Surveillance Robo Car

A Mini Project Report submitted in partial fulfillment
For the award of the degree of

Bachelor of Technology
in
Electronics and Computer Engineering

By

1. **Mr. Akshay Mittha (Roll. No. 64)**
2. **Mr. Soham Nipunage (Roll. No. 68)**
3. **Mr. Darshan Junavane (Roll. No. 69)**

Under the Guidance of
Prof. R. N. Chavan

Walchand Institute of Technology, Solapur

(An Autonomous Institute)

Department of Electronics and Computer Engineering

Affiliated To
Punyashlok Ahilyadevi Holkar Solapur University, Solapur

Academic Year 2024-25

DECLARATION

We declare that

a. The work contained in this report is original and has been done by us under the guidance of our supervisor.

b. The work has not been submitted to any other Institute for any degree or diploma.

c. We have followed the guidelines provided by the Institute in preparing the report.

d. We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references.

Name of the Student	**Signature**
1. Mr. Akshay Mittha	
2. Mr. Soham Nipunage	
3. Mr. Darshan Junavane	

Place: Walchand Institute Of Technology, Solapur

Date:

Walchand Institute of Technology, Solapur
Department of Electronics and Computer Engineering

CERTIFICATE

This is to certify that the project report entitled ***Surveillance Robo Car*** *submitted by -*

Name of the Student	*Roll No*	*Exam Seat No*
1. Mr. Akshay Mittha	*64*	*2303112001*
2. Mr. Soham Nipunage	*68*	*2303112005*
3. Mr. Darshan Junavane	*69*	*2303112006*

to the Walchand Institute of Technology, Solapur in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Electronics& Computer Engineering is a bonafide record of work carried out by them under my guidance and supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute for the award of any Degree. This project is approved for the award of the Degree of Bachelor of Technology in Electronics & Computer Engineering.

Mr. R. N. Chavan	*Dr. S.R. Gengaje*	*Dr. V.A. Athavale*
Supervisor	**Head of the Department**	**Principal**

Date: 22/4/2025

Place: Solapur

ACKNOWLEDGEMENT

We take this opportunity to express our heartfelt gratitude to all those who supported us throughout the course of our mini project titled **"Surveillance Robo Car."**

First and foremost, we would like to express our sincere thanks to **Mr. R. N. Chavan Sir**, our respected project guide, for his valuable guidance, encouragement, and continuous support at every stage of this project. His expertise and insightful suggestions helped us improve the quality and scope of our work.

We also extend our gratitude to **Dr. A. S. Gundale**, Mini Project Coordinator, for his coordination, encouragement, and support throughout the project phase. A special thanks to **Mr. Khamitkar Sir**, IoT Expert, for sharing his technical knowledge and practical insights which greatly enhanced our understanding and helped us implement the project effectively.

We would also like to thank the entire **Department of Electronics and Computer Engineering** for providing us the workspace and a basic environment which enabled us to carry out our project activities smoothly.

Last but not least, we are thankful to our friends, classmates, and family members who provided us with continuous support and motivation during the entire duration of this project.

Thank you all.

ABSTRACT

In today's world, surveillance and security are of paramount importance, yet traditional CCTV systems are limited by their fixed position and high cost. Our project, **"Surveillance Robo Car"** addresses this issue by providing a low-cost, mobile, and wireless surveillance solution that combines live video streaming with remote control functionality. The primary objective of this project is to design and develop a compact robotic car capable of navigating various indoor environments while streaming real-time video over a wireless network.

The system is built using an ESP32-CAM module, integrated with an L298 motor driver, DC motors, and powered through a compact chassis. The ESP32-CAM is programmed using Arduino IDE in C language, and video streaming is accessed via a web interface. Control commands are also executed through the same interface, ensuring complete wireless operation.

The robot was successfully tested in multiple scenarios for mobility, video clarity, and wireless performance. The results confirmed the system's ability to perform real-time surveillance, even during power outages or in areas where fixed cameras cannot be installed. This project demonstrates a scalable and cost-effective approach to mobile security and monitoring using embedded systems and IoT technologies.

CONTENTS

LIST OF FIGURES

Introduction

The Surveillance Bot using ESP32-CAM and Mobile-Operated Control System is a low-cost, IoT-based robotic platform designed for real-time video monitoring and remote control. The core of the system is the ESP32-CAM module, which integrates a microcontroller and a camera module capable of streaming video over Wi-Fi. The bot is controlled using a smartphone through a web interface or app, allowing the user to maneuver the robot and monitor its surroundings in real time.

This project is especially relevant in the context of increasing demand for affordable and smart surveillance solutions. By combining wireless video streaming and remote mobility, it provides a compact and scalable solution for real-time security monitoring in both indoor and outdoor environments.

Surveillance is a real time collection and analysis of data that is timely distributes the information to the operator. Surveillance in Defense Applications plays an important role for keeping an eye out in order to protect its citizens and take necessary actions. Surveillance is the task of monitoring the set of conditions. This generally occurs in a military scenario where surveillance war areas, adversary territory. Human surveillance is carried by experienced work forces in close sensitive areas so as to continually monitor for changes. Whereas there is always added risks of losing work force in the time of getting caught by the adversary. With advanced technology in past years, there it is possibility to monitor areas of importance remotely by the use of robots instead of human. Apart from the given advantages of not losing any work forces, physical and elegant robots can be used detect subtle elements that are not conspicuous to people.

Surveillance is a crucial task, we cannot put someone life to risk, instead of that we can use this kind of robots which do not need sleep, they don’t get hungry, they don’t have emotions, they are just stick to their duties and follow the orders. Nothing can be more important than human life. Use of such robots can help to save many lives on border areas. And we can use this manpower in other tasks.

Literature Survey

The paper "Surveillance Robot" explores the development of a mobile surveillance bot equipped with an ESP32-CAM module, controlled via an Android device over Wi-Fi. The system incorporates components such as an L298N motor driver, DC motors, a mini breadboard, and a servo motor for dynamic camera rotation. The primary aim is to enable remote monitoring in sensitive environments like war zones and industrial areas, minimizing human risk. The ESP32-CAM streams real-time video over an IP interface, allowing operators to observe and control the robot through mobile devices.

In our implementation, we adopted the core architecture from this design but excluded the servo motor, thereby fixing the camera's angle to face forward. This simplification reduces mechanical complexity while maintaining core surveillance functionality. Related works and DIY projects, such as those from Viral Science and #include electronics, also validate this approach, emphasizing the importance of low-cost, wireless surveillance solutions.

These systems highlight the growing relevance of remotely operated surveillance bots in areas like border monitoring, hazardous inspection, and search-and-rescue missions. By minimizing physical intervention, they provide a scalable and safe alternative to traditional surveillance methods.

Problem Statement

Implementing comprehensive surveillance solutions often proves financially unfeasible for small businesses, residential users, and institutions with limited budgets. Traditional systems require costly infrastructure and maintenance, creating barriers to adoption. Simultaneously, there is an increasing need for safe and real-time monitoring of environments that are hazardous or inaccessible to humans—such as disaster-stricken areas, industrial zones, or high-risk security locations.

Solution

To address these dual challenges, there is a pressing need for a cost-effective, mobile surveillance robot that combines affordability, portability, and wireless control without sacrificing core surveillance functionalities. By leveraging compact and low-cost components like the ESP32-CAM for real-time video streaming and integrating remote control via smartphones or web interfaces, such a system can provide a reliable alternative to traditional surveillance systems. This not only democratizes access to security but also enables efficient monitoring in environments where human presence is risky or impossible.

Objectives and Scope

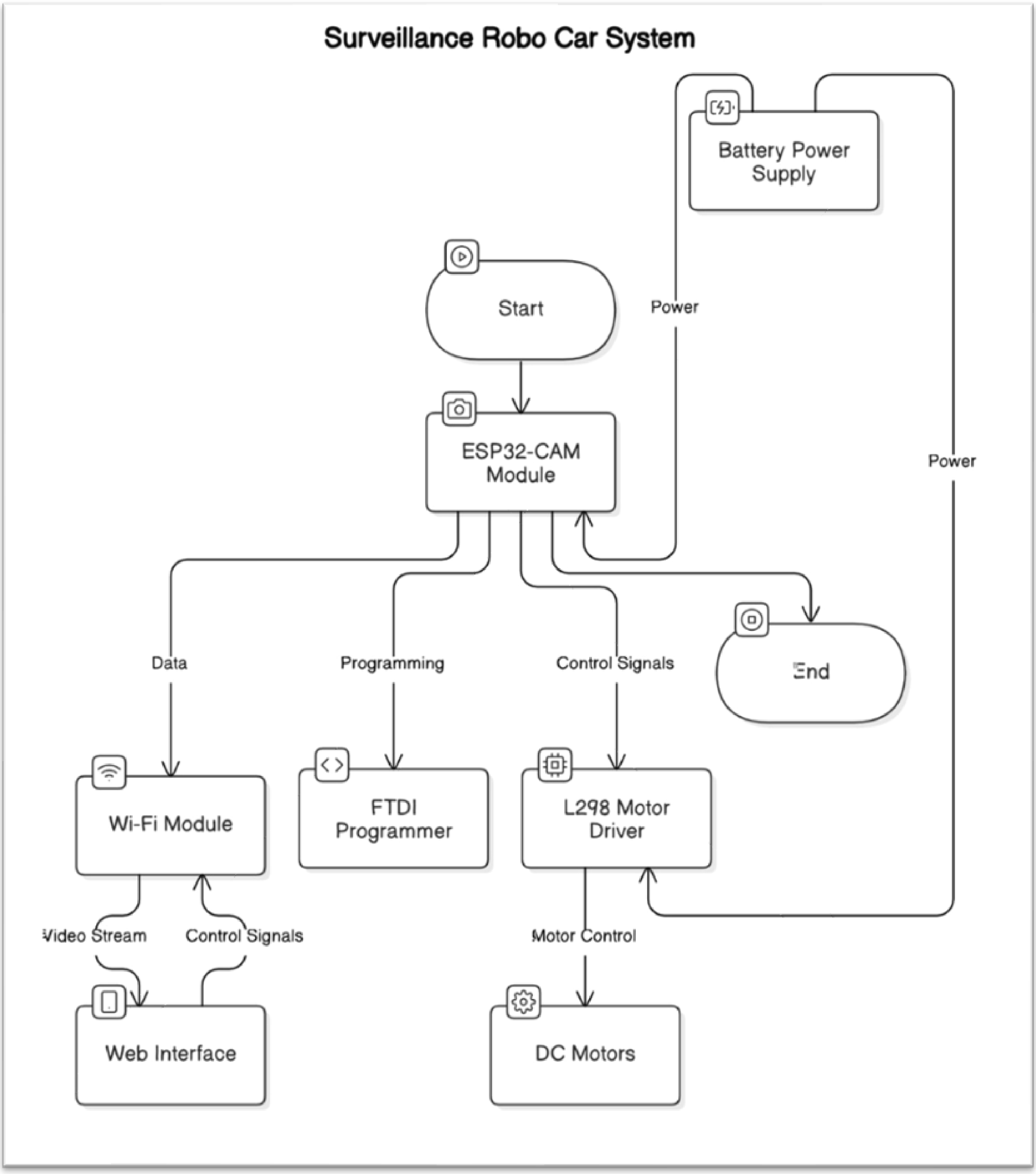
Objectives:

- To design and develop a mobile surveillance robot using the ESP32-CAM module capable of capturing and streaming live video over Wi-Fi.
- To implement remote control functionality via a smartphone or web interface to navigate the bot in real-time.
- To provide a low-cost, portable surveillance system that can be deployed in restricted, hazardous, or inaccessible environments.
- To eliminate the need for servo mechanisms by fixing the camera at a strategic angle to reduce hardware complexity while retaining effective surveillance capability.
- To ensure easy scalability and customization for future upgrades such as night vision, object detection, or autonomous navigation.

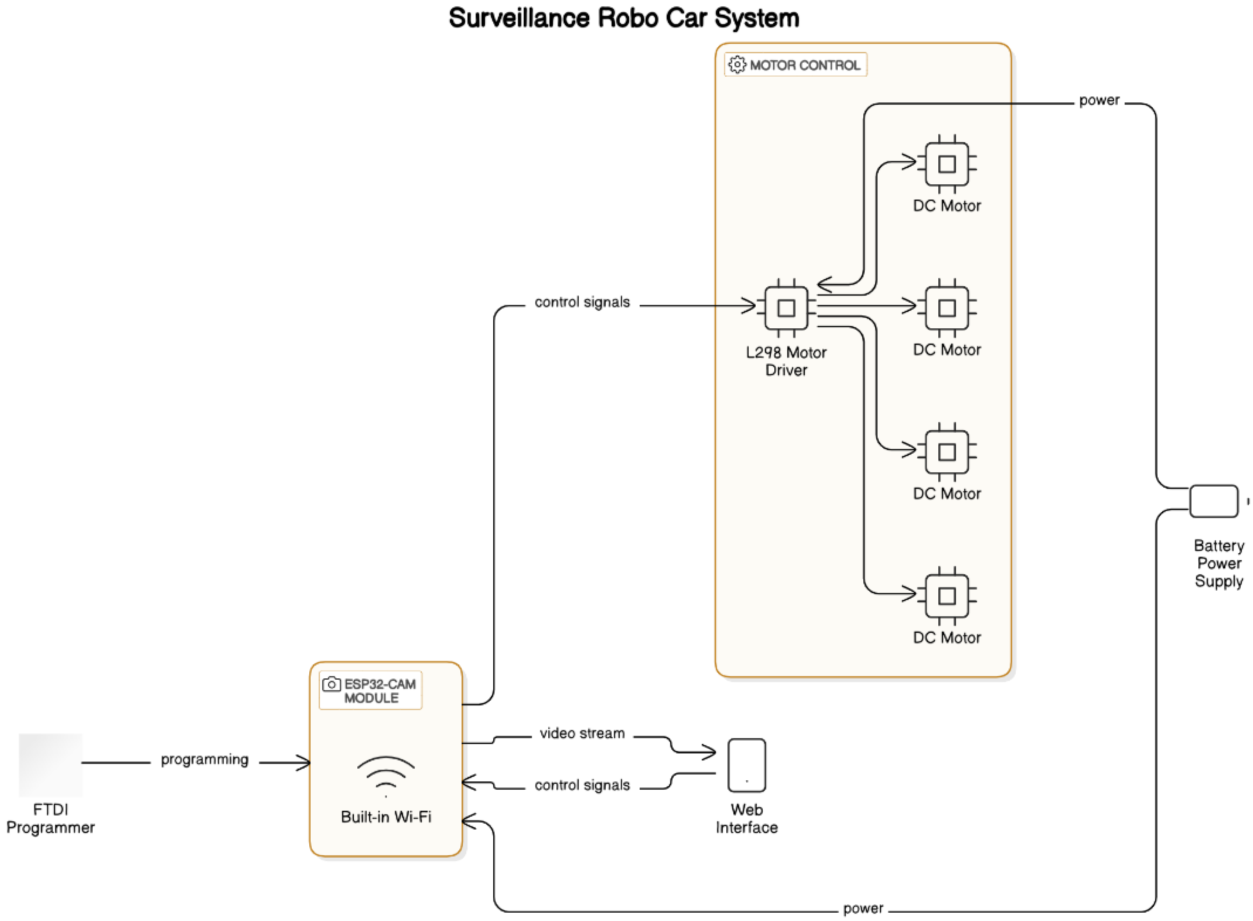
Scope:

- The project focuses on building a mobile, Wi-Fi enabled surveillance robot using ESP32-CAM, L298N motor driver, and DC motors.
- The bot will offer live video streaming to the user through a web-based interface (accessible via smartphone or computer).
- The system is intended for small-scale, real-time monitoring applications in areas such as homes, offices, warehouses, or disaster zones.
- The camera will be mounted in a fixed forward-facing position, simplifying the design and control.
- This project does not includ**e** advanced features like AI-based tracking, obstacle avoidance, GPS, or night vision in its current scope, but it can be extended in future iterations.

Block Diagram



System Architecture

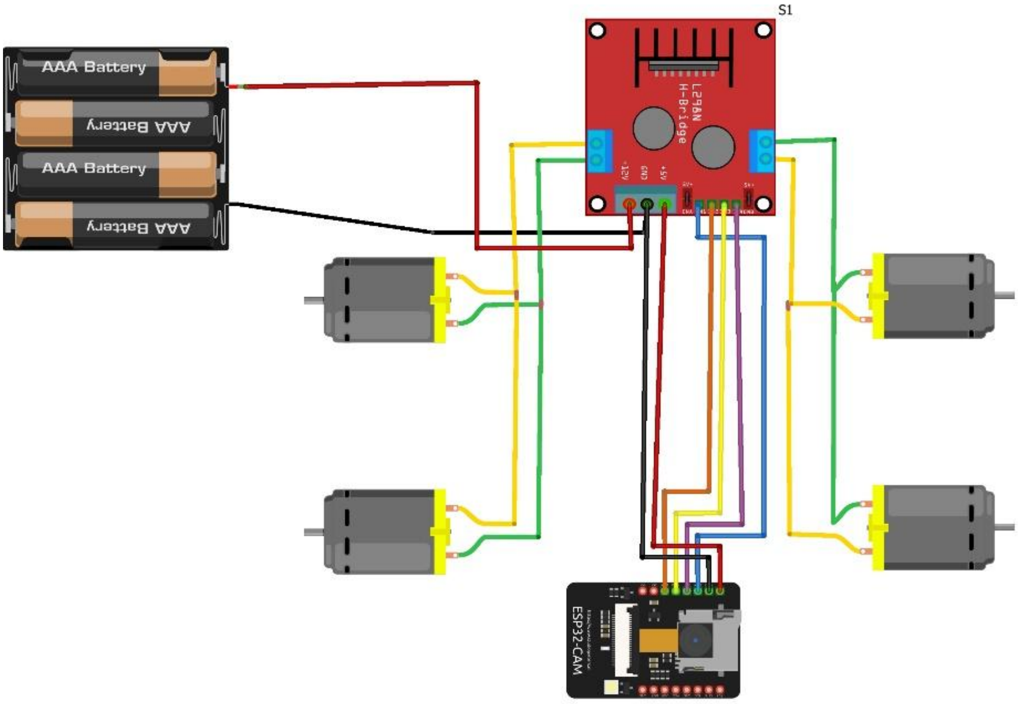


Components and Specifications

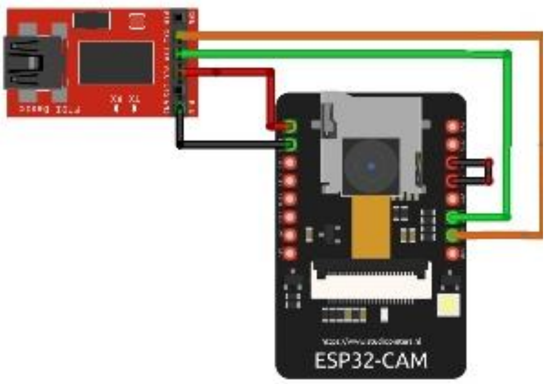
1. **DC Motors (150–300 RPM) – 4 Units**
   - Used to drive the wheels of the robot.
   - Operate at 6–12V DC with rotational speeds between 150–300 RPM.
   - Provide sufficient torque for indoor and outdoor movement.
2. **L298N Motor Driver**
   - Dual H-Bridge driver module used to control the speed and direction of two pairs of DC motors.
   - Supports up to 2A current per channel, compatible with ESP32 logic levels.
   - Provides separate power for motor and logic circuitry.
3. **FTDI Programming Module for ESP32-CAM**
   - USB-to-Serial adapter used to upload code to the ESP32-CAM.
   - Converts USB signals to serial TTL, typically using a CP2102 or FT232RL chipset.
   - Required due to ESP32-CAM's lack of onboard USB port.
4. **OV2640 Camera (with ESP32-CAM)**
   - 2MP camera module integrated with the ESP32-CAM board.
   - Supports image resolutions from 160x120 to 1600x1200.
   - Capable of real-time video streaming over Wi-Fi.
5. **ESP32-CAM (AI Thinker Module)**
   - Microcontroller with Wi-Fi + Bluetooth and onboard OV2640 camera.
   - 32-bit dual-core processor, 520KB SRAM, 4MB Flash.
   - Used for camera streaming and web server hosting for mobile control.

6. **Lithium-Ion Battery (3.7V) – 3 Cells**
    - Rechargeable power source used to operate motors and ESP32-CAM.
    - Provides enough voltage and current when used in series (up to ~11.1V).
    - Lightweight and energy-dense.
7. **Battery Holder (3x18650 Cell Holder)**
    - Holds and connects three 18650 Li-ion cells in series.
    - Provides convenient wiring and stable support for power supply.
    - Typically includes leads for easy integration into circuits.
8. **Jumper Wires (Male-to-Female and Female-to-Female)**
    - Used for electrical connections between modules, motors, and power supply.
    - Allow flexible, plug-and-play wiring without soldering.
    - Available in various lengths for layout customization.
9. **Wheels (7 x 2 cm) – 4 Units**
    - Rubber or plastic wheels attached to DC motors.
    - Provide mobility to the bot and allow stable navigation.
    - Sized for balance between speed and torque.

Circuit Design & Implementation



Peripheral Interfacing & Hardware Integration.



Programming ESP32 CAM using FTDI Module.

Software Algorithm

Surveillance Robo Car Using ESP32-CAM

Step 1: Define Wi-Fi Credentials

- Set the SSID and Password for Wi-Fi connectivity.

Step 2: Configure GPIO Pins

- Assign specific GPIO pins for:
- Motor control (Forward, Backward, Left, Right)
- LED light
- Camera module (data lines, sync lines, control lines)

Step 3: Initialize Serial Communication

- Start serial communication at 115200 baud rate for debugging.

Step 4: Set Pin Modes for Motor and LED

- Define the motor and LED pins as OUTPUT.
- Set all pins to LOW initially to prevent any unwanted motion.

Step 5: Configure Camera

- Create a camera_config_t structure.
- Set camera parameters like:
- Pixel format (JPEG)
- Frame size (based on PSRAM availability)
- Clock frequency and GPIO pin assignments

Step 6: Initialize Camera

- Call esp_camera_init() to initialize the camera with the configuration.
- If initialization fails, print the error and exit.

Step 7: Adjust Frame Size

- Access the camera sensor object.
- Set frame size to CIF for better frame rate during live streaming.

Step 8: Connect to Wi-Fi

- Use WiFi.begin() to initiate Wi-Fi connection.
- Wait until connection is successful.
- Print IP address once connected.

Step 9: Start Camera Server

- Call startCameraServer() to initialize the web server.
- The server provides:
- Live video stream through the IP address
- Remote robot control through the web interface.

Step 10: Loop

- Loop () function remains empty unless additional control logic is added later.
- ESP32-CAM starts a web server.
- Users can access live camera feed and control the robot using the assigned IP via browser.

Testing & Result

Stage 1: ESP32-CAM Module Testing

Objective: Verify that the ESP32-CAM can be programmed and stream video.

Steps:

1. Connect ESP32-CAM to FTDI module using correct wiring (Tx-Rx, GND, 5V).
2. Upload test code using Arduino IDE.
3. Check serial monitor for IP address.
4. Open IP in browser to verify live video stream.

Result: Video streaming confirmed; camera module working as expected.

Stage 2: FTDI Programmer Testing

Objective: Ensure successful communication between computer and ESP32-CAM.

Steps:

1. Connect FTDI to PC via USB.
2. Install drivers (if needed).
3. Test with basic "blink" or "serial print" code on ESP32-CAM.

Result: Code uploaded successfully; FTDI working properly.

Stage 3: L298 Motor Driver Testing

Objective: Validate motor driver functionality for bi-directional motor control.

Steps:

1. Connect L298 to 9V battery and DC motor.
2. Send basic forward/reverse control signals from Arduino.
3. Observe motor direction and speed.

Result: Motors rotate correctly; driver responds to control signals.

Stage 4: DC Motor Testing (4× Motors)

Objective: Ensure all motors are functional and have equal RPM.

Steps:

1. Connect each motor directly to power.
2. Observe rotation speed, direction, and noise.
3. Compare RPM and balance across all 4 motors.

Result: All motors functional; slight imbalance adjusted during mounting.

Stage 5: Chassis & Mechanical Integration Testing

Objective: Verify mechanical stability and wheel alignment.

Steps:

1. Mount all components on the plastic box.
2. Fit motors securely and check for wobble.
3. Perform dry run (without camera).

Result: Robot moves smoothly in all directions; alignment is stable.

Stage 6: Full System Integration Testing

Objective: Check complete functionality — video + control.

Steps:

1. Power the entire system (ESP32 + Motors).
2. Connect via Wi-Fi and access the ESP32 video stream.
3. Test controls from web interface.
4. Verify robot responds in real time.

Result: Seamless operation with live video and directional control.

Stage 7: Real-Environment Testing

Objective: Evaluate performance in home/factory-like conditions.

Steps:

1. Operate bot indoors and in low-light conditions.
2. Test Wi-Fi range and video clarity.
3. Monitor heat or power issues during continuous use.

Result: Bot operated efficiently; video quality acceptable; no overheating.

CHALLENGES & LIMITATIONS

During the design and development of the Surveillance Robo, we encountered several technical and implementation-related challenges:

1. FTDI Programming Issues

   - Challenge: Uploading code to ESP32-CAM was inconsistent due to timing issues.
   - Solution: Ensured proper sequence—IO0 pulled LOW and EN reset pressed before upload; used a stable 5V power supply from FTDI.

2. Power Management for ESP32-CAM

   - Challenge: Camera module would reset or fail to boot with motors operating simultaneously.
   - Solution: Used a separate power source for motors and isolated the ESP32-CAM's supply with voltage regulation.

3. Wi-Fi Connectivity Fluctuations

   - Challenge: Unstable video stream due to poor signal strength or interference.
   - Solution: Switched to an external antenna and optimized stream resolution for better performance.

4. Motor Synchronization and Speed Control

   - Challenge: Uneven rotation of DC motors leading to unstable movement.
   - Solution: Balanced load distribution and used PWM for fine motor control via the L298N driver.

5. Limited GPIOs on ESP32-CAM

   - Challenge: Only a few usable GPIOs remained after camera and SD card initialization.
   - Solution: Careful GPIO planning and efficient pin multiplexing to accommodate motor driver inputs.

PROJECT OUTCOMES

This project aligns with and satisfies the following Course Outcomes (COs):

CO1: Design and develop embedded systems using microcontrollers

- Applied ESP32-CAM for real-time surveillance and control applications, integrated with hardware drivers.

CO2: Implement interfacing techniques with actuators and sensors

- Successfully interfaced motor drivers, camera module, and Wi-Fi communication within a unified system.

CO3: Apply engineering principles to solve practical problems

- Resolved hardware-software integration challenges through debugging and iterative development.

CO4: Demonstrate teamwork and project management skills

- Collaborative planning, division of roles, and meeting timelines contributed to the successful completion of Phase I and II.

CONCLUSION & FUTURE SCOPE

This project demonstrated the feasibility of using the ESP32-CAM module to develop a compact and cost-effective mobile surveillance system. The integration of live video streaming, wireless control, and autonomous mobility showcased the potential of IoT and embedded systems in security applications.

This project not only fulfilled academic objectives but also encouraged problem-solving, hardware interfacing, and collaborative learning. Despite challenges such as GPIO limitations, power constraints, and streaming optimization, we were able to achieve reliable performance through iterative testing and technical research

Future Enhancements:

- Add facial recognition or object detection using onboard ML models (such as TinyML with TensorFlow Lite) for intelligent surveillance and automated alerts.
- Integrate GPS for location-based tracking and geo-fencing to restrict or monitor robot movement within specific areas.
- Use ultrasonic or IR sensors for real-time obstacle avoidance, path planning, and autonomous navigation in dynamic environments.
- Implement a mobile application (Android/iOS) for live video streaming, directional control, and notification-based alerts.
- Enable voice control through integration with voice assistants like Google Assistant or Alexa for hands-free operation.
- Add cloud connectivity (IoT platforms like Firebase, Blynk, or ThingsBoard) for storing image/video logs, remote access, and data analysis.

REFERENCES

1. Espressif Systems, ESP32-CAM Datasheet & Technical Reference Manual
2. L298N Motor Driver Module Documentation – STMicroelectronics
3. Arduino IDE with ESP32 Core – https://github.com/espressif/arduino-esp32
4. OV2640 Camera Module Technical Datasheet
5. Online forums: Arduino, Stack Overflow, and ESP32 Community