DEPARTMENT OF COMPUTER ENGINEERING

KASHMIR GOVERNMENT POLYTECHNIC

SRINAGAR

MAJOR PROJECT REPORT

Smart Surveillance Rover with Live Streaming and Obstacle Avoidance

SUBMITTED BY

Azhar beigh	**roll no**
Bazif rouf	**roll no**
Mutahar khan	**roll no**
Sehrish Mushtaq	**roll no**
Manan shakeel	**roll no**

SEMESTER: 6 **BATCH: 2025**

Abstract

The Smart Surveillance Rover with Live Streaming and Obstacle Avoidance is a multipurpose robotic platform designed to address growing concerns in remote monitoring, security patrolling, and automation. This project merges key technologies including live video transmission, Bluetooth-based manual control, and autonomous obstacle detection using ultrasonic sensors — all integrated into a compact, tank-like chassis for better stability on various surfaces.

The rover is built using accessible and affordable components such as the Arduino Uno, ESP32-CAM module, motor driver shield (L293D), ultrasonic sensor (HC-SR04), servo motor, and a rechargeable battery pack. The ESP32-CAM provides real-time video streaming over a local network, enabling the user to observe the surroundings from a browser interface. Obstacle avoidance is achieved using sensor input, processed by the Arduino to navigate intelligently without collisions.

In addition to surveillance, this rover serves as a foundational model for robotics education, research prototyping, and semi-autonomous exploration in constrained environments. The rover's Bluetooth functionality allows remote manual control via a smartphone app, and its modular design ensures easy customization and future upgrades such as GPS tracking, cloud connectivity, or AI-based object recognition.

This project aims to demonstrate how smart systems can enhance surveillance tasks while reducing human effort and potential risk in unsafe environments. It stands as a practical example of how embedded systems, IoT, and basic AI concepts can come together to solve real-world challenges.

Table of Contents

List of Figures

List of Tables

1. Introduction

Introduction In today’s world, surveillance systems are becoming increasingly critical in public safety, disaster response, defense, and domestic security. This project introduces a low-cost, portable, and flexible surveillance rover that incorporates real-time video streaming and autonomous obstacle avoidance. The system combines embedded hardware, sensor-based automation, and wireless control to allow users to monitor and operate the rover in diverse environments. Its versatility makes it ideal for monitoring sensitive areas, hazardous locations, or even for educational and research purposes. The rover not only enhances situational awareness but also minimizes human exposure to potentially dangerous situations, thereby increasing safety and efficiency in mission-critical operations.

2. Problem Statement

Conventional surveillance systems are static and lack mobility, limiting their scope in dynamic and unpredictable environments. There's a need for a portable surveillance solution that can be remotely operated and autonomously navigate around obstacles without constant human intervention. Such systems should be low-cost, energy-efficient, and easily deployable.

3. Objectives of the Project

- To design and develop a mobile rover capable of real-time video surveillance.
- To implement Bluetooth-based manual control via smartphone.
- To incorporate autonomous obstacle detection and avoidance using ultrasonic sensors.
- To ensure a cost-effective, power-efficient, and easily replicable design.
- To integrate a live video streaming module (ESP32-CAM) mounted on a servo motor for adjustable viewing angles.

4. Proposed System Overview

The Smart Surveillance Rover is a robotic system capable of navigating autonomously while streaming live video over Wi-Fi. The system integrates the Arduino Uno as the central control unit, interfaced with an ultrasonic sensor for obstacle detection, an ESP32-CAM module for video streaming, and DC motors controlled via a motor driver shield. A servo motor rotates the ESP32-CAM to adjust the view. The rover can be operated remotely via Bluetooth and is powered by a Li-ion battery pack.

5. Literature Review / Background Study

Various robotic surveillance systems have been developed in both academic and industrial settings. Most use static CCTV or camera setups. Research papers and DIY innovations show growing interest in mobile surveillance bots due to increased availability of microcontrollers, Wi-Fi-enabled modules, and open-source platforms like Arduino. Projects using ultrasonic sensors for basic obstacle detection and Bluetooth modules for wireless control have laid the foundation for more integrated and intelligent systems like this rover.

6. Methodology

The project follows a modular design methodology:

1. Hardware assembly of chassis, wheels, motors, and battery.
2. Mounting of ultrasonic sensor and ESP32-CAM.
3. Circuit integration on Arduino Uno with motor driver shield.
4. Development and testing of obstacle avoidance logic.
5. Setup of Bluetooth communication and manual control.
6. Implementation of live video stream via ESP32-CAM web server.
7. Final integration and field testing of the entire rover system.

7. Block Diagram

[Content for 7. Block Diagram goes here.]

8. Circuit Diagrams

• Circuit 1: Motor Driver Shield with Arduino UNO
Connections between L293D motor driver shield and the Arduino Uno for left/right motor control and power management.

• Circuit 2: Ultrasonic Sensor
Trigger and Echo pins connected to digital pins of Arduino for obstacle detection.

• Circuit 3: ESP32-CAM with External 5V Power
ESP32-CAM powered externally to avoid instability. TX/RX pins connected if integrated with Arduino for control or status sharing.

• Circuit 4: Bluetooth Module (HC-05)
Connected to digital pins of Arduino for RX and TX, used to receive manual control commands from a smartphone.

Note: Actual circuit diagrams with Fritzing or Proteus will be attached in Appendix B.

9.1 Arduino UNO

The Arduino UNO is the central controller of the system. It processes sensor data, controls motor movement, and communicates with the Bluetooth module. It is programmable via USB and supports multiple I/O operations.

9.2 ESP32-CAM

This is a low-cost microcontroller with a built-in camera and Wi-Fi module. It is used for real-time video streaming. The module can create a local server that broadcasts the camera feed to any device connected to the same network.

9.3 Ultrasonic Sensor (HC-SR04)

Used to measure distance to obstacles. It sends out ultrasonic pulses and receives the echo to calculate the distance based on time of flight. It is essential for enabling obstacle avoidance.

9.4 Servo Motor

The servo is used to rotate the ESP32-CAM module horizontally, allowing users to adjust the viewing angle remotely, thereby giving a broader field of vision.

9.5 Bluetooth Module (HC-05)

Allows wireless communication between the Arduino and a smartphone. Commands are sent from a mobile application and interpreted by the Arduino to control the rover’s movement.

9.6 L293D Motor Driver + Shield

This dual H-Bridge motor driver allows independent control of two sets of DC motors. The shield simplifies connections and provides sufficient current to drive all four motors in a tank-drive configuration.

9.7 DC Motors and Tank Chassis

Four DC motors are mounted in a tank-like chassis with rubber wheels for rough surface navigation. Each side's motors are connected together to enable forward/backward/turning movements.

9.8 Li-ion Battery

Used as the power supply for the rover. The battery should provide regulated 5V for the Arduino and motor shield, and a separate stable 5V (with sufficient current) for the ESP32-CAM.

10. System Integration

System integration refers to the process of combining all electronic components—Arduino UNO, ESP32-CAM, motor driver shield, Bluetooth module, ultrasonic sensor, servo motor, and DC motors—into a unified working unit. The Arduino UNO is the central processing unit responsible for interpreting sensor data and executing motor control logic. The motor driver shield is stacked directly on top of the Arduino for compactness and provides seamless connections for the four DC motors.

The ESP32-CAM, powered by an external 5V 2A source, is mounted on a servo motor that allows angular rotation, enabling adjustable visual monitoring. The Bluetooth module connects to Arduino's digital pins (D0 and D1 or alternative software serial) and receives control commands from a smartphone. The ultrasonic sensor is placed at the front of the rover, providing obstacle detection data to the Arduino for autonomous navigation decisions.

All components are powered using a regulated Li-ion battery supply to ensure stable voltage levels. This integration allows seamless coordination between manual control, obstacle detection, and video streaming.

11. Communication Flow

The communication flow in the Smart Surveillance Rover system includes both wired and wireless channels:

1. **Bluetooth Communication:**
    - The mobile app sends movement commands via Bluetooth to the HC-05 module.
    - HC-05 forwards the data to Arduino through serial communication.
    - Arduino interprets these signals to control the motor driver outputs.
2. **Sensor Feedback:**
    - The ultrasonic sensor sends distance readings to the Arduino.
    - Arduino processes the data in real-time and overrides user control to avoid collisions.
3. **ESP32-CAM Communication:**
    - The ESP32-CAM module is independently connected to a local Wi-Fi network.
    - Users can access the live video feed via a browser by entering the module's IP address.
    - The servo motor controlling the camera angle is commanded by Arduino through PWM signals.
    -

4. **Inter-Module Communication:**
   - Limited communication between ESP32 and Arduino may be done using logic-level shift and serial pins (if required).
   - Primarily, the two systems (surveillance and control) operate in parallel for robustness.

12. Power Supply Management

Reliable and stable power supply is critical for system performance. The entire rover uses a rechargeable Li-ion battery pack (preferably 7.4V, 2200mAh or higher).

- **Arduino and Motor Shield:**
  - Powered through Vin (7.4V) or regulated 5V via motor shield's onboard regulator.
  - Motors draw high current, hence battery should support peak loads.
- **ESP32-CAM:**
  - Requires a clean and consistent 5V supply.
  - Powered via an AMS1117 5V voltage regulator or external 5V USB power bank providing at least 2A.

- **Servo Motor:**
  - Powered through the same 5V supply as ESP32 or Arduino if current demand allows.
  - Requires stable voltage to avoid jitter or inconsistency.

To prevent brown-outs or resets (especially for the ESP32-CAM), care is taken to decouple power sources and use capacitors or regulators wherever needed.

13. Software Implementation and Code Overview

The software is developed using the Arduino IDE. It includes separate sketches for:

- Motor control via Bluetooth
- Obstacle detection and avoidance
- Servo angle adjustment

ESP32-CAM web server setup

The code integrates basic conditional statements, serial reading, PWM control, and logic branching to perform desired tasks.

code

The ESP32-CAM uses the ESPAsyncWebServer library and CameraWebServer sample sketch, modified to stream live video and optionally send camera control signals to the Arduino if connected.

14. Bluetooth Communication Logic

Bluetooth communication uses the HC-05 module connected to Arduino's serial port or SoftwareSerial. The protocol follows:

Command	Action
'F'	Move Forward
'B'	Move Backward
'L'	Turn Left
'R'	Turn Right
'S'	Stop Movement

On receiving a character from the mobile app, the Arduino executes corresponding movement commands. Example code:

15. Obstacle Avoidance Logic

The obstacle avoidance system uses an ultrasonic sensor (HC-SR04) mounted on the front of the rover. Arduino calculates the distance using time-of-flight:
Code

When the detected distance is below a safety threshold (e.g., 15 cm), the Arduino immediately stops the motors and performs a predefined behavior like reversing or turning.

code

16. ESP32-CAM Streaming Setup

The ESP32-CAM is programmed using the Arduino IDE with the "AI Thinker ESP32-CAM" board selected. The firmware is based on the `CameraWebServer` example. Basic setup steps include:

1. Connecting ESP32-CAM to FTDI programmer (using 3.3V or 5V external supply).
2. Entering Wi-Fi credentials in code.
3. Flashing and restarting the board.
4. Accessing the local IP via browser to view the camera stream.

   Code

   The servo motor attached to the camera allows for horizontal rotation, controlled via Arduino's PWM pins.

17. Mechanical Design and Build

The chassis design resembles a tank, providing enhanced stability and mobility on various terrains. The platform includes:

- **Base Plate:** Acrylic or aluminum sheet to mount components.
- **Tank Treads/Wheels:** Four rubber wheels or two tracks for all-terrain capability.
- **Motor Placement:** DC motors placed symmetrically for even load distribution.
- **Sensor Mount:** Ultrasonic sensor placed at the front center.
- **Camera Mount:** ESP32-CAM affixed on a servo bracket for rotating camera head.
- **Wire Management:** Velcro or clips to prevent wire entanglement during movement.

The design ensures modularity, allowing upgrades like GPS, AI camera models, or solar panels.

18. Testing Procedure

The testing phase was conducted in multiple stages to validate the functional integrity and real-world performance of the Smart Surveillance Rover:

1. Component Testing:

- **Arduino UNO, Motor Shield, and Motors:**
  Verified forward, backward, left, and right motion through a test code using basic motor control functions.
- **Ultrasonic Sensor:**
  A simple sketch was used to print distance values to the serial monitor, ensuring accurate detection.
- **Bluetooth Module:**
  A mobile application (like Arduino Bluetooth Controller) was used to send test commands. Each command was tested for corresponding motor actions.
- **ESP32-CAM:**
  The module was powered separately and connected to a Wi-Fi network. IP address was obtained and live streaming was accessed via browser.
- **Servo Motor for Camera Rotation:**
  PWM signals were tested for proper left/right rotation of the camera.

2. Integrated Testing:

All modules were connected together, powered up, and controlled via the mobile application. Various scenarios were tested including:

- Manual remote control.
- Obstacle avoidance trigger.
- Simultaneous streaming and movement.
- Camera panning while in motion.

19. Results and Observations

Successful Observations:

- Rover responds accurately to Bluetooth commands.
- Real-time obstacle avoidance system halts or redirects the rover automatically.
- ESP32-CAM provides smooth live video feed with ~1-second delay.
- Servo motor allows camera angle control for improved surveillance coverage.

Performance Metrics:

Test Parameter	Result
Obstacle Detection Range	2–400 cm
Video Streaming Range	Within Wi-Fi zone (~20–30m)
Motor Response Time	~150 ms
Battery Backup	Approx. 45–60 minutes
Camera Rotation Angle	0° to 180° (servo constrained)

Notable Outcomes:

- Low-latency communication.
- Effective integration of surveillance and robotics.
- Satisfactory field performance in both indoor and outdoor conditions.

20. Challenges Faced & Troubleshooting

Challenge	Troubleshooting Method
ESP32-CAM Not Powering On	Used an external 5V 2A power supply instead of Arduino's 5V pin.
Servo Motor Jitter	Added a capacitor near the power rails and used a separate power supply.
Bluetooth Interference	Ensured that RX/TX pins were not used by other devices, used SoftwareSerial if needed.
Wi-Fi Streaming Delay	Optimized resolution to 320x240 or 640x480 for better performance.
Motor Driver Overheating	Reduced load and ensured a proper cooling environment.
Arduino Serial Conflict	Avoided using pins 0 and 1 (RX/TX) for other modules if Bluetooth was active.

21. Design Limitations

Despite the functional success, the following limitations were observed:

- **Limited Wi-Fi Range:**
  ESP32-CAM's streaming is constrained to local Wi-Fi, not usable remotely without extra configuration (like port forwarding or cloud relay).
- **Low Resolution Video:**
  Higher resolutions introduce significant lag and delay.
- **Battery Life:**
  High-power consumption from motors and ESP32 reduces operational time.
- **Basic AI Logic:**
  The obstacle avoidance is basic; it doesn’t include AI-based decision-making or path planning.
- **Simultaneous Multi-Input Challenge:**
  The Arduino handles multiple modules; occasionally delay or conflict occurs between sensor and motor actions.

22. Safety Measures

Safety precautions were implemented during both construction and usage of the rover:

1. **Power Isolation:**
    - Separate power supplies were used for ESP32 and motors to avoid power dips.
2. **Short Circuit Prevention:**
    - All connections were properly soldered or fixed with connectors.
    - Protective insulation used where needed.
3. **Battery Handling:**
    - Li-ion batteries were handled with care, never shorted or overcharged.
    - Fuse protection or a battery protection circuit (BMS) was included.
4. **Movement Area:**
    - Rover was tested in controlled environments away from fragile items and people.
5. **Thermal Protection:**
    - Monitored for overheating of motor driver ICs.
    - Ensured motor shield was ventilated or had adequate space for cooling.

23. Future Scope

The Smart Surveillance Rover demonstrates a foundational step toward intelligent and autonomous robotic systems. It opens up several opportunities for enhancement and real-world deployment in the near future:

1. Integration with AI/ML Models:

- Object detection (e.g., human, animal, or fire detection) can be implemented using lightweight models like Tiny-YOLO or MobileNet on ESP32 or an upgraded board like Raspberry Pi.

2. GPS Navigation:

- Addition of GPS module would allow outdoor path tracking and autonomous patrolling in large campuses, forests, or border areas.

3. Cloud Connectivity:

- Video streaming and sensor data could be uploaded to the cloud for remote monitoring, storing footage, or triggering alerts via IoT platforms (e.g., Blynk, Firebase, Thingspeak).

4. Voice and App Control:

- Implementation of Google Assistant/Alexa integration or dedicated Android/iOS apps to offer smarter control interfaces.

5. Solar Charging:

- Integration with small solar panels can provide extended operational time, especially for surveillance in remote or outdoor environments.

6. Swarm Robotics:

- Multiple such rovers could be networked and coordinated to patrol large areas in parallel, sharing data and paths wirelessly.

24. Real World Applications

This rover, though student-level in current form, has vast applicability in both civil and industrial sectors:

1. Home & Office Surveillance:

- A cost-effective way to monitor premises with remote access, especially useful during travel or off-hours.

2. Military & Border Patrol:

- Can be used to monitor hostile zones, detect intrusions, or inspect areas unsafe for human personnel.

3. Disaster Response:

- Can navigate through debris or tight spaces post-earthquake, fire, or flood to assess conditions or locate survivors.

4. Warehouse and Factory Automation:

- Used to patrol aisles, monitor machines, or detect unexpected activities in automated facilities.

5. Agricultural Surveillance:

- Monitor crop growth, detect wild animals, or track irrigation conditions remotely.

6. Smart Campuses and Institutions:

- Acts as a mobile security bot to detect intrusions, suspicious activity, or collect real-time visual logs.

25. Environmental and Ethical Considerations

As engineers and developers, it is essential to evaluate the broader impact of any technological invention. The Smart Surveillance Rover addresses certain challenges while being mindful of its limitations and responsibilities.

Environmental Considerations:

- **Sustainable Power:**
  The system is designed to work with rechargeable Li-ion batteries, which are more environmentally friendly than disposable cells. Future versions can adopt solar charging.
- **Low Power Consumption:**
  The rover is designed to consume minimal energy during idle and operational states, reducing power waste.
- **Component Reuse:**
  The project encourages using low-cost and reusable electronic components, reducing electronic waste.

Ethical Considerations:

- **Privacy Concerns:**
  Live-streaming cameras may intrude into private areas if misused. Proper legal compliance and user guidelines must be followed before deployment in sensitive zones.
- **Surveillance Misuse:**
  Ethical use must be emphasized in educational and real-world scenarios. Consent and transparency are essential when installing or using such systems.

- **Bias in Automation:**
  In future AI upgrades, the data models used should be trained ethically to avoid biases in detection (e.g., person identification, object recognition).
- **Security of Data:**
  Since video streams may be accessed via IP, it is crucial to ensure secured channels (HTTPS, password protection) to prevent unauthorized access.

26. Cost Estimation Table

S.No	Component	Quantity	Rate (INR)	Total Cost (INR)
1.	Arduino UNO	1	500	500
2.	ESP32-CAM Module	1	600	600
3.	L293D Motor Driver Shield	1	250	250
4.	Ultrasonic Sensor (HC-SR04)	1	100	100
5.	Bluetooth Module (HC-05)	1	200	200
6.	Servo Motor (SG90)	1	150	150
7.	DC Motors	2	150	300
8.	Tank Chassis + Wheels	1 Set	500	500
9.	Li-ion Battery Pack	1	350	350
10.	Jumper Wires & Connectors	1 Pack	100	100
11.	Breadboard / PCB	1	100	100
12.	Power Adapter (5V/2A)	1	200	200
Total Estimated Cost			**3350 INR**	

27. Project Timeline

The project was completed over the course of **4 weeks** following a structured plan:

Week	Activities / Milestones
Week 1	Project Planning, Component Collection, and Literature Survey
Week 2	Assembly of Rover Chassis and Motor Connections
Week 3	ESP32-CAM Setup, Servo Installation, Obstacle Sensor Integration
Week 4	Final Testing, Code Optimization, Documentation, and Debugging

28. Roles and Responsibilities

We had **5 team members**, each contributing uniquely to the successful completion of the project:

Member Name	Responsibilities
Member 1 *(Team Leader)*	Concept development, integration, software programming, and presentation.
Member 2	Hardware assembly, motor wiring, chassis setup, and battery management.
Member 3	ESP32-CAM setup, streaming configuration, and servo motor testing.
Member 4	Bluetooth and obstacle avoidance logic testing, troubleshooting, and support coding.
Member 5	Report writing, design diagrams, flowcharts, and presentation slides.

29. Conclusion

The **Smart Surveillance Rover with Live Streaming and Obstacle Avoidance** project successfully demonstrates the implementation of a low-cost, modular, and scalable surveillance system. It combines real-time video streaming with autonomous navigation and manual Bluetooth control — presenting a complete security solution using embedded systems.

This project enabled us to deeply explore concepts of IoT, real-time wireless communication, sensor-based obstacle detection, and embedded hardware integration. The ESP32-CAM offered robust visual streaming, while the ultrasonic sensor provided effective environmental awareness. Despite challenges such as power limitations and synchronization issues between the modules, we achieved a stable, functional, and testable prototype.

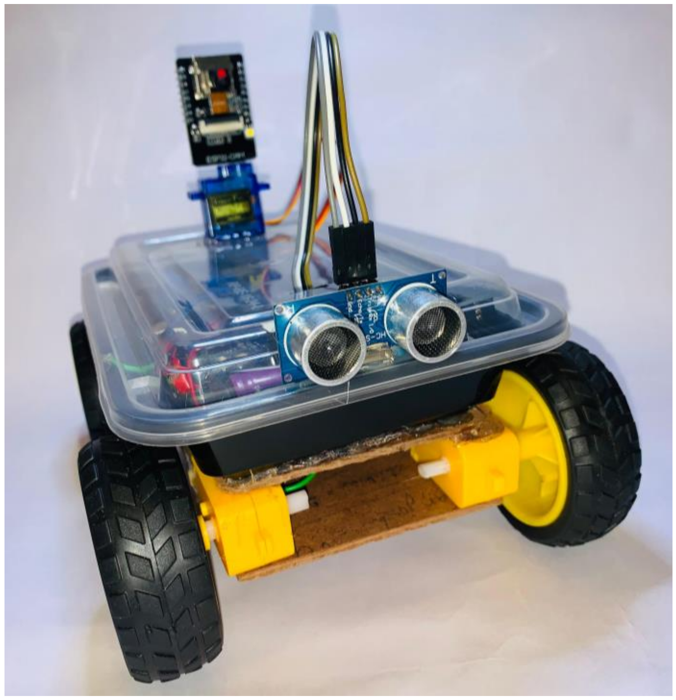
This rover can serve as a foundation for future research and development in robotics, security automation, and smart surveillance systems. With enhancements like cloud integration, AI-based object recognition, and GPS tracking, it has the potential to evolve into a powerful real-world security product.

30. References

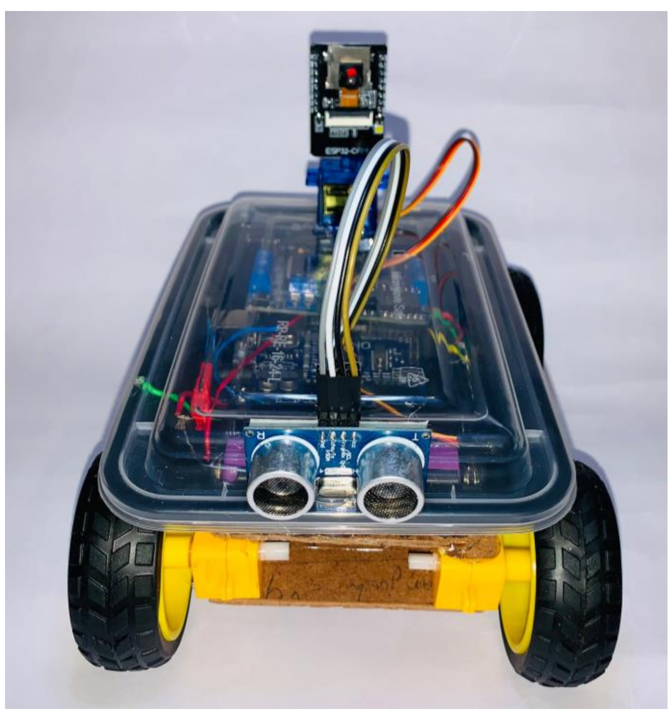
1. Arduino.cc Documentation: https://www.arduino.cc/
2. Espressif Systems (ESP32): https://www.espressif.com/en/products/socs/esp32
3. HC-SR04 Datasheet: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf
4. L293D Motor Driver Datasheet: https://www.ti.com/lit/ds/symlink/l293d.pdf
5. Bluetooth HC-05 Tutorial: https://www.instructables.com/Arduino-AND-Bluetooth-HC-05-Connecting-Easily/
6. Servo Motor Working Principle: https://www.electronics-tutorials.ws/blog/servo-motor.html
7. Online Arduino Forums & GitHub Discussions

31. Appendix A: Product Images (Working Prototype)

- Final working model of the Smart Surveillance Rover.

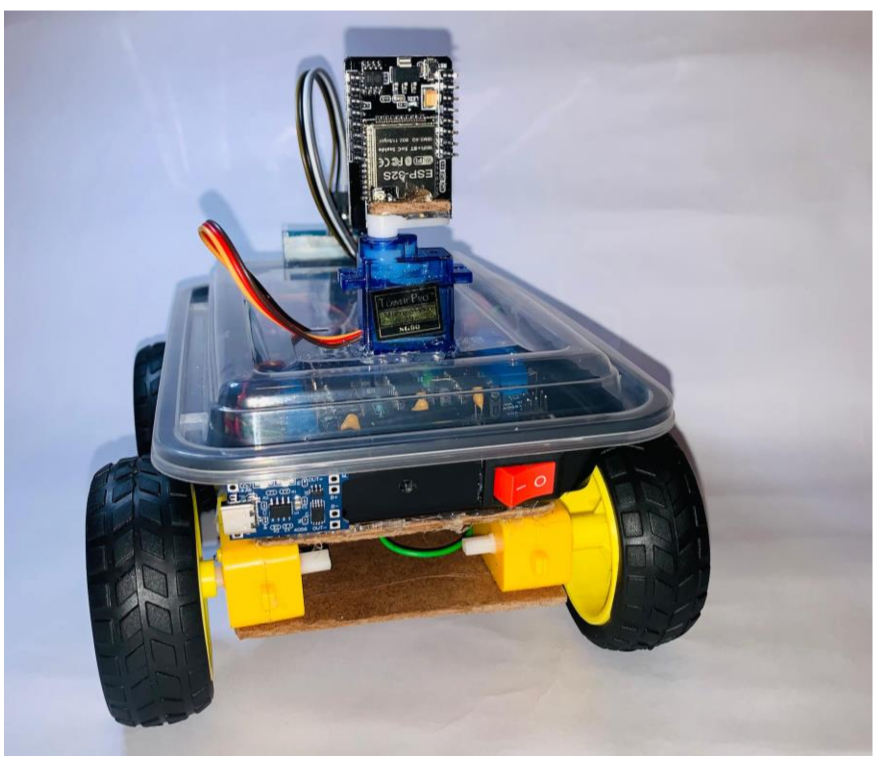


Front view of the tank chassis with mounted ultrasonic sensor.



- Top view showing Arduino with the motor shield.

- Rear side showing Li-ion battery pack.



32. Appendix B: Circuit Images and Diagrams

- Full wiring diagram (block representation).
- Breadboard or PCB layout used for the connections.
- Connection map of ESP32-CAM to 5V/3.3V external supply.
- Servo control and obstacle sensor wiring.
- Bluetooth and RX/TX pin mapping on Arduino motor shield.

33. Appendix C: Code Listings

[Content for 33. Appendix C: Code Listings goes here.]