and testing. In line with the ACM Code of Ethics and the IEEE Code of Ethics, the system should avoid harm, evaluate possible risks, protect the privacy and welfare of the public, and be designed and implemented in a robust and usable manner [1], [2].

**Electrical Safety:** The system consists of a power module, control circuit, sensors, and projection equipment. All circuit connections are correct, the power supply operates within the rated range, and necessary insulation protection measures are taken.

**Optical Safety:** Projection brightness and direction are controlled to prevent direct strong light from shining into the user's eyes, reducing eye discomfort.

**Mechanical Safety:** The projection module, camera, and associated components are securely affixed to prevent dislodgement, falling, or instability.

**Thermal Safety:** Processors, projection modules, and power components generate heat during operation. Basic heat dissipation conditions are ensured to prevent excessive temperature.

**Laboratory Safety:** During development and testing phases, laboratory safety regulations are followed. Tools, power sources, and soldering equipment are used with caution, and experiments are conducted in as controlled an environment as possible.

This project does not involve high-risk medical or invasive functions. Overall, with basic engineering safety measures in place, the safety risks are expected to remain low and controllable; this is consistent with ACM guidance on avoiding harm, performing thorough risk evaluation, and designing systems that are robustly and usably secure, as well as the IEEE requirement to hold paramount the safety, health, and welfare of the public and to disclose factors that might endanger the public or the environment [1], [2].

# References

[1] Association for Computing Machinery. “Acm code of ethics and professional conduct.” Accessed: Apr. 2, 2026, Accessed: Apr. 2, 2026. [Online]. Available: https://www.acm.org/code-of-ethics.

[2] IEEE. “Ieee code of ethics.” Accessed: Apr. 2, 2026, Accessed: Apr. 2, 2026. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html.