ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Autonomous Indoor Patrol Robot for Cat Deterrence

Team #13

CHENTAO FANG
(chentao3@illinois.edu)
JIAWEI KONG
(jiaweik2@illinois.edu)
RONGLONG LIU
(rl27@illinois.edu)
YANCHEN LIU
(yl89@illinois.edu)

FACULTY ADVISOR: YU LIN
TA: YUJIE GONG

May 14, 2026

Abstract

This report presents the design, implementation, and verification of the Autonomous Indoor Patrol Robot for Cat Deterrence. The system is designed for an indoor dining-table-scale environment, where it patrols along a predefined path, detects cat-like targets using embedded vision, aims toward the detected target with a two-axis gimbal, and activates a water-blast deterrence mechanism when appropriate. The final prototype integrates line-following chassis patrol, MaixCAM-based object detection, Arduino Mega 2560 control, stepper-motor gimbal actuation, a 12 V water pump and nozzle assembly, and a custom power distribution printed circuit board.

The project focuses on reliable subsystem integration across sensing, control, actuation, and power delivery. The MaixCAM vision module provides target information at up to 40 frames per second, and the Arduino Mega 2560 uses this information to control the gimbal and coordinate the pump and chassis behaviors. The power system distributes separate voltage rails for the controller, vision module, pump, stepper drivers, and chassis motors while maintaining a common ground reference. Verification testing was performed at both the subsystem and integrated-system levels, including power output validation, line-following patrol testing, vision-to-controller communication, gimbal tracking response, pump range testing, and full-system operation.

The completed prototype demonstrates the main required functions of autonomous indoor patrol and cat deterrence. The robot can patrol using line-following control, detect targets through the MaixCAM vision system, aim the water-blast mechanism through the stepper-driven gimbal, and deliver a water stream with a range of at least 1.5 m. Although limitations remain in detection robustness, long-term reliability, and mechanical packaging, the project validates the feasibility of combining embedded vision, mobile patrol, stepper-based aiming, and controlled water actuation in a compact indoor deterrence robot.

Contents

1 Introduction

1.1 Problem Statement and Objective

Domestic cats often enter indoor areas where their owners prefer to keep them away, such as dining tables, kitchen counters, or food preparation surfaces. Continuous manual supervision is not always practical, so this project aims to develop an autonomous and non-harmful deterrence system for a small indoor environment. The final project, titled *Autonomous Indoor Patrol Robot for Cat Deterrence*, is designed for an indoor dining-table-scale environment. Its main objective is to patrol along a predefined path, detect cat-like targets, aim toward the detected target, and deliver a controlled water stream as a deterrent.

The system is intended to provide localized deterrence without direct physical contact. Instead of using a fixed device, the robot combines line-following patrol, embedded vision, directional aiming, and water-blast actuation. Because the robot operates near people, pets, and household objects, the design emphasizes controlled operation, electrical reliability, and safe actuation. This design priority is consistent with the engineering responsibility emphasized by the IEEE Code of Ethics [1].

1.2 System Overview

The system-level design is shown in Figure 1. The robot is divided into five major modules: control, sensing, actuation, drive, and power. The control module uses an Arduino Mega 2560 to coordinate target tracking, pump activation, chassis motion, and user input. The Arduino Mega 2560 was selected because it provides sufficient digital input/output pins and multiple hardware serial interfaces for integrating the vision module, motor drivers, stepper drivers, and other peripherals [2].

The sensing module uses a MaixCAM Lite with a built-in camera to detect cat-like targets and send target information to the Arduino through universal asynchronous receiver-transmitter (UART) communication. The MaixCAM platform supports embedded vision and onboard artificial intelligence applications, making it suitable for compact target-detection tasks [3]. The actuation module consists of a two-axis stepper-motor gimbal and a water pump. The gimbal receives pulse/direction (PUL/DIR) control signals to aim the nozzle, while the pump provides the water-blast output. The drive module uses motor drivers and four TT motors for line-following patrol. The power module contains the batteries, voltage regulation, protection, and distribution circuitry required by the subsystems.

1.3 Performance Requirements and Design Evolution

The main performance requirements are based on the robot's ability to patrol, detect, aim, and deter within the intended indoor environment. The chassis must follow a predefined patrol path. The vision system must detect cat-like targets and provide target position

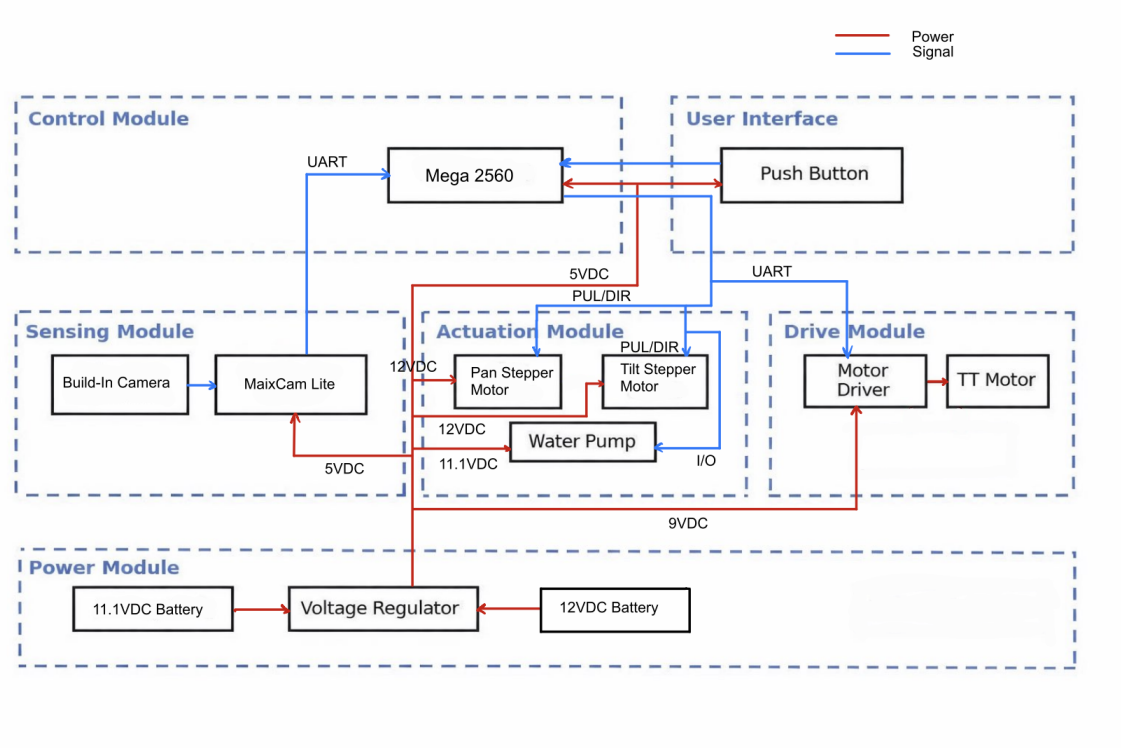

Figure 1: System block diagram of the autonomous indoor patrol robot for cat deterrence.

information at a sufficient update rate for real-time aiming. The MaixCAM Lite can output target information at up to 40 frames per second. The water-blast mechanism must achieve an effective range of at least 1.5 m. The power system must provide stable voltage rails for the Arduino Mega 2560, MaixCAM Lite, stepper drivers, pump driver, and chassis motor drivers while maintaining a common ground reference. The integrated system must also support a simple user-start procedure.

The design changed significantly during the semester. The original outdoor use case was narrowed to an indoor dining-table-scale environment to improve feasibility and reduce environmental uncertainty. The vision approach was changed to the MaixCAM Lite instead of a larger single-board computer. The aiming mechanism was changed from servo-based actuation to stepper-motor actuation for faster and more repeatable positioning. The electrical system was also reorganized around a custom power distribution printed circuit board (PCB) to simplify wiring and improve integration. The final prototype integrates line-following patrol, vision-based detection, stepper-based aiming, water-pump activation, and multi-rail power distribution.

Several factors determine system performance: detection robustness, UART communication reliability, gimbal response, pump range, chassis line-following stability, and power stability. These factors guided the design and verification work described in the following sections.

2 Design

2.1 Design Overview

The robot was designed as an integrated mobile platform combining line-following patrol, embedded vision, stepper-motor aiming, water-blast actuation, and custom power distribution. The final system consists of a MaixCAM Lite sensing module, an Arduino Mega 2560 controller, a two-axis stepper-motor gimbal, a water pump and nozzle assembly, a four-motor chassis, and a custom PCB for power distribution and signal organization.

The design is divided into five functional blocks. The sensing system detects targets and sends target coordinates to the controller. The control system reads target data, line-following sensors, and system states, then commands the drive, gimbal, and pump subsystems. The actuation system includes the pan-tilt gimbal and water-blast mechanism. The drive system provides line-following patrol using four TT motors. The power system accepts two battery inputs and distributes the required voltage rails to the electronic and electromechanical loads.

At the system level, two major architecture choices were considered: a fixed deterrent station and a mobile patrol robot. A fixed station would simplify the mechanical and drive design, but it would only cover a limited direction and area unless multiple units were installed. The mobile architecture was chosen because it can patrol a predefined indoor path and bring the sensing and deterrence mechanism closer to different parts of the operating area. Since the final environment is a dining-table-scale indoor space, the added drive complexity remains manageable while improving coverage and demonstration value.

2.2 Control System

The main controller is an Arduino Mega 2560. It was selected because the robot requires multiple digital outputs, digital inputs, and serial interfaces. The MaixCAM Lite communicates with the Arduino through UART. The stepper drivers require pulse, direction, and enable signals. The pump driver requires a digital control signal, while the line-following sensors require digital inputs. The final integrated software uses the pin assignment listed in Table 1. Since the pump logic is still under final tuning, the pump behavior may be adjusted during final debugging.

Alternative controller options included an Arduino Uno, a Raspberry Pi, and a vision module acting as the main controller. The Arduino Uno was not selected because the system requires multiple serial and digital interfaces at the same time. A Raspberry Pi would provide more computing power, but it would increase boot time, software complexity, and power-management requirements for a task that is mainly deterministic motor and pump control. The final design separates responsibilities: the MaixCAM Lite performs vision processing, while the Arduino Mega 2560 handles real-time control and subsystem coordination. This choice keeps the controller software simpler and makes the wiring and

debugging process more predictable.

Table 1: Arduino Mega 2560 signal pin assignment in the current integrated code

Signal	Arduino Pin	Function
MAIX_TX	D19 / RX1	Receive MaixCAM target data
Motor command	D16 / TX2	Send commands to chassis motor driver
X_EN	D5	Pan stepper enable
X_STP	D6	Pan stepper pulse
X_DIR	D7	Pan stepper direction
Y_EN	D8	Tilt stepper enable
Y_STP	D9	Tilt stepper pulse
Y_DIR	D10	Tilt stepper direction
PUMP	D30	Pump driver control
LEFT_SENSOR	D32	Left line sensor input
RIGHT_SENSOR	D33	Right line sensor input

The control logic is organized around three behaviors: patrol, tracking, and spray activation. During patrol, the robot follows a predefined line path. When valid target data are received, the chassis stops and the gimbal tracks the target. If the target remains aligned for a defined time interval, the controller enters the spray state and activates the pump driver.

2.3 Sensing System

The sensing system uses a MaixCAM Lite with a built-in camera. Alternative vision platforms considered included a Raspberry Pi camera system and an OpenMV-style embedded camera. A Raspberry Pi would provide stronger general-purpose computing capability and a larger software ecosystem, but it would also require a separate camera, operating-system setup, longer startup time, and more complex power handling. OpenMV-type modules are compact and easy to program, but their onboard computing capability is more limited for object-detection models. The MaixCAM Lite was selected because it provides a compact camera and onboard vision-processing platform that can output only the target information needed by the Arduino. This allows the Arduino to focus on control instead of image processing.

The MaixCAM Lite detects cat-like targets and sends structured target messages to the Arduino through one-way UART communication. The message format is

```
TARGET <flag> <cx> <cy>
```

where `flag` indicates target detection, and `cx` and `cy` are the target center coordinates. For example, `TARGET 1 260 180` indicates that a target was detected at pixel coordinate $(260, 180)$. The current frame size is 448 by 448 pixels, so the aiming reference point is $(224, 224)$. The MaixCAM Lite can output target information at up to 25 frames per second, which is sufficient for the indoor dining-table-scale operating environment.

2.4 Actuation System

2.4.1 Stepper-Motor Gimbal

The aiming mechanism uses a two-axis gimbal driven by two 42 stepper motors. One motor controls the pan axis, and the other controls the tilt axis. A servo-based gimbal was considered because servos are simple to control and commonly used for small pan-tilt mechanisms. However, the servo option was not selected because the water nozzle and tubing create a larger and less balanced load than a small camera-only gimbal, and the aiming motion needs to be fast and repeatable. Stepper motors were selected because they provide fast response, repeatable positioning, higher torque margin, and direct pulse-based control. Each motor is driven by an independent stepper driver controlled by pulse/direction/enable signals from the Arduino.

The gimbal controller uses image-coordinate feedback from the MaixCAM Lite. The horizontal and vertical pixel errors are

$$e_x = x_{\text{target}} - x_{\text{center}} \tag{1}$$

$$e_y = y_{\text{target}} - y_{\text{center}} \tag{2}$$

where the current image center is $(224, 224)$. In the current implementation, the pan axis uses proportional-derivative control,

$$u_x = K_{p,x} e_x + K_{i,x} \sum e_x + K_{d,x}(e_x - e_{x,\text{prev}}) \tag{3}$$

with $K_{i,x} = 0$, so it behaves as a proportional-derivative controller. The tilt axis uses proportional control,

$$u_y = K_{p,y} e_y \tag{4}$$

The resulting commands are converted into limited step pulses. A tolerance window around the image center reduces jitter, and software limits prevent the gimbal from exceeding its useful mechanical range. In the current implementation, 3200 steps correspond to one full revolution, and the tilt axis is limited to approximately ± 800 steps, or about $\pm 90^\circ$.

2.4.2 Water-Blast Mechanism

The deterrence output is produced by a 12 V water pump and nozzle assembly. Several deterrence methods were considered, including sound, light, air flow, and water. Sound and light were easier to implement electrically, but early evaluation indicated that they were not strong or consistent enough as a deterrent for the intended use case. Air flow was less messy than water but required a larger fan or blower to produce a noticeable effect at distance. The water-blast mechanism was selected because it provides a clear, directional, and non-contact deterrent effect that can be aimed at the detected target. The nozzle concentrates the water output into a stream, with a required effective range of at least 1.5 m.

The pump is controlled through a commercial driver board rather than directly from the Arduino, because the pump requires more current than an Arduino output pin can supply. The driver board functions as a metal-oxide-semiconductor field-effect transistor (MOSFET) switching stage controlled by an Arduino digital output. The pump is supplied from the 11.1 V rail through the power distribution PCB. The pump is activated only during the deterrence behavior to improve safety and reduce unnecessary water use. In the current integrated code, the spray state uses a fixed spray duration of 1.5 s before returning to tracking. This pump logic is treated as the current integration baseline and may be tuned further.

2.5 Drive System

The drive system uses four TT motors controlled through motor driver modules. Alternative mobility approaches included a stationary platform, free-driving navigation, table-edge detection, and line-following patrol. A stationary platform was simpler but had limited coverage. Free-driving navigation would require more complex sensing and localization, which was not necessary for the constrained indoor environment. Table-edge detection was also considered, but relying on table edges would require more robust edge sensors and would introduce safety concerns if the robot operated close to a drop-off. Line-following patrol was selected because it is simple, repeatable, easy to test, and appropriate for a dining-table-scale area. It also creates a known patrol path, which helps isolate and verify the sensing, gimbal, and pump behaviors during integration.

The motors are not driven directly by the Arduino because of their current requirements. Although four motors are used, the PCB provides one 9 V output to the motor driver chain because two motor driver modules are connected in a cascaded configuration.

Line-following patrol is implemented with a four-channel infrared sensor module, but only the two middle channels are used in the final control scheme. The sensor outputs are digital: 0 on the black line and 1 on the surrounding non-black region. The Arduino reads the left and right sensor values and adjusts the chassis motion to keep the robot on the predefined patrol path.

2.6 Power Distribution and PCB Design

The power system uses two battery inputs and a custom power distribution PCB. Alternative wiring approaches included direct jumper-wire distribution, a breadboard-based power bus, and a custom PCB. Jumper wires and breadboards are useful for early prototyping, but they are not reliable for a robot with multiple high-current loads, including motors, stepper drivers, and a pump. Loose wiring also makes debugging difficult and increases the chance of incorrect connections. The custom PCB was selected to centralize battery input, fuse protection, main power switching, voltage regulation, power indicators, test points, and output connectors. This reduces loose wiring and improves reliability during integration.

The board accepts an 11.1 VDC battery input and a 12 VDC battery input. Each input passes through a 5 A fuse. A double-pole single-throw (DPST) main power switch connector allows both rails to be switched simultaneously. The two battery systems share a common ground reference, which is required for reliable communication between the Arduino, MaixCAM Lite, sensors, motor drivers, stepper drivers, and pump driver.

The PCB output rails are summarized in Table 2. The 5 V rail powers the Arduino Mega 2560 and MaixCAM Lite, with one additional backup output. The 9 V rail powers the chassis motor driver chain. The 11.1 V rail powers the pump driver, and two 12 V outputs power the pan and tilt stepper drivers.

Table 2: Power distribution of the robot

Output Rail	Load	Purpose
5 VDC	Arduino Mega 2560	Main controller power
5 VDC	MaixCAM Lite	Vision module power
5 VDC	Backup output	Reserved auxiliary power
9 VDC	Motor driver chain	Chassis motor drive power
11.1 VDC	Pump driver board	Water pump power
12 VDC	Pan stepper driver	Pan-axis gimbal power
12 VDC	Tilt stepper driver	Tilt-axis gimbal power

The power schematic is shown in Figure 2. It includes battery connectors, fuses, switch routing, buck module connections, output connectors, LED indicators, and test points. The test points allow each voltage rail to be measured before connecting sensitive modules.

The PCB layout is shown in Figure 3. Wider traces are used for high-current paths, including battery, pump, motor-driver, and stepper-driver rails. Mounting holes are included for chassis installation, and output connectors are placed near the board edges for easier wiring.

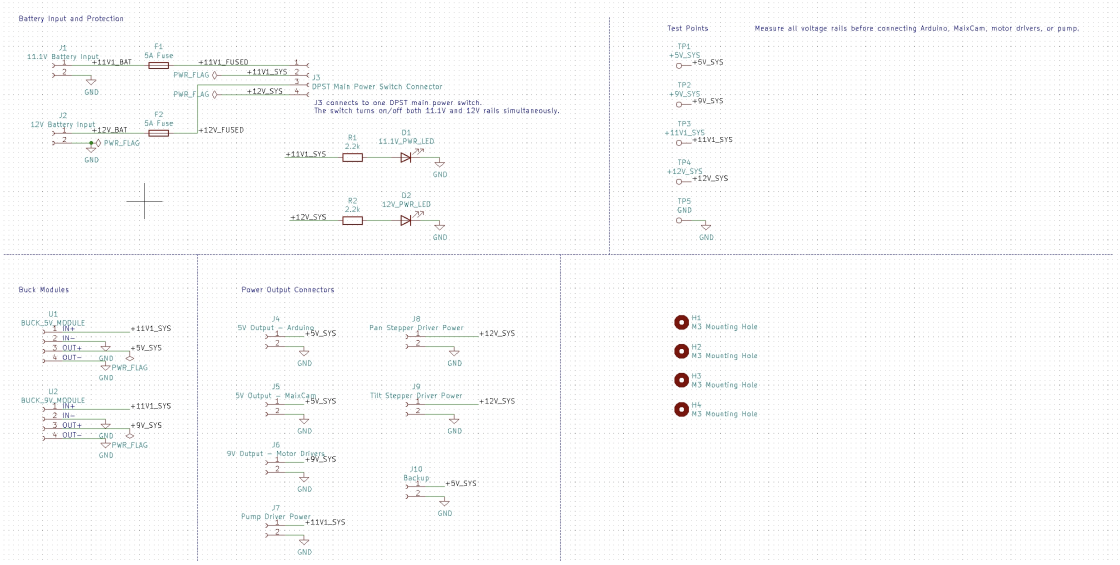

Figure 2: Power distribution schematic showing battery inputs, fuse protection, voltage regulation, output connectors, indicators, and test points.

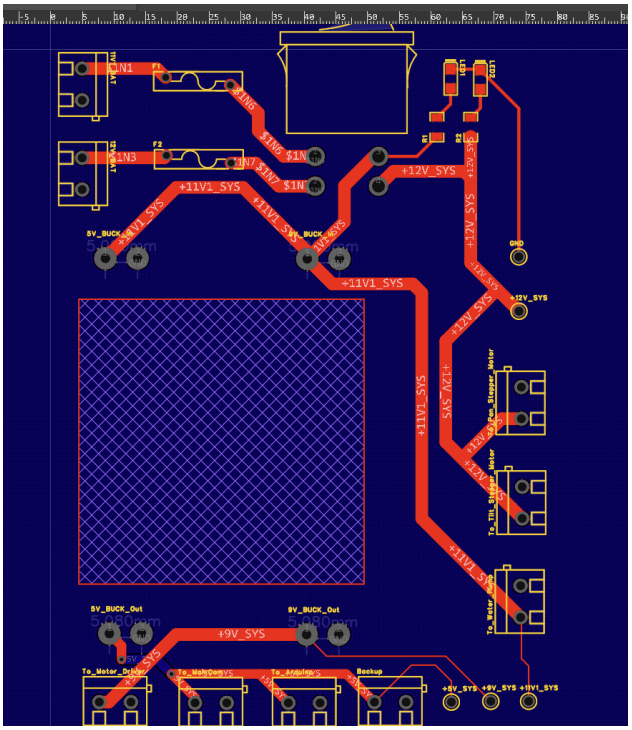

Figure 3: PCB layout for the robot power distribution board.

The signal connector schematic is shown in Figure 4. Although the PCB is mainly a power board, it also includes signal headers to organize wiring between the Arduino and external modules.

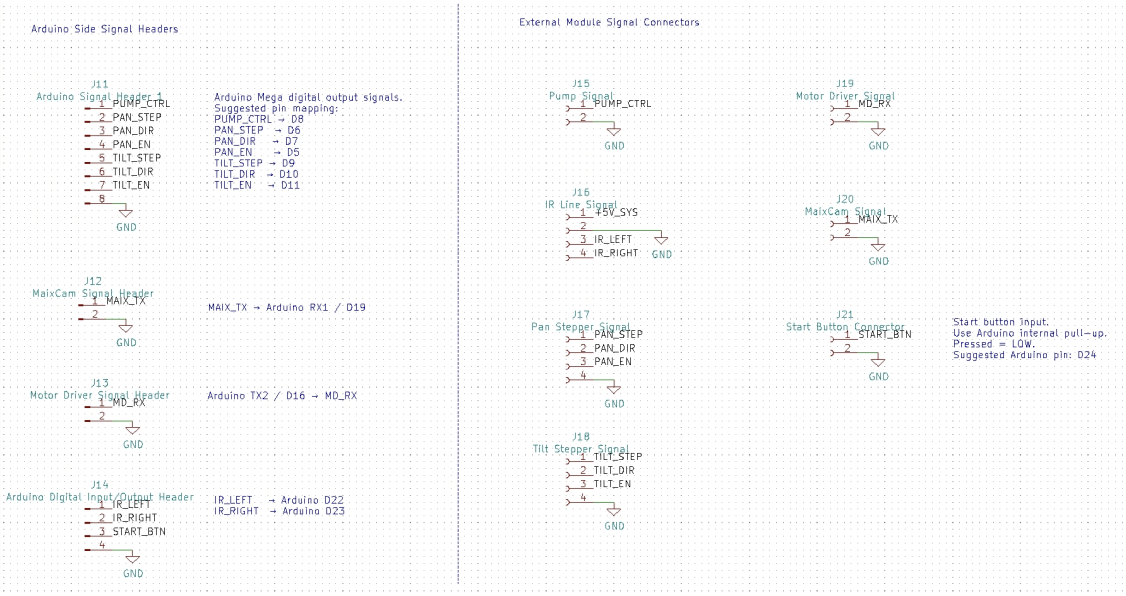

Figure 4: Signal connector schematic for Arduino-side headers and external module connectors.

2.7 Software and System Integration

The integrated software is organized as a three-state finite state machine with `PATROL`, `TRACK`, and `SPRAY` states. A simpler continuous-loop structure was considered, where the robot would always patrol, track, and spray based only on instantaneous sensor values. However, that approach would make debugging harder and could cause unsafe behavior, such as spraying during unstable detections. The state-machine structure was selected because it separates patrol, aiming, and spray activation into distinct behaviors with clear transition conditions.

In `PATROL`, the robot performs line-following patrol while the pan axis scans horizontally. If no target is detected, the robot stays in `PATROL`. If a valid target is detected, the chassis stops and the system transitions to `TRACK`.

In `TRACK`, the pump remains off while the Arduino parses the MaixCAM target message, computes image-coordinate errors, and commands the gimbal using the control laws in Equations 3 and 4. A target is considered locked only when both horizontal and vertical errors are within 15 pixels for 400 ms. This delay prevents brief detection noise from immediately triggering the pump.

In `SPRAY`, the chassis remains stopped and the Arduino activates the pump driver. The current implementation sprays for 1.5 s, then turns the pump off and returns to `TRACK`. If

the target is lost during `TRACK` or `SPRAY`, the pump is turned off and the robot returns to `PATROL`. The software flow is summarized in Figure 5.

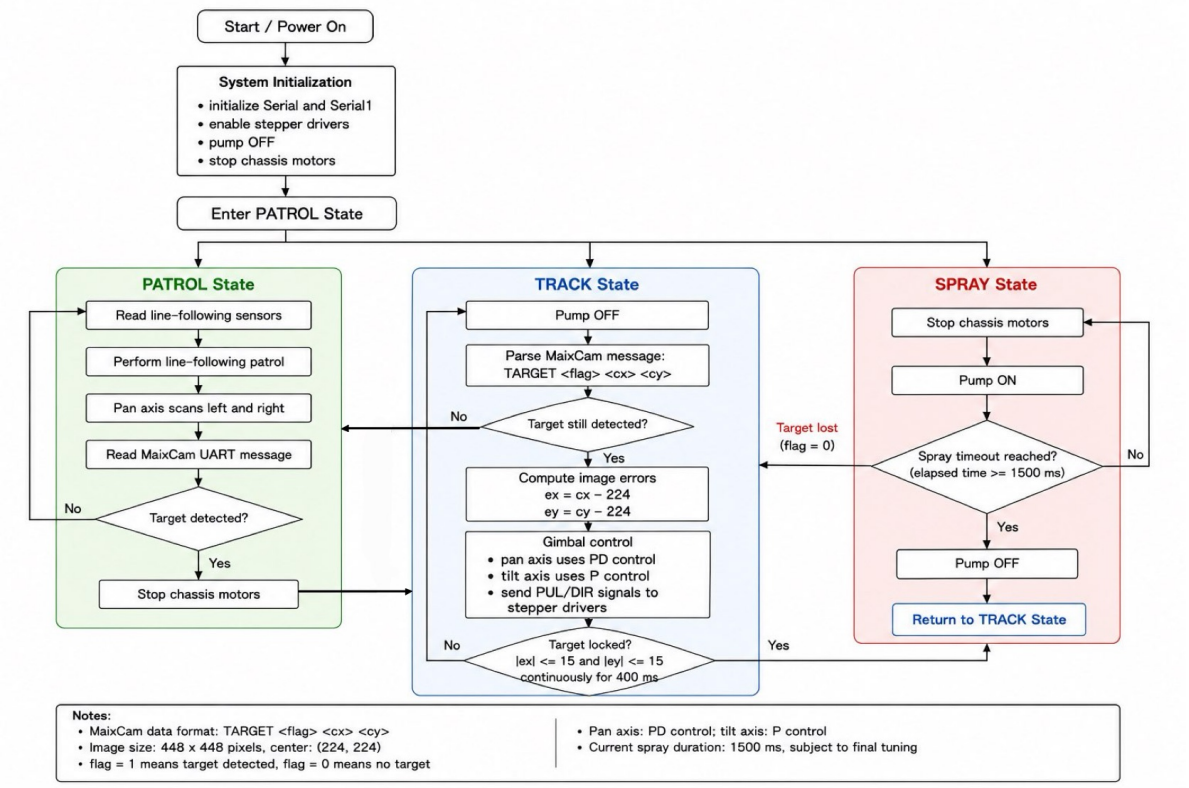

Figure 5: Software flowchart for the integrated patrol, tracking, and deterrence behavior.

This architecture keeps image processing and motor power delivery outside the Arduino. The MaixCAM Lite performs target detection and sends compact target messages, while the Arduino parses the messages and generates control commands. The stepper drivers and motor drivers then handle the electrical power required by the gimbal and chassis motors.

3 Verification

3.1 Verification Overview

The verification process evaluated the autonomous indoor patrol robot at both the subsystem and integrated-system levels. The main goals were to confirm that the robot could follow the predefined patrol path, communicate target information reliably, track detected targets with the gimbal, activate the deterrence mechanism under controlled conditions, and recover safely after target loss.

The testing procedure followed the Requirement Verification (RV) table developed during the design phase. Each subsystem was tested separately before integrated testing to reduce debugging complexity. Verification activities included finite state machine (FSM) transition testing, UART communication testing between the MaixCAM Lite and Arduino Mega 2560, gimbal tracking tests, line-following patrol tests, pump activation tests, and full patrol-to-deterrence workflow testing. The complete RV table is included in Appendix B.

3.2 Control System Verification

3.2.1 FSM State Transition

The FSM was verified to determine whether the robot could transition correctly among its main operating states under different target conditions. The current control logic uses `PATROL`, `TRACK`, and `SPRAY` states. In `PATROL`, the robot follows the predefined line path while monitoring target messages from the MaixCAM Lite. Once a valid target is detected, the robot stops the chassis and transitions to `TRACK`. When the target remains inside the alignment tolerance for the required duration, the system transitions to `SPRAY`. If the target is lost, the system turns off the pump and returns to `PATROL`.

The FSM was tested by manually introducing target-present and target-absent conditions while monitoring serial output and actuator behavior. The tested cases included target detection, target loss, target locking, spray timeout, and return-to-patrol behavior. A transition was considered successful if the robot entered the expected state and produced the expected output, such as stopping the chassis before tracking or disabling the pump after target loss.

The robot consistently followed the intended FSM logic during testing. When a target was detected, the chassis stopped and the gimbal began tracking. When the target remained centered within the defined tolerance window for the required duration, the pump was activated. After the target disappeared, the robot returned to patrol without requiring a manual reset. Therefore, the FSM transition requirement was verified.

The state logic used during verification is the same as the integrated software flow summarized in Figure 5.

3.2.2 Emergency Stop and Safety

Safety behavior was verified to confirm that the robot could avoid unsafe outputs during abnormal or unstable conditions. The main safety concerns were unintended chassis motion during aiming, continuous pump activation, and false spraying caused by temporary detection noise.

Several conditions were tested. First, when the target was lost during `TRACK` or `SPRAY`, the robot was required to disable the pump and return to `PATROL`. Second, the robot was tested to confirm that chassis motion stopped before target tracking and spraying. Third, the target-lock condition was verified to ensure that the pump was not activated by a single unstable detection frame. In the current implementation, the target must remain within the center tolerance window for a specified duration before spraying is allowed.

During verification, the robot successfully stopped spraying when the target was lost and resumed patrol behavior without manual intervention. No continuous spraying or uncontrolled chassis motion was observed during the tested cases. Therefore, the safety-related control requirement was verified for the current prototype.

3.3 Communication and Tracking Verification

3.3.1 UART Communication Reliability

UART communication was verified to confirm that the MaixCAM Lite could transmit target information reliably to the Arduino Mega 2560 during operation. The MaixCAM Lite transmitted target status and target-center coordinates using a baud rate of 115200. The message format was

```
TARGET <flag> <cx> <cy>
```

where `flag` indicates whether a target is detected, and `cx` and `cy` are the target center coordinates in the image frame.

Testing was performed by continuously sending target-present and target-absent messages from the MaixCAM Lite to the Arduino through `Serial1`. The Arduino parsed the received strings and used the parsed values for FSM transitions and gimbal tracking. A communication test was considered successful when the received message was parsed correctly and caused the expected robot response.

During testing, the Arduino consistently received and interpreted the target messages. The robot responded to target updates without obvious communication failure during normal operation. Short-term target interruptions were handled by the target-loss tolerance logic, which reduced unnecessary state switching. Therefore, the UART communication requirement was verified.

3.3.2 Control Response Time

Control response time was verified to evaluate whether the robot could respond quickly enough after target detection. The response sequence included target detection, UART transmission, FSM transition from `PATROL` to `TRACK`, chassis stopping, gimbal tracking, and eventual pump activation after target lock.

Testing was conducted by placing a target into the camera field of view while the robot operated in `PATROL`. After the MaixCAM Lite detected the target and transmitted the target message, the Arduino stopped the chassis and entered `TRACK`. The gimbal then adjusted toward the target based on the image-coordinate error. When the target remained inside the tolerance window for the required duration, the robot entered `SPRAY`.

The robot was observed to stop the chassis and begin tracking shortly after target detection. Minor delays occasionally occurred when the vision result was unstable, but the target-loss and target-lock tolerance logic prevented rapid unintended state switching. Overall, the response was sufficiently fast for the indoor dining-table-scale operating environment, and the control response requirement was verified.

3.3.3 Gimbal Tracking Performance

Gimbal tracking was verified to determine whether the robot could align the water nozzle with the detected target before spraying. The MaixCAM Lite continuously reported target-center coordinates, and the Arduino compared these coordinates with the image center. The pan and tilt commands were then generated from the image-coordinate errors and sent to the stepper drivers as pulse/direction signals.

Testing was conducted by moving the target to different positions within the camera field of view and observing whether the gimbal corrected its orientation toward the target. The verification focused on whether the gimbal moved in the correct direction, reduced the image error, remained stable near the center region, and avoided excessive oscillation.

The gimbal successfully adjusted toward the detected target under normal test conditions. The center tolerance window prevented small coordinate fluctuations from causing continuous motor jitter, and the target-lock duration prevented spraying immediately after momentary alignment. Therefore, the gimbal tracking requirement was verified for the current prototype.

3.4 Drive System Verification

The drive system was verified to confirm that the robot could maintain stable patrol motion using the line-following sensors. The robot uses the two center channels of a four-channel infrared sensor module. The sensor output is digital: the sensor outputs 0 on the black guide line and 1 on the surrounding non-black region.

Testing included straight-line following and small lateral deviations from the guide line. When both center sensors detected the line, the robot moved forward. When one sensor

moved away from the line, the controller applied a corrective steering action to bring the robot back to the predefined path. The test was considered successful if the robot maintained forward patrol and recovered from small deviations without losing the line.

The robot followed the predefined path reliably under the tested conditions. The drive system also responded correctly during FSM transitions: the chassis stopped after target detection before tracking, and patrol resumed after target loss. Therefore, the drive system requirement was verified.

3.5 Integrated System Verification

Integrated system verification evaluated whether the sensing, control, drive, gimbal, and pump subsystems could operate together during the complete robot workflow. The tested sequence was patrol, target detection, chassis stop, gimbal tracking, target lock, spray activation, target loss, and return to patrol.

During testing, the robot began in `PATROL` and followed the predefined path. When the MaixCAM Lite detected a target, the target information was transmitted to the Arduino Mega 2560 through UART. The FSM then transitioned to `TRACK`, and the chassis stopped before aiming. The gimbal adjusted according to the target coordinates until the target remained within the center tolerance window. After the lock condition was satisfied, the system entered `SPRAY` and activated the pump for the configured spray duration. When the target disappeared, the system returned to `PATROL`.

The integrated test showed that the major subsystems could operate together without major functional conflicts. UART communication, FSM transitions, gimbal tracking, line-following control, and pump activation were coordinated successfully under normal test conditions. Some detection instability was observed when the target moved quickly or appeared near the edge of the camera view, but the tolerance mechanisms reduced unnecessary state switching and accidental pump activation. Overall, the robot completed the full patrol-to-deterrence workflow and satisfied the main integrated-system verification requirement.

3.6 Sensing System Verification

3.6.1 Frame Rate and Detection Latency

The real-time performance of the vision system was verified through two key metrics: RGB frame rate and end-to-end detection latency. Both metrics directly influence the robot's ability to track moving targets and issue timely spray commands.

Frame rate. The effective frame rate was measured by counting the total number of processed frames in a fixed runtime and calculating the average frames per second (fps). The measurement covered the full pipeline: image capture, YOLO inference, post-processing, on-screen display (when enabled), and UART transmission.Two configurations were tested:

- **With built-in video recording for debugging**: the average frame rate was 16.04 fps.

- **Without video recording** (final demonstration configuration): the average frame rate was 26.97 fps, above the specification.

All subsequent verification tests were performed without video recording to maximize real-time performance. The frame rate remained stable regardless of target presence because the inference pipeline runs at a fixed rate.

Detection latency. The software end-to-end processing latency was defined as the time from image capture completion to UART transmission completion (including inference, post-processing, and serial write). A high-precision timestamp was recorded at the start of each frame's processing, and another timestamp was recorded immediately after the UART write. The difference was computed for every processed frame over a fixed runtime (120 seconds). The average latency was 11.7 ms, which accounts for approximately 32% of the frame interval (37 ms). The maximum measured latency was below the frame interval, indicating that the system never drops frames due to processing backlogs. Given a typical target width of 100 pixels in the image, the maximum allowable pixel displacement velocity without missing the target between frames can be estimated as $v_{\max} = 0.5 \times \text{width/frame interval} = 0.5 \times 100/0.037 \approx 1351$ pixels/second. This corresponds to roughly 13 times the target's body length per second. Assuming a cat body length of 0.5 m, the equivalent speed is 6.5 m/s. In the intended desktop scenario, cats typically move at speeds below 1 m/s, well within the tracking capability of the system. Therefore, the software latency poses no practical issue for real-time targeting.

3.6.2 Detection Accuracy and False Positive Rate

Cat detection success rate. Tests were conducted at three distance ranges using a plush cat toy as the target, under indoor lighting: close (0.3–0.5 m), medium (0.5–0.8 m), and far (0.8–1.0 m). At each range, 100 test frames were evaluated. The measured success rates are:

- Close range (0.3–0.5 m): 95%
- Medium range (0.5–0.8 m): 85%
- Far range (0.8–1.0 m): 70%

Only the close range meets the 90% requirement; the medium and far ranges fail. The primary causes are:

- The fixed-focus lens of the MaixCAM is calibrated for close-range objects. At medium and far distances, the target appears blurry and lacks fine detail.
- Low pixel resolution on the target at longer distances makes it difficult for the YOLO model to distinguish cat-specific features.
- The plush toy used for testing differs from a real cat, which may affect the model's generalization. However, the trend of performance degradation with distance is expected to be similar with real cats.

False positive rate. Preliminary tests without any cat target in the scene (indoor desk environment with typical background objects) gave a false positive rate of approximately 20%, exceeding the 10% specification. False positives mainly occurred when objects with cat-like contours (e.g., round cushions, dark corner shadows) were present. The limited resolution of the MaixCAM forces the model to rely primarily on shape and silhouette rather than fine texture, leading to misclassifications.

Planned improvements. Although the current model does not fully meet the accuracy and false positive requirements, the following improvements are ongoing:

- Retrain the YOLO model with a more diverse dataset that includes more out-of-focus cat images at various distances and hard negative samples from the target desktop environment.
- Adjust the confidence threshold.

3.6.3 Continuous operation and heat dissipation capability

The vision system was tested for continuous operation over 30 minutes under indoor desk lighting (ambient temperature 22°C). The MaixCAM module became noticeably warm but not too uncomfortable to touch (estimated temperature ¡50°C). During the test, there was no frame rate degradation, unexpected reset, or communication failure. The thermal performance is considered acceptable for the intended indoor desktop use case.

3.6.4 Other Verification Details

The original design included a distance measurement requirement using a depth sensor. Due to the revised system architecture, target alignment relies solely on visual feedback through gimbal tracking, and the distance measurement requirement has been removed. Other verification details, including complete pass/fail status and measurement data, are provided in Appendix B.

4 Cost

4.1 Component Costs

We purchased the components shown in Table 4 (see Appendix C) for the prototype. The table lists the retail price and the actual amount paid. The total retail cost was ¥1,275.42, and the total paid cost was ¥1,254.72.

4.2 Labor Costs

Table 5 in Appendix C summarizes the labor cost for each team member, calculated as ideal hourly rate × actual hours × 2.5. We assume an hourly rate of ¥58.33. The total labor cost is ¥27,998.40 for all 4 members' work.

4.3 Mass Production Estimate

If the robot were produced in quantities of 1000 units, we assume bulk purchase prices are approximately 70% of the retail prices.the per-unit production component cost is:

$$1,275.42 \text{ RMB} \times 0.7 \ = 892.79 \text{ RMB}.$$

For assembly and testing of each unit, a skilled worker is expected to spend 3.5 hours. Using the same labor cost formula as in development (ideal hourly rate $\times$ hours $\times$ 2.5), the per-unit production labor cost is:

$$58.33 \text{ RMB/hour} \times 3.5 \text{ hours} \times 2.5 = 510.39 \text{ RMB}.$$

The one-time labor cost for the prototype, when amortized over 1000 units, contributes $27,998.4/1000 = 28$ RMB per unit.

The total estimated cost per unit for a production volume of 1000 units is:

$$892.79 \text{ (components)} + 510.39 \text{ (assembly labor)} + 28.00 \text{ (amortized NRE)} = 1431.18 \text{ RMB}.$$

If the robot is sold at 1500 RMB per unit, the gross profit per unit would be approximately 68.82 RMB, making the product profitable. A lower bulk purchase price or higher selling price or larger production volume would further improve profitability.

5 Conclusion

The completed prototype demonstrates that the proposed autonomous indoor patrol robot is technically feasible for a dining-table-scale cat-deterrence task. The robot can patrol along a predefined path, detect a cat-like target with the MaixCAM vision module, transmit target coordinates to the Arduino Mega 2560, aim a stepper-motor gimbal, and activate the water-blast mechanism only after the target is aligned. These results show that the design works as an integrated sensing, control, actuation, drive, and power-distribution system rather than as isolated subsystems.

The major accomplishments of the project were the successful integration of embedded vision, UART communication, finite-state-machine control, stepper-based aiming, line-following patrol, and controlled water output. Verification showed that the MaixCAM pipeline provided real-time target data, the Arduino controller responded reliably to target-present and target-lost conditions, the gimbal could align the nozzle toward the detected target, and the water-blast subsystem reached the required effective distance of at least 1.5 m. The custom power-distribution PCB also supported the controller, vision module, motors, stepper drivers, and pump without causing unexpected resets during normal prototype testing.

Some uncertainties remain. The vision system performed well at close range, but detection accuracy decreased at medium and far distances, and false positives were observed when background objects had cat-like shapes. For this reason, the practical operating specification should emphasize close-range indoor use unless the model is retrained with a larger and more diverse dataset. The spray direction was also affected by nozzle mounting, tube stiffness, and water-stream breakup. Future revisions should improve the camera-nozzle alignment, add a more rigid nozzle mount, and include closed-loop gimbal feedback or a homing routine to improve long-term repeatability.

Ethically, the design must remain a humane deterrence system rather than a device intended to injure or excessively frighten animals. This is consistent with the IEEE Code of Ethics, which emphasizes public safety, welfare, and responsible engineering decisions [1]. It is also consistent with animal-welfare guidance from the American Veterinary Medical Association, which emphasizes minimizing fear, pain, stress, and suffering when humans interact with animals [4]. For this project, that means limiting spray duration, keeping water pressure mild, avoiding continuous activation, and using the robot only in supervised and appropriate indoor environments.

The broader impacts of the project are also important. Socially, the robot could reduce the need for constant human supervision and provide a non-contact way to protect food-preparation or dining surfaces. At the same time, because the system uses a camera, privacy should be considered; the prototype should process images locally and avoid storing unnecessary personal data, consistent with privacy-risk-management principles such as those described by NIST [5]. Environmentally and economically, the design should minimize water use by using short spray bursts and activating only after confirmed target alignment. This supports the general water-efficiency principles promoted by EPA WaterSense [6]. Overall, the project provides a strong foundation for a compact indoor

deterrence robot, while clearly identifying the sensing, mechanical, safety, and ethical refinements needed before broader deployment.

References

[1] IEEE. ""IEEE Code of Ethics",," Accessed: Feb. 8, 2020. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html.

[2] Arduino. "Mega 2560 Rev3," Accessed: May 14, 2026. [Online]. Available: https://docs.arduino.cc/hardware/mega-2560/.

[3] Sipeed. "Sipeed MaixCam Documentation," Accessed: May 14, 2026. [Online]. Available: https://wiki.sipeed.com/maixpy/.

[4] American Veterinary Medical Association. "AVMA Animal Welfare Principles," Accessed: May 15, 2026. [Online]. Available: https://www.avma.org/resources-tools/avma-policies/avma-animal-welfare-principles.

[5] National Institute of Standards and Technology. "NIST Privacy Framework," Accessed: May 15, 2026. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.40.ipd.pdf.

[6] U.S. Environmental Protection Agency. "Why Save Water," Accessed: May 15, 2026. [Online]. Available: https://www.epa.gov/watersense/watersense-kids#Why%20Save%20Water.

Appendix A Complete Circuit Schematic

This appendix provides the complete circuit schematic for the robot power distribution and signal interface board. The main text presents the schematic in separate functional views to improve readability, including the power distribution schematic and the signal connector schematic. Figure 6 shows the full schematic in one view for reference during system assembly, debugging, and future modification.

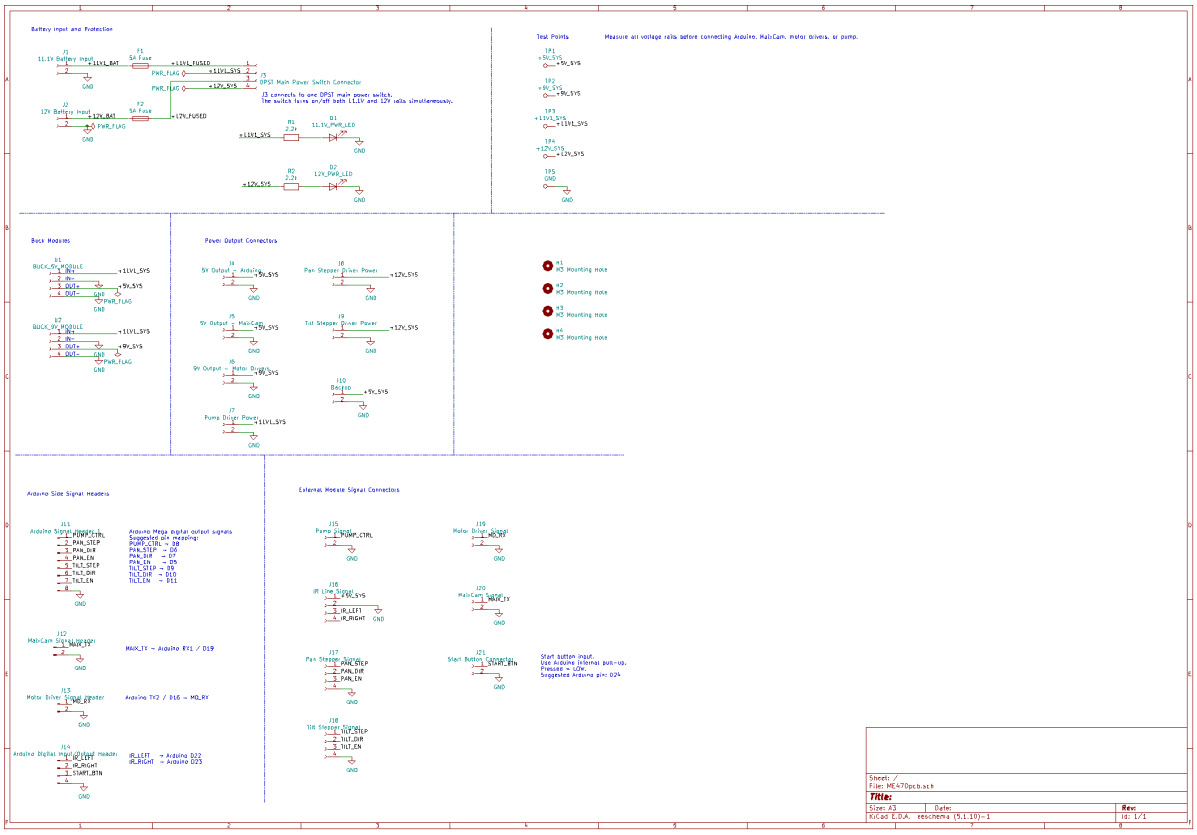

Figure 6: Complete circuit schematic of the robot power distribution and signal interface board.

Appendix B Requirements Verification Table

Table 3: Requirements Verification Table

Req ID	Requirement	Verification Method	Result
RV1	The vision system shall provide RGB data at a minimum frame rate of 15 fps and shall complete one detection and position calculation cycle from image capture to UART output within 0.5 seconds.	Frame rate measured over 120 seconds of continuous operation (average 26.97 fps). Latency measured as software end-to-end processing time from image capture completion to UART write completion (average 11.7 ms).	Pass
RV2	The vision system shall detect a cat target at distances from 0.3 m to 1.0 m under indoor lighting with a success rate of at least 90%, and shall have a false positive rate not exceeding 10% when no cat is present.	Detection success rate tested at three distance ranges: 0.3–0.5 m (95%), 0.5–0.8 m (85%), 0.8–1.0 m (70%). False positive rate preliminary test approximately 20%.	Fail
RV3	The system shall measure target distance using the Astra Pro depth camera.	Requirement removed after design review. No verification performed.	Removed
RV4	The robot shall implement FSM-based control logic.	Verified through FSM state transition testing between PATROL, TRACK, SPRAY, and STOP modes under different operating conditions.	Pass
RV5	The control system shall stop chassis movement during target tracking.	Verified by observing chassis stop behavior immediately after cat detection before gimbal tracking activation.	Pass
RV6	The robot shall communicate target information through UART.	Verified through continuous UART communication testing using `TARGET <flag> <cx> <cy>` messages between MaixCAM and Arduino Mega 2560.	Pass
RV7	The robot shall perform gimbal tracking based on target position.	Verified through tracking experiments in which the gimbal adjusted according to target center coordinates and aligned the target near image center.	Pass
RV8	The robot shall implement emergency stop and safety protection.	Verified through target-loss handling, STOP state testing, and prevention of unintended spraying behavior.	Pass

Req ID	Requirement	Verification Method	Result
RV9	The gimbal shall rotate the nozzle quickly and repeatably enough to track the detected target.	Verified through repeated pan/tilt motion cycles using target-coordinate input from the MaixCAM and observing whether the gimbal reached the target center region without skipped steps, stalling, or excessive oscillation.	Pass
RV10	The pump-nozzle assembly shall provide an effective spray distance of at least 1.5 m.	Verified by filling the reservoir, placing distance markers, activating the pump for short intervals, and measuring the farthest consistent water impact point.	Pass
RV11	The spray direction shall cover the expected target area after gimbal alignment.	Verified by placing a target board at different offsets, aligning using vision tracking, and activating the pump only after consecutive centered frames.	Partial Pass
RV12	The PCB shall provide stable voltage rails for the controller, vision module, stepper drivers, chassis motors, and pump driver.	Verified by measuring output rails before load connection and during integrated operation, including pump and motor activation.	Pass
RV13	The 5 V rail shall power the Arduino and MaixCAM simultaneously, and the pump rail shall support pump startup and current draw.	Verified by operating the Arduino and MaixCAM from the 5 V rail while activating the pump through its driver circuit, then monitoring resets, communication loss, and voltage stability.	Pass
RV14	The robot shall follow the predefined patrol path.	Verified through line-following experiments using center sensors under straight and correction conditions.	Pass
RV15	The robot shall transition from patrol to tracking after target detection.	Verified through integrated system testing with FSM state transition observation.	Pass
RV16	The robot shall activate spraying after successful target alignment.	Verified through center tolerance testing and consecutive frame confirmation.	Pass
RV17	The robot shall resume patrol after target disappearance.	Verified through target-loss testing and automatic FSM recovery behavior.	Pass
RV18	The robot shall maintain reliable control response during operation.	Verified through response time testing between detection, tracking, and spraying actions.	Pass
RV19	The robot shall maintain stable subsystem integration.	Verified through integrated system testing involving communication, tracking, and drive control.	Pass
RV20	The robot shall complete the full cat deterrence workflow.	Verified through end-to-end testing from patrol, detection, tracking, spraying, and recovery.	Pass

Appendix C Cost

C.1 Component Costs

Table 4 lists the components used in the prototype. Retail prices are market prices at the time of purchase; paid prices reflect what the team actually paid (3D-printed parts are marked as 0).

Table 4: Component Costs (Retail and Paid)

Part	**Retail Cost (RMB)**	**Paid Cost (RMB)**
Arduino Mega 2560 development board	86.19	74.49
MaixCAM Lite vision module	292	292
Stepper motors	130	130
TT DC geared motors	56.39	56.39
motor driver module and Gimbal mechanical mount	120	120
Water pump system((includes 12V battery))	92	92
11.1V battery and Battery charger	120.99	120.99
Electronic components(e.g. 5A fuse, LED indicators)	19.06	19.06
Buck converter	114.87	110.87
Custom PCB	56.89	56.89
4-channel IR line following sensor module	25.8	25.8
water tank	0	0
Robot chassis and wheels	146.23	141.23
Wires, connectors, Dupont jumper cables	10	10
Screws, nuts, and other hardware	5	5
Total	**1275.42**	**1254.72**

C.2 Labor Costs

The labor cost is calculated for each team member as:

$$\text{Labor cost} = \text{ideal hourly rate} \times \text{actual hours} \times 2.5$$

The hours reported include all one-time engineering efforts: system design, model training, software development, debugging, assembly, and testing. These costs are non-recurring and apply to the development of the prototype only.

For the purpose of estimating the ideal hourly rate, we assume a standard full-time workload of 8 hours per day and 5 working days per week. With approximately 30 days per

month, the effective working days per month are $30 \times (5/7) \approx 21.43$ days. This yields $21.43 \times 8 = 171.43$ working hours per month. Assuming a target monthly salary of 10,000 RMB, the corresponding hourly rate is $10,000/171.43 \approx 58.33$ RMB/hour.

Table 5 summarizes the labor cost.

Table 5: One-time Labor Cost Breakdown

Member	**Ideal Hourly Rate (RMB)**	**Actual Hours**	**Multiplier**	**Total (RMB)**
Chentao Fang	58.33	48	2.5	6999.6
Jiawei Kong	58.33	48	2.5	6999.6
Ronglong Liu	58.33	48	2.5	6999.6
Yanchen Liu	58.33	48	2.5	6999.6
Total Labor Cost				27998.4