CHAPTER 1

INTRODUCTION

The Robotic surveillance vehicle for monitoring is an innovative robotic system designed to autonomously navigate rail and uneven surfaces, addressing the limitations of conventional wheeled robots on inclined terrains. This project utilizes an Arduino Uno as the central controller, paired with a variety of components such as DC motors**,** L298N motor driver**,** HC-05 Bluetooth module, GPS GSM module and a camera module for wireless control and real-time monitoring. The robot is powered by a rechargeable battery, ensuring portability and independent operation. With increasing threats to security and a growing need for automated solutions in both public and private spaces, the demand for mobile surveillance robots has seen a significant rise. This project aims to bridge the gap between the need for versatile robotic movement and effective surveillance capabilities by engineering a compact, remotely controllable robot that is capable of climbing and navigating over rails, providing a robust solution for multi-terrain monitoring.

Surveillance in multi-level structures such as buildings, construction sites, warehouses, and critical infrastructure often presents challenges for conventional robots. These systems are typically limited to flat ground and face difficulties when encountering obstacles such as stairways or uneven surfaces. Human patrolling in such scenarios may also be dangerous, expensive, and inefficient. The proposed robotic vehicle offers a safer, cost-effective, and intelligent solution that can maneuver through these complex environments with precision.

Built using the Arduino Uno microcontroller, the robotic system integrates a range of essential components including DC motors for movement, an L298N motor driver for directional control, a Bluetooth module (HC-05) for wireless command reception, and a live camera module for real-time visual feedback. The Arduino Uno acts as the brain of the robot, interpreting control commands and managing motor functions to ensure responsive movement and balance. The integration of a climbing mechanism based on belts and dummy wheels allows the robot to grip and ascend stair edges with stability. Dummy wheels provide extra support and reduce the risk of tipping or imbalance during vertical movements.

The wireless operation is achieved using Bluetooth connectivity, enabling users to control the robot remotely through a smartphone application. Commands are transmitted from the mobile app to the

Bluetooth module, which relays them to the Arduino for processing. Specific characters such as 'F' for forward, 'B' for backward, 'L' for left, 'R' for right, and custom commands for rail climbing are mapped in the code to corresponding motor functions. This approach not only simplifies user interaction but also ensures reliable communication within a defined range.

In addition to movement capabilities, the robot features a compact surveillance camera mounted on its chassis. This camera can be used for real-time monitoring and recording, providing a live video feed to the operator. The inclusion of the camera elevates the functionality of the robot from a simple mobile platform to an integrated surveillance system suitable for real-world applications such as rescue operations, security patrols, and hazardous environment inspections.

The power system is designed with efficiency in mind, using a rechargeable battery unit and a buck converter to maintain stable voltage levels across all components. This ensures uninterrupted operation during long-duration tasks while minimizing the risk of component failure due to power fluctuation. This robotic vehicle is modular in design, allowing for the easy addition of sensors or hardware upgrades without extensive redesign. Its chassis is built using lightweight yet sturdy materials to support mechanical durability while maintaining mobility. The control code, written in embedded C and deployed via the Arduino IDE, enables efficient processing and real-time responsiveness. From an educational standpoint, this project provides an excellent learning platform for students and enthusiasts in the fields of robotics, embedded systems, and automation. It offers practical exposure to hardware-software integration, wireless communication, and mechanical design, encouraging innovation and further development.

The Robotic Surveillance Vehicle for Monitoring is not just a prototype but a scalable solution with broad applicability. Its real-world use cases span across industries including construction, disaster management, military reconnaissance, and home security. In emergency situations such as building collapses or industrial accidents, where human access is restricted or dangerous, the robot can be deployed to gather information, detect life signs, or deliver small payloads.

Furthermore, the growing emphasis on smart infrastructure and the Internet of Things (IoT) creates new opportunities for integrating this robotic system with broader networks for data sharing and automation. With additional developments, the robot can be equipped with IoT modules, enabling cloud connectivity, remote diagnostics, and data analytics to enhance decision-making and predictive maintenance.

1.1 Problem Statement

Surveillance robots are often deployed in areas where human access is limited or dangerous. However, most existing models are restricted to flat surfaces, which limits their usability in complex indoor environments like buildings with multiple floors. There is a growing need for robotic systems that can navigate such environments autonomously or with remote control. This project seeks to address this gap by developing a robotic vehicle that can climb stairs and rails, providing enhanced mobility and surveillance capabilities.

- **Blind Spots**: Stationary systems fail to monitor areas that lie beyond their set range or if obstacles block the camera.
- **Lack of Real-Time Response**: While footage may be recorded, active response to threats or suspicious activity often requires human intervention, which can be delayed.
- **High Operational Costs**: Staffing 24/7 surveillance teams is expensive and requires shifts, training, and coordination.

1.2 Aim and Objectives

The primary aim of this project is to design a robotic surveillance vehicle capable of rail climbing, equipped with wireless control and real-time monitoring capabilities.

Objectives:

- To enable real-time monitoring through an onboard camera.
- To control the vehicle wirelessly using Bluetooth and a smartphone interface.
- To design a climbing mechanism capable of handling standard rails.
- To integrate all electronic components onto a compact mobile chassis.
- To test the robot's performance in real-world conditions for surveillance applications.

CHAPTER 2

LITERATURE SURVEY

1. Title: Rail Develop Autonomous Rail Inspection and Repair System

- **Author:** Sabrina Guillen
- **Year:** 2021
- **Abstract:** The demand for safe and well-maintained railway lines is clear.
  Currently though, inspection and repair tasks on a railway are dangerous and difficult jobs for humans. They include unsociable hours, long distances, exposure to the elements, and a variety of hazards. To maintain an appropriate degree of safety, it is important to reduce the number of people out on the track, while improving the observation and measurement of the railway infrastructure.

2. Title: Railway Inspection Robot Takes on Maintenance Challenges

- **Author:** Shenhao Technology
- **Year:** 2022
- **Abstract:** Railways are large infrastructures and are the prime mode of transportation in many countries. Operators of railway systems must regularly inspect railway tracks as part of maintaining their system as it is closely associated with passenger and cargo transportation. However, there are always challenges to the maintenance department to perform effective inspection and reduce labor cost.
  The application of intelligent railway inspection robot can easily solve the above challenges. It can inspect 24 hours a day without dead angle and transmit the monitoring data in real time for analysing**.**

3. Title: Japanese Railway introduces infrastructure robot

- **Author:** Patrick Rhys Atack
- **Year:** 2024
- **Abstract:** West JR said the project was needed due to “a labour shortage in infrastructure maintenance work, not just for railways”.

  The new “heavy equipment” is carried by a railway construction vehicle, which hosts the control room, with the robot on an extendable boom. It can carry up to 40kg and work remotely at up to 12 metres above ground.

  The robot, which has been likened to Disney Pixar character Wall-E, is operated by a single engineer in the control room. “In addition, the weight and recoil received by the robot are fed back to the operator, allowing intuitive operation.

4. Title: Simultaneous Location of Rail Vehicles and Mapping

- **Authors:** Yusheng Wang, Weiwei Song, Yidong Lou
- **Year:** 2021
- **Abstract:** Precise and real-time rail vehicle localization as well as railway environment monitoring is crucial for railroad safety. In this letter, we propose a multi-LiDAR based simultaneous localization and mapping system for railway applications. Our approach starts with measurements preprocessing to denoise and synchronize multiple LiDAR inputs. Different frame-to-frame registration methods are used according to the LiDAR placement. In addition, we leverage the plane constraints from extracted rail tracks to improve the system accuracy. The local map is further aligned with global map utilizing absolute position measurements. Considering the unavoidable metal abrasion and screw loosening, online extrinsic refinement is awakened for long-during operation. The proposed method is extensively verified on datasets gathered over 3000 km.

5. Title: Development of Railway Track Crack Detection System using Arduino

- **Authors:** Najma Siddiqui, Akil Ahmed
- **Year:** 2023
- **Abstract:** The Indian Railways has one of the largest railway networks in the world, criss-crossing over 1,15,000 km in distance, all over India. However, with regard to reliability and passenger safety Indian Railways is not up to global standards. Among other factors, cracks developed on the rails due to absence of timely detection and the associated maintenance pose serious questions on the security of operation of rail transport. A recent study revealed that over 25% of the track length is in need of replacement due to the development of cracks on it. Manual detection of tracks is cumbersome and not fully effective owing to much time consumption and requirement of skilled technicians. This paper is aimed towards addressing the issue by developing an automatic railway track crack detection system with the proliferation of Internet of Things (IoT).

2.1 Key Observations from the Literature

1. Shift from Manual to Autonomous Systems
    - Traditional rail inspection is dangerous, labor-intensive, and inefficient (exposure to hazards, unsociable hours).
    - Robotic solutions (e.g., autonomous drones, rail-crawling robots) are emerging to reduce human risk and improve efficiency.
2. 24/7 Monitoring & Real-Time Data
    - AI-powered robots (e.g., Shenhao’s inspection bot) can scan tracks continuously without blind spots.
    - LiDAR-based SLAM (Wang et al., 2021) enables real-time rail mapping with sub-meter accuracy.
3. Labor Shortages Driving Automation
    - Japan’s West JR introduced remote-controlled repair robots due to workforce shortages.

- Single-operator systems (like Wall-E-like bots) reduce manpower needs while improving precision.

4. IoT & Low-Cost Automation
    - Arduino-based crack detection (Siddiqui & Ahmed, 2023) shows cost-effective IoT solutions for rail defects.
    - AI + sensor fusion (LiDAR, cameras, ultrasonic) improves defect identification over manual checks.
5. Challenges in Implementation
    - Sensor reliability (metal abrasion affects LiDAR calibration).
    - Heavy-duty robotics (e.g., Japan's 40kg-capable repair bot) require stable power & mobility.

2.2 Limitations on Existing Survey

Table 1: Limitations

Study	**Primary Limitation**	**Impact**
Wang et al. (2021)	LiDAR calibration drifts over long distances	Reduces SLAM accuracy beyond 3000 km
Shenhao Tech (2022)	No solution for curved/turnout tracks	Partial network coverage
West JR (Atack, 2024)	40kg payload limits tool versatility	Cannot handle heavy repairs
Siddiqui & Ahmed (2023)	Arduino-based system lacks precision	Miss micro-cracks (<0.5mm)

CHAPTER 3

METHODOLOGY

The development of the Robotic Surveillance Vehicle followed a structured process, beginning with identifying key requirements such as mobility over stairs and rails, real-time video monitoring, wireless control, and GPS tracking. The Arduino Uno microcontroller was chosen for its simplicity, low cost, and compatibility with various modules.

Mechanically, a lightweight yet durable chassis was designed using aluminium. A belt-driven climbing mechanism with dummy wheels was integrated to ensure the robot could navigate uneven surfaces and stairs with stability. For movement, DC motors were controlled via an L298N motor driver, with commands transmitted through an HC-05 Bluetooth module connected to a smartphone.

Surveillance functionality was enabled using a CP Plus camera, providing live video via the ezyLiv mobile app. GPS tracking was incorporated using a NEO-6M module for real-time location awareness. Power was supplied through a rechargeable 12V battery with a buck converter to regulate voltage for various components. The robotic surveillance vehicle was developed using Arduino Uno, DC motors, Bluetooth, GPS, and a camera. It features a belt-driven mechanism for stair and rail climbing. Controlled via smartphone, it streams live video and track location. The system is battery-powered, lightweight, and tested for stability, mobility, and real-time monitoring. All systems were integrated with modular wiring and programmed using the Arduino IDE. The robot was tested on flat surfaces, stairs, and rail tracks to ensure stable movement, reliable Bluetooth communication, and effective real-time video streaming. Adjustments were made for motor speed and mechanical balance to optimize performance in real-world conditions.

3.1 Block Diagram

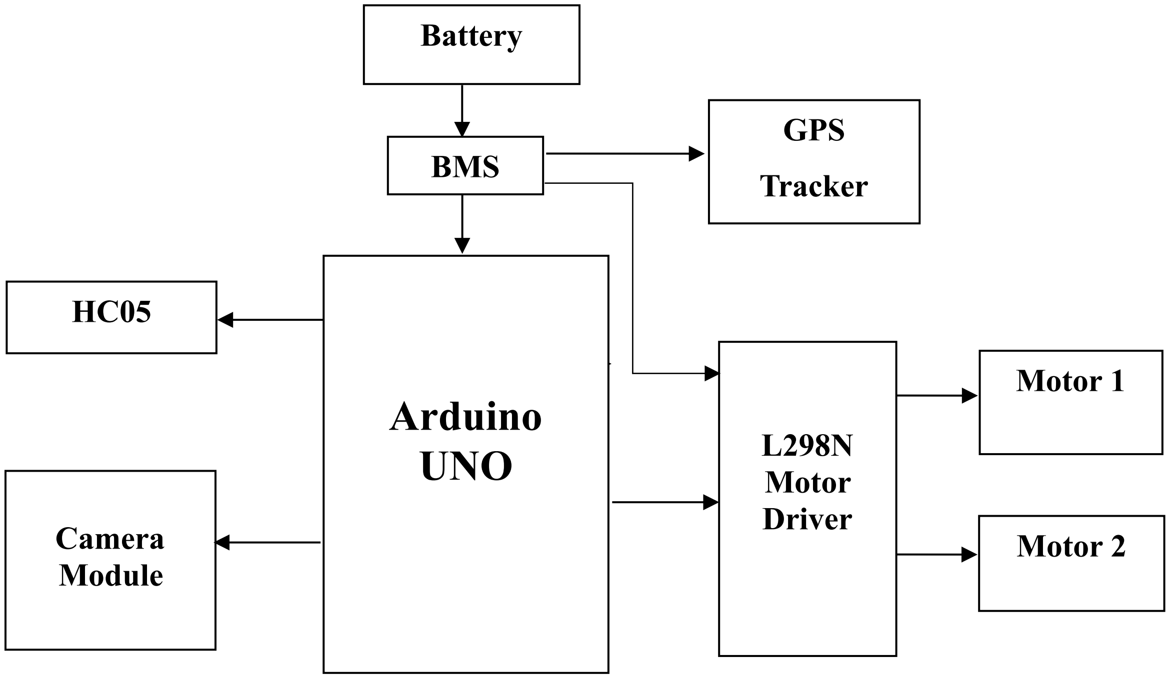


Fig 3.1 Block diagram of Robotic surveillance vehicle for monitoring

The Robotic Surveillance Vehicle represents an innovative integration of mobility, surveillance, and communication technologies designed for comprehensive monitoring applications. At its core, the system combines robust mechanical design with intelligent electronic control to navigate challenging environments while maintaining continuous operational awareness.

Powering the entire system is a carefully engineered 12V Li-ion battery pack with sophisticated power conditioning that delivers stable voltage to all subsystems. This energy source enables extended mission durations while supporting the varied power requirements of computational, sensory, and mobility components. The heart of the vehicle's intelligence lies in its Arduino-based control unit, which processes real-time sensor data, executes motion algorithms, and manages all wireless communication protocols simultaneously. The vehicle's mobility system features a precisely calibrated arrangement of high-torque DC motors coupled with an advanced motor driver that enables both smooth surface navigation and reliable stair-climbing capability. This mechanical foundation supports the sophisticated surveillance package, which includes an HD camera with infrared capabilities for around-the-clock monitoring, complemented by comprehensive data recording and transmission systems.

A dual-layer communication architecture combines short-range Bluetooth control with long-range GSM connectivity, creating a resilient command and feedback network. This is enhanced by precise GPS tracking that maintains constant location awareness. During operation, the system executes a sophisticated workflow that begins with secure command reception, progresses through optimized motion execution with real-time power monitoring, and culminates in comprehensive environmental surveillance with continuous status reporting. These features are unified through carefully implemented fail-safe protocols that ensure reliable operation even in challenging conditions. Together, these elements create a versatile surveillance solution capable of addressing diverse operational requirements while maintaining high reliability and performance standards.

3.2 Functional Description

The functionality of the Robotic Surveillance Vehicle is distributed across several interconnected modules, each with a specific role in achieving the overall objectives of mobility, monitoring, and remote control.

3.2.1 Movement and Navigation Subsystem

- **DC Motors**: Two high-torque 12V DC motors are used to drive the robot's wheels. These are connected to a dual-channel L298N motor driver.
- **Motor Driver (L298N)**: Acts as an interface between the Arduino and the motors, receiving control signals and supplying the necessary voltage/current to operate the motors. The L298N is a dual H-Bridge motor driver that allows control of DC motors and pumps using low-power signals from a microcontroller like Arduino.
- **Control Commands**: The movement is controlled using predefined commands sent via Bluetooth:
    - F – Move Forward
    - B – Move Backward
    - L – Turn Left
    - R – Turn Right
    - S – Stop

- **Stair/Rail Climbing**: The robot is mechanically designed with a **belt and dummy wheel** system that enables it to climb stairs or rails, providing additional traction and stability.

3.2.2 Central Control Unit (Arduino Uno)

- The **Arduino Uno** serves as the brain of the robot.
- It is responsible for:
  - Interpreting commands from the Bluetooth module.
  - Sending motor control signals to the L298N driver.
  - Managing data communication with optional modules such as GSM and GPS.
  - Maintaining power regulation and system logic.
- The microcontroller is programmed using **Embedded C** in the **Arduino IDE**.

3.2.3 Wireless Communication Subsystem

- **HC-05 Bluetooth Module**:
  - Facilitates short-range wireless communication between the robot and the user's smartphone.
  - Uses UART (TX/RX) serial communication to send commands to the Arduino.
  - Paired with mobile apps like "Bluetooth Terminal" for manual control.
- **Communication Range**: Typically, 10 meters in open space.

Pairing HC-05 with Mobile (Android/iOS)

Steps:

1. **Power ON** the Arduino with **HC-05** connected.
2. Open **Bluetooth Settings** on your phone.
3. Search for new devices & select **"HC-05".**
4. Enter **default PIN: 1234 or 0000** and pair the device.
5. Once paired, install a **Bluetooth Terminal App** (e.g., "Serial Bluetooth Terminal" from Play Store).

Testing Connection (Using Serial Bluetooth Terminal)

For Android:

1. Open the **Bluetooth Terminal App**.
2. Select the paired **HC-05 module**.

3.2.4 Surveillance Subsystem (Camera Module)

- A **Wi-Fi-enabled CP Plus camera** is mounted on the robot.
- It provides:
  - **Real-time video streaming:** via a mobile application (e.g., ezyLiv).
  - **Remote surveillance capabilities:** allowing users to visually monitor the robot's surroundings.
- The camera operates independently from the Arduino (as Arduino lacks video processing capability) but shares the power supply.
- **Role in the Project:** The CP Plus camera serves as the primary vision and monitoring system in the robotic surveillance vehicle, enabling real-time data capture, remote monitoring, and environmental analysis

3.2.5 Location Tracking and Remote Alerting

- **GSM Module**: The GPS (Global Positioning System) tracker in this project provides real-time location tracking, enabling the robotic vehicle to transmit its coordinates for remote monitoring. This is critical for surveillance, search-and-rescue, and security patrols where knowing the robot's exact position is essential.
  - Can be programmed to send location coordinates or status updates.
- **GPS Module (NEO-6M)**:
  - Provides real-time location data.
  - Useful for tracking the robot in outdoor or large-scale environments.
- These modules enhance the robot's remote accessibility and situational awareness.
- **Role in the Project:** The GPS tracker serves as the geo-awareness system of the robotic surveillance vehicle, enabling location intelligence, route tracking, and mission coordination.

3.2.6 Power Supply Subsystem

- A **12V rechargeable Lithium-ion battery** is used to power all components.
- A **buck converter** is used to step down voltage for devices requiring 5V or 3.3V.
- Power regulation ensures:
  - Consistent voltage supply across all components.
  - Protection against over-voltage or under-voltage conditions.
- Optional **LED indicators** may be included to show battery level and system status.
- **Role in the Project**: The power supply ensures that the Arduino Uno and other components have the necessary voltage and current to operate properly.

3.2.7 Software Functionality

1. **Arduino software:** The Arduino software is programmed using Embedded C in Arduino IDE. Main code functions include:
   - Receiving and interpreting Bluetooth commands.
   - Controlling motor direction and speed using PWM.
   - Managing serial communication with GSM/GPS modules.
   - Providing logic control for mode switching or fail-safe actions.

Fig 3.2 Arduino IDE Software

2. **Bluetooth RC Car Application:** The Bluetooth RC Car Application serves as a dedicated Android control interface designed specifically for wireless operation of Arduino-based robotic vehicles. This specialized mobile application establishes seamless communication with the robotic system through standard HC-05/HC-06 Bluetooth modules, creating a reliable wireless control channel.

How It Works

1. Pairing
    - Turn on the robot’s HC-05 module.
    - Open the app → Scan/select the Bluetooth device (e.g., "HC-05").
2. Control Commands
    - Press buttons to send predefined characters (e.g., 'F' = Forward, 'B' = Backward).
    - Arduino decodes these to control motors via the L298N driver.
3. Customization
    - Edit the app’s source code (if open-source) to add new functions.
    - Example: Send 'C' to capture a photo via the robot’s camera.

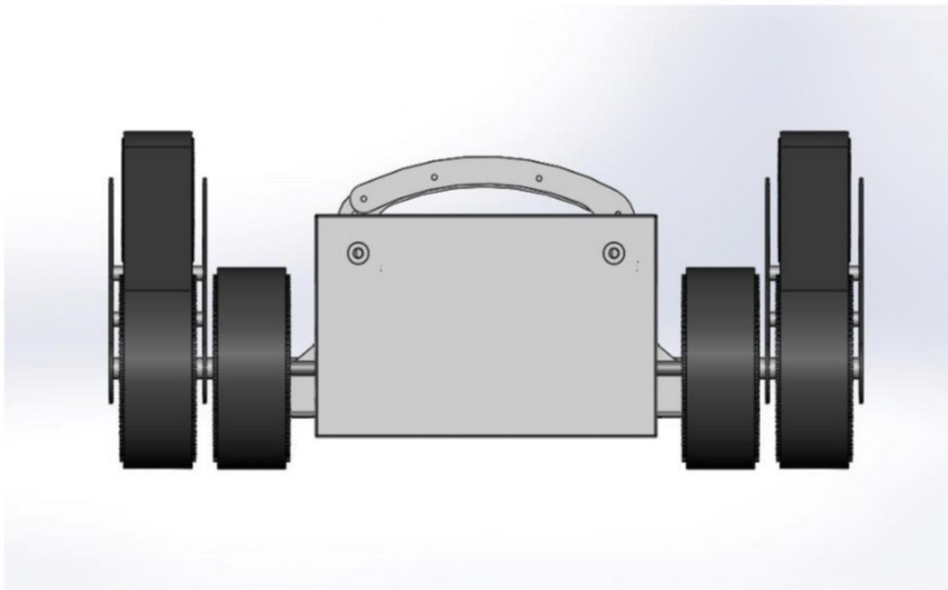
3. **GPS Tracker - Google Find Device:** Google's Find My Device service, accessible on any computer or mobile device, allows you to locate, secure, and even erase your lost Android device. To use it, simply sign in to your Google account on android.com/find, which is linked to your lost phone. The service uses your device's location to display its approximate location on a map.

CHAPTER 4

DESIGN AND IMPLEMENTATION

4.1 3D Model

Front View



Side View

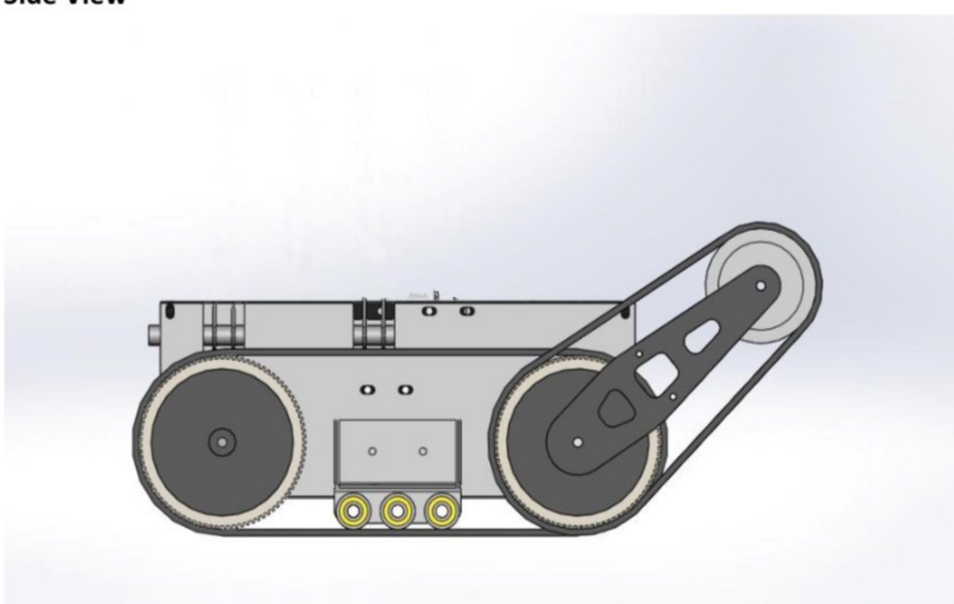


Fig 4.1 3D model

4.2 Components

Table 2: List of Components

SL No.	Component	Specification	Quantity	Purpose
1	Arduino Uno	ATmega328P Microcontroller Board	1	Acts as the central processing unit for the entire system
2	DC Motor	12V, High Torque	2	Provides movement to the robot
3	L298N Motor Driver	Dual H-Bridge, 2A	1	Controls direction and speed of DC motors
4	HC-05 Bluetooth Module	3.3V - 5V compatible, Serial Communication	1	Enables wireless control via smartphone
5	GPS Module	U-Blox with antenna	1	Provides real-time location tracking
6	Camera Module (CP Plus)	WIFI/Bluetooth supported IP Camera	1	Provides real-time video surveillance
7	Li-ion Battery Pack	12V Rechargeable	1	Powers the entire robot
8	Buck Converter	Step-down voltage regulator (12V to 5V/3.3V)	1	Ensures stable voltage to all components
9	Chassis Frame	Lightweight Aluminium/Plastic	1	Mechanical support for components and motors

10	Belts & Wheels	Rubber-based, suitable for stair/rail climbing	As needed	Climbing support mechanism for stairs/rails
11	Jumper Wires	Male-Male, Male-Female	Several	Used to connect different modules to Arduino
12	Switches	ON/OFF toggles	1–2	For power control and manual resets

4.2.1 Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328P and is the central controller in this robotic surveillance vehicle project. It coordinates all hardware components, including sensors, motors, Bluetooth, GPS, and camera modules. The Arduino reads inputs, executes logic, and sends output signals to drive the robot's operations.

In this project, the Arduino Uno processes commands received via the HC-05 Bluetooth module, enabling wireless control of the robot. It also interfaces with the L298N motor driver to control the movement of DC motors, allowing the robot to navigate various terrains. The integration with the GPS module provides real-time location tracking, which is critical for monitoring and remote operations. With 14 digital I/O pins, 6 analog inputs, and robust functionality, the Arduino Uno is ideal for embedded system applications like this one

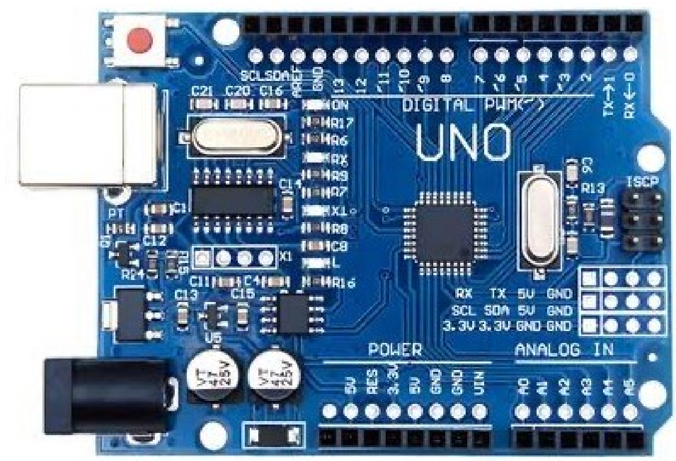


Fig.4.2 Arduino Uno

4.2.2 DC Motors

A DC (Direct Current) motor is an electromechanical device that converts electrical energy into rotational motion. In this project, DC motors are used to drive the wheels of the robotic vehicle. They provide the necessary torque and speed to move the robot forward, backward, and turn. Their simplicity, cost-effectiveness, and ease of control make them ideal for mobile robotics applications.

Its round shaft ensures secure coupling with various mechanical systems, while its compact and durable design allows for easy integration in tight spaces. The motor is known for its smooth performance, low noise, and high efficiency, making it suitable for applications like robotic arms, conveyor systems, medical devices, and automotive mechanisms. With optional IP-rated protection, some models can withstand dust and moisture, enhancing their versatility in different environments.

Fig.4.3 DC (Direct Current) motor

4.2.3 L298N Motor Driver

The L298N Motor Driver is a popular dual H-bridge module capable of controlling two DC motors or one stepper motor, making it ideal for robotics and automation projects. It supports a wide voltage range (5V to 35V) and can deliver up to 2A per channel (peak 3A), allowing it to drive medium-power motors like the Johnson Geared Motor 200RPN. The module features built-in protection diodes to prevent back EMF damage and includes an onboard 5V regulator (which can power an Arduino if enabled via a jumper). Key control pins (IN1, IN2, IN3, IN4) determine motor direction, while ENA and ENB (PWM-enabled) adjust speed via Arduino's analog signals. The L298N is commonly used with microcontrollers like the Arduino Nano for bidirectional motor control, offering simple wiring and reliable performance. However, it requires external heat sinks for high-current applications due to power dissipation. Its versatility, affordability, and ease of integration make it a staple in DIY robotics, CNC machines, and wheeled robot projects.

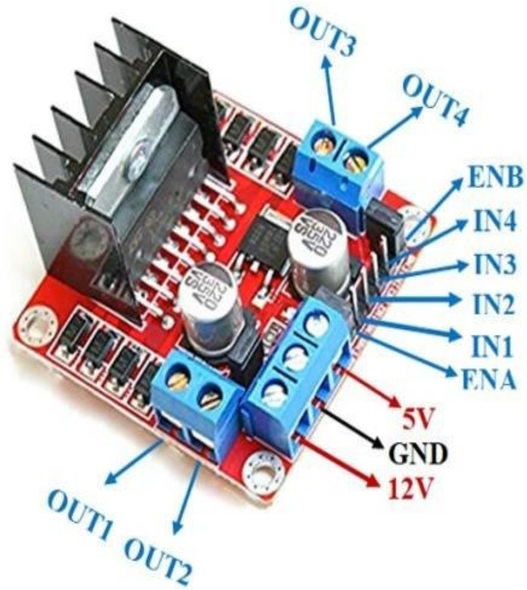


Fig.4.4 L298N Motor Driver

4.2.4 HC-05 Bluetooth Module

The HC-05 Bluetooth Module is a versatile, low-cost Bluetooth 2.0+EDR (Enhanced Data Rate) transceiver widely used for wireless serial communication between microcontrollers (like Arduino, ESP8266) and smartphones/PCs. Operating at 3.3V–5V logic levels (with onboard voltage regulation), it supports UART (Serial) communication at default baud rates (commonly 9600 or 38400 bps) and can function in Master, Slave, or Loopback modes, configurable via AT commands. With a typical range of 10 meters (class-2 power) and 2.4GHz frequency, it pairs effortlessly with devices like Android/iOS apps (e.g., Bluetooth Terminal). The module features an embedded antenna, status LEDs (red/green for connection/power), and 6 pins (VCC, GND, TX, RX, KEY, STATE)—where KEY toggles AT-command mode (held high during power-up) and STATE indicates pairing status.

Commonly used in robotics (remote control), IoT projects, wireless sensors, and data logging, the HC-05 draws ~30mA in operation. Note: Logic-level shifting (5V→3.3V) may be needed for Arduino Nano compatibility to prevent signal damage. For setup, libraries like Software Serial simplify Arduino integration.

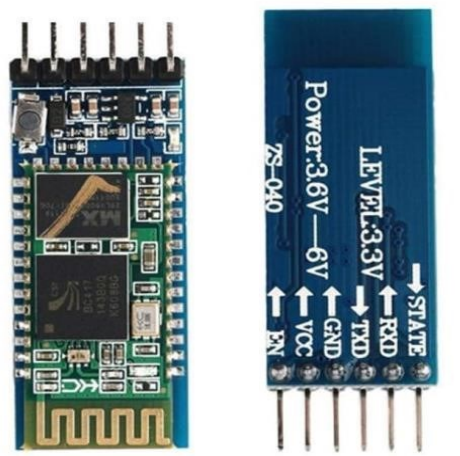


Fig.4.5 HC-05 Bluetooth Module

Table 3: HC-05 Bluetooth Module Pin Connections (with Arduino)

HC-05 Pin	**Arduino Pin**	**Description**
VCC	**5V**	Power Supply (Use 3.3V regulator if needed)
GND	**GND**	Common Ground
TX	**2 (RX)**	Bluetooth Transmit (TX) to Arduino Receive (RX)
RX	**3 (TX) via Voltage Divider (Optional)**	Bluetooth Receive (RX) to Arduino Transmit (TX) (Use voltage divider to convert 5V to 3.3V)
EN/KEY	**Not Used (Optional for AT Mode)**	Enable AT Command Mode

4.2.5 Camera Module (CP Plus)

The CP Plus camera module is a compact surveillance camera used for real-time video monitoring. It is mounted on the robot and connected via Wi-Fi or Bluetooth, providing a live feed to the user's smartphone or remote device. The camera offers high-resolution imaging and may include features like night vision, making it useful for surveillance in both daylight and dark environments.

- High-resolution video capture (up to 1080p Full HD) for clear surveillance footage
- Wireless connectivity via Wi-Fi and/or Bluetooth for remote access
- Wide-angle lens to cover a larger field of view
- Motion detection capability for real-time alert and recording triggers
- Supports remote monitoring through mobile apps like **ezyLiv**
- Cloud storage support for saving video footage remotely (model-dependent)
- Compact and lightweight, ideal for mobile robotic platforms
- Low power consumption, suitable for battery-operated systems

Fig.4.6 CP Plus camera module

4.2.6 Li-ion Battery Pack (9v)

The 18650 Li-Ion battery is a widely used rechargeable lithium-ion cell, named for its 18mm diameter and 65mm length. Known for its high energy density, it typically provides 3.7V nominal voltage (4.2V when fully charged) and capacities ranging from 2000mAh to 3500mAh, with high-drain variants supporting 10A+ discharge currents. These batteries power devices like laptops, flashlights, electric vehicles (EVs), and DIY electronics (e.g., Arduino/robotics projects). Key advantages include long cycle life (~500 charges), lightweight design, and stable voltage output, though they require protection circuits (PCB) to prevent overcharge, over-discharge, and short circuits. Popular in power banks and solar setups, 18650s are often grouped in series/parallel for higher voltage/capacity. Handling precautions include avoiding piercing, extreme temperatures, or improper charging (use dedicated Li-Ion chargers like TP4056). Compared to alternatives (e.g., 21700 cells), 18650s remain a cost-effective choice for balanced power and portability.

Fig.4.7 Li-Ion Battery pack

4.2.7 Wheels and track belts

Wheels and track belts are two fundamental locomotion systems used in robotics and vehicles, each offering distinct advantages depending on the application and terrain.

Wheels are circular mechanical components that allow for fast, efficient movement, especially on smooth, flat surfaces. They are commonly used in mobile robots, carts, and vehicles where speed, energy efficiency, and simple control are priorities. Due to their low rolling resistance and ease of turning, wheels are ideal for indoor or paved environments.

In contrast, track belts (also known as continuous tracks or caterpillar tracks) consist of a flexible belt driven by a system of wheels or sprockets. This design enables better traction and stability on uneven, soft, or slippery terrain. Track belts distribute the vehicle's weight over a larger surface area, reducing ground pressure and making them suitable for off-road or rugged environments such as construction sites or exploration robots.

Fig.4.8 Wheels and track belts

4.2.8 GPS Module

The GPS module is a compact, low-cost u-blox-based receiver offering 50-channel satellite tracking, 2.5m accuracy, and support for GPS/GLONASS/Galileo/BeiDou (firmware dependent). It operates at 5V with UART (9600 bps default) output, providing real time NMEA-0183 data (latitude, longitude, speed, UTC time) via its ceramic patch antenna (or external MMCX antenna). With 1Hz update rates (configurable to 5Hz), 67mA power consumption, and hot/warm/cold start modes, it's ideal for drones, trackers, robotics, and IoT projects.

Compatible with Arduino/ESP32/RPi, it's widely used in navigation, geofencing, and data logging. Libraries like Tiny GPS++ simplify integration.

Role in the Project: The GPS tracker serves as the geo-awareness system of the robotic surveillance vehicle, enabling location intelligence, route tracking, and mission coordination.

Fig.4.9 GPS Module

4.3 Circuit Diagram

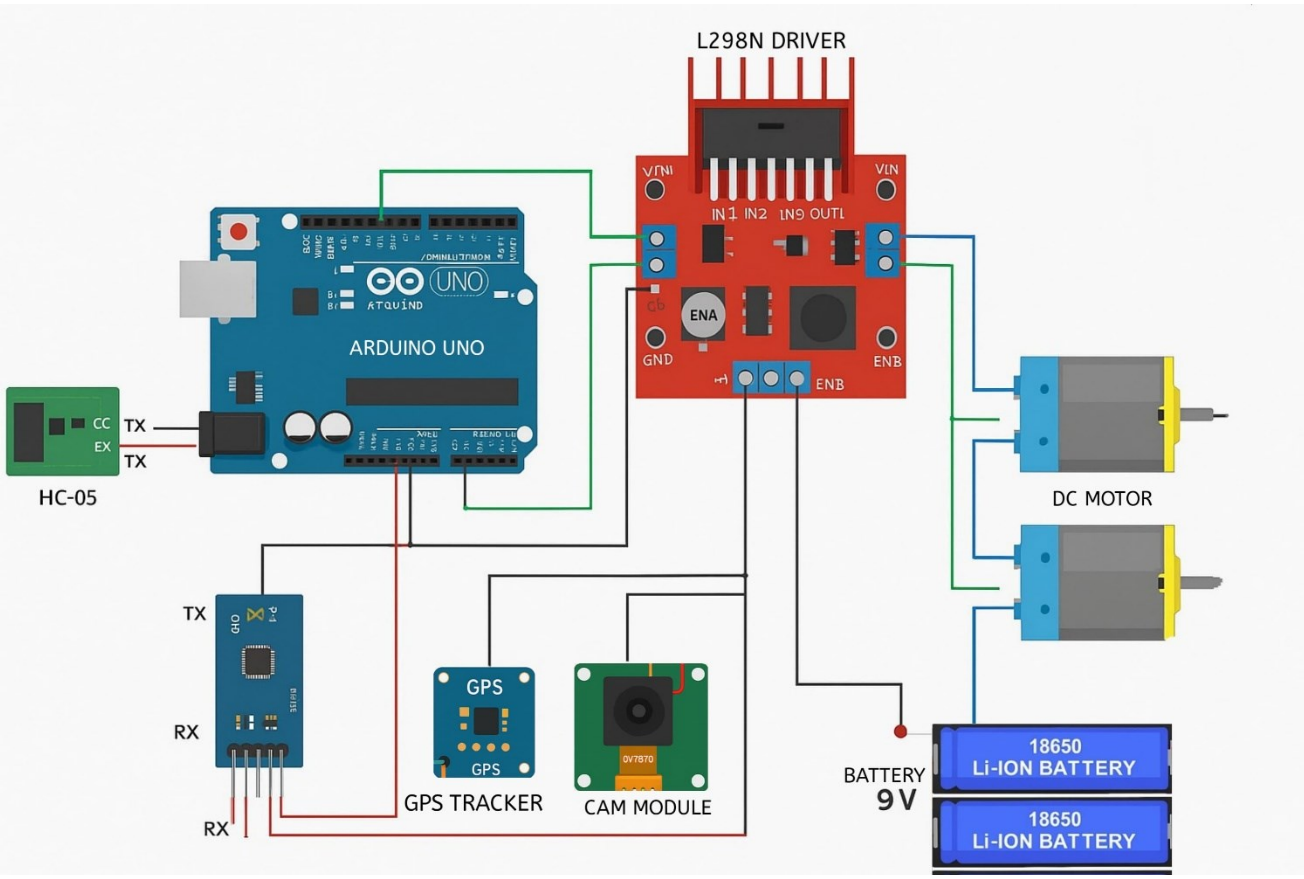


Fig 4.10 Circuit Diagram of control and processing unit

4.3.1 Hardware Description and PIN Configuration:

1. Arduino Uno (Main Controller)

The Arduino Uno is responsible for processing inputs, controlling outputs, and communicating with all connected modules.

Digital Pins Used:

Pin 2: RX for HC-05 (via software serial)

Pin 3: TX for HC-05

Pin 4: IN1 (L298N Motor Control)

Pin 5: IN2 (L298N Motor Control)

Pin 6: ENA (Motor A Enable - PWM for speed control)

Pin 7: IN3 (Optional - for Motor B if used)

Pin 8: IN4 (Optional - for Motor B if used)

Pin 9: ENB (Motor B Enable - PWM)

Power Pins:

5V and GND: Power supply to modules such as the GPS and HC-05.

2. HC-05 Bluetooth Module

Used for wireless communication between the robot and smartphone.

VCC → 5V (Arduino)

GND → GND (Arduino)

TX → Arduino Pin 2 (RX via SoftwareSerial)

RX → Arduino Pin 3 (TX via voltage divider)

Note: A voltage divider or level shifter is used to drop Arduino's 5V TX to 3.3V for HC-05 RX.

3. L298N Motor Driver

Controls the direction and speed of two DC motors based on commands from Arduino.

IN1 → Arduino Pin 4

IN2 → Arduino Pin 5

ENA → Arduino Pin 6 (for PWM speed control)

GND → Common Ground

VCC (VIN) → 9V Battery

OUT1 & OUT2 → DC Motor terminals

ENB, IN3, IN4 → Optional (for second motor channel)

4. DC Motors (2 Units)

Receive power and direction control signals from the L298N motor driver.

Connected to OUT1 & OUT2 and OUT3 & OUT4 on the motor driver.

5. GPS Module

Provides location tracking using satellite communication.

VCC → 5V (Arduino)

GND → GND (Arduino)

TX → Arduino Pin 10 (RX via SoftwareSerial)

RX → Arduino Pin 11 (TX via SoftwareSerial)

SoftwareSerial library is used to manage multiple serial devices.

6. Camera Module (CP Plus)

Used for live surveillance; connected to power for operation.

VCC → 5V or external supply

GND → Common ground

Streams live video over Wi-Fi/Bluetooth to mobile using apps like ezyLiv.

7. Power Supply (9V Battery)

Provides power to:

Arduino via VIN

L298N Motor Driver via VIN

Camera module and GPS module via regulated 5V

4.3.2 Software Description:

1. Bluetooth RC Car:

The Bluetooth RC Car Application serves as a dedicated Android control interface designed specifically for wireless operation of Arduino-based robotic vehicles. This specialized mobile application establishes seamless communication with the robotic system through standard HC-05/HC-06 Bluetooth modules, creating a reliable wireless control channel with an effective range of approximately 10 meters in open spaces. The intuitive user interface features a responsive control panel with tactile directional buttons for precise forward, backward, left, and right movement commands, complemented by a dynamic speed adjustment slider that allows real-time PWM modulation for smooth acceleration and deceleration control.

Fig 4.11 Bluetooth RC car Application

How It Works

4. **Pairing**
    - Turn on the robot's HC-05 module.
    - Open the app → Scan/select the Bluetooth device (**e.g., "HC-05"**).
5. **Control Commands**
    - Press buttons to send **predefined characters** (e.g., 'F' = Forward, 'B' = Backward).
    - Arduino decodes these to control motors via the **L298N driver**.
6. **Customization**
    - Edit the app's **source code** (if open-source) to add new functions.
    - Example: Send 'C' to capture a photo via the robot's camera.

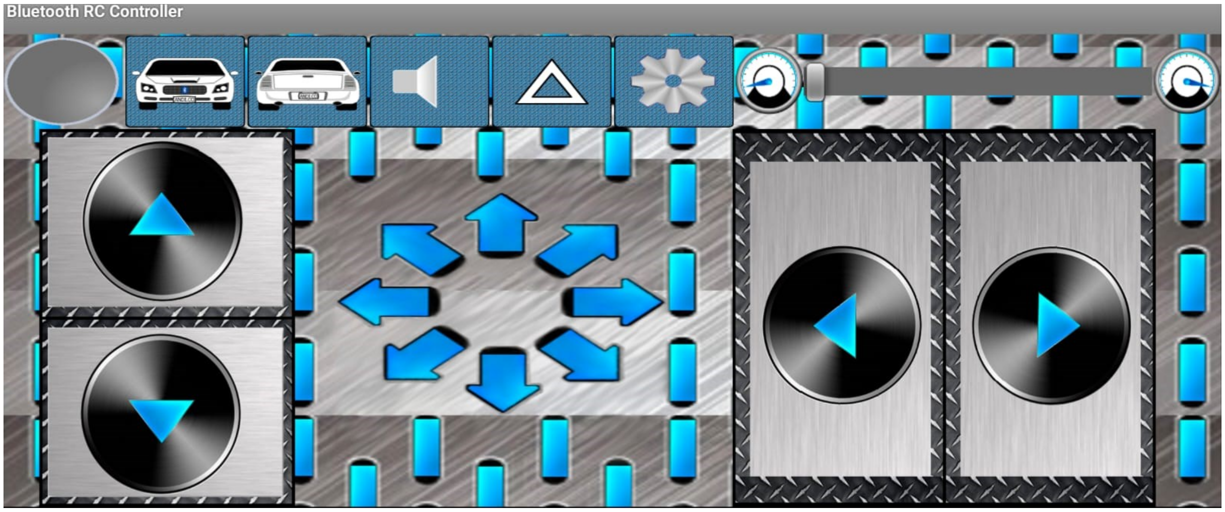


Fig 4.12 Control Interface

2. ezyLiv Application (APK):

ezyLiv is a mobile app, primarily for Android, that allows users to view and control video streams from their cameras. It's developed by CP Plus and falls under the Business & Productivity category, enabling users to monitor and control live video feeds remotely

Fig 4.13 ezyLiv Application

Technical Specifications

- **Compatibility**: Android 8.0+ (optimized for tablets)
- **Connection Protocols**: WiFi (2.4GHz/5GHz), Bluetooth 5.0
- **Cloud Services**: AWS-backed infrastructure
- **API Support**: RESTful API for custom integrations

For Robotic Vehicles

When adapted for robotic control:

- Provides joystick-style driving interface
- Displays real-time telemetry (battery, signal strength)
- Supports waypoint navigation for autonomous robots

3. GPS Tracker - Google Find Device:

Google's Find My Device service, accessible on any computer or mobile device, allows you to locate, secure, and even erase your lost Android device. To use it, simply sign in to your Google account on android.com/find, which is linked to your lost phone. The service uses your device's location to display its approximate location on a map.

Fig 4.14 Google Find Device

How AirTag Works for Tracking

- Attach AirTag to Robotic vehicle.
- Uses Bluetooth and Ultra Wideband to locate the AirTag with nearby Apple devices (via the Find My network).
- If it's nearby, you can make it play a sound.
- If it's far, you can see its last known location on the map using the Find My app.

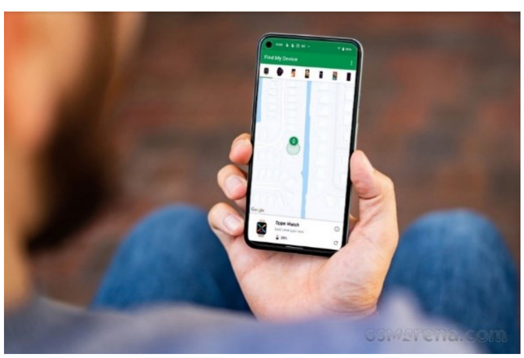


Fig 4.15 GPS Live Tracking

CHAPTER 5

APPLICATIONS, ADVANTAGES & CHALLENGES

5.1 Applications:

The applications on Robotic surveillance vehicle for monitoring.

1. Track Inspection and Maintenance: The robotic vehicle can autonomously inspect railway tracks for damage, wear, and obstructions, ensuring timely maintenance and preventing accidents.
2. Real-time Monitoring: It can provide continuous, real-time video surveillance of railway tracks and junctions, allowing operators to monitor conditions remotely, reducing the need for human patrols.
3. Incident Detection: The vehicle can use sensors to detect abnormalities such as derailments, signal malfunctions, or unauthorized access to tracks, providing early warnings.
4. Traffic Control and Junction Management: The vehicle can monitor junctions to optimize train schedules and prevent traffic congestion, ensuring safe and efficient operations.
5. Surveillance for Security: It can also act as a security system, detecting potential threats such as vandalism, trespassing, or sabotage, helping prevent crimes on railway property.
6. These applications enhance safety, efficiency, and reduce operational costs associated with manual track inspections and monitoring.

5.2 Advantages:

1. **Stair and Rail Climbing Capability**: Unlike conventional surveillance robots limited to flat surfaces, this robot can climb stairs and navigate over rails, making it ideal for multi-level buildings and uneven terrains.
2. **Wireless Control**: The robot uses Bluetooth communication, allowing users to control it remotely via a smartphone, enhancing flexibility and safety during operation.
3. **Real-Time Surveillance**: Equipped with a camera, the robot provides live video feedback, enabling users to monitor environments remotely in real-time.

4. **GPS Tracking Integration**: The inclusion of a GPS module enables real-time location tracking, which is essential for navigation, security, and deployment in large or complex environments.
5. **User-Friendly Interface**: The control system is designed for simplicity, using smartphone applications with basic commands, making it accessible even to non-experts.
6. **Enhanced Safety**: It can be deployed in hazardous or inaccessible areas, reducing the need for human presence in dangerous conditions like fires, structural collapses, or gas leaks.
7. **Long Operational Time**: Efficient power management using rechargeable batteries and voltage regulators extends operational time during field use.
8. **Open-Source Compatibility**: As the system is Arduino-based, it is compatible with open-source platforms and community-supported resources, encouraging continuous development.

5.3 Challenges:

The challenges of using robotic surveillance vehicles for monitoring railway tracks and junctions include:

1. Environmental Factors: Adverse weather conditions, such as rain, snow, or fog, can interfere with sensor performance and vehicle movement.
2. Terrain and Track Variability: Rail tracks and junctions often have varying surfaces, obstacles, and sharp curves, which can be difficult for robots to navigate smoothly.
3. Communication and Connectivity: Ensuring a stable and secure connection for transmitting data, especially in remote or rural locations with limited network infrastructure.
4. Safety and Security: Protecting the robot from potential accidents or malicious attacks while ensuring that it does not interfere with train operations.
5. Regulatory and Compliance Issues: Adhering to safety and legal standards for operating robotic vehicles in railway environments.

CHAPTER 6

RESULT & DISCUSSION

6.1 Result

1. Stair/Rail Climbing:

The stair and rail climbing mechanism is a crucial feature that distinguishes this robotic surveillance vehicle from traditional ground-based robots. It enables the robot to traverse complex terrains such as stairs, railway tracks, and inclined surfaces, making it suitable for real-world surveillance in buildings, construction zones, and railway infrastructure.

- Mechanical Design:

The robot's base chassis is designed with high ground clearance to prevent obstruction when climbing vertical steps or rails. It uses a belt-driven wheel mechanism, where the belt acts as a traction surface that maintains constant contact with the stair or rail surface. Dummy wheels are placed strategically to provide additional balance and support, preventing the robot from tipping over during ascent or descent.

- Belt System:

A continuous rubber or chain belt runs over a set of pulley-like wheels mounted at the front and rear. The belt provides grip and traction on edges of stairs or rails, allowing the robot to "crawl" upwards without slipping. This mechanism also distributes the robot's weight more evenly across the climbing surface.

- Motor Control:

The high-torque DC motors, connected via the L298N motor driver, provide sufficient power to lift the robot vertically one step at a time. The Arduino Uno controls the motor speed and direction using PWM signals, allowing precise movement control when climbing.

Fig 6.1 Climbing image of the Model

2. Accurate GPS Navigation:

The Accurate GPS Navigation system is a vital feature of the Robotic Surveillance Vehicle, enabling it to determine and report its real-time geographic location. This allows users to track the robot remotely, especially in outdoor environments such as railway tracks, disaster sites, or industrial areas.

1. GPS Module Used – NEO-6M:

The robot uses the NEO-6M GPS module, which communicates with satellites to obtain latitude, longitude, altitude, speed, and UTC time. It connects to the Arduino Uno via serial communication (TX/RX pins) and outputs location data in NMEA format.

3. Accuracy & Performance:

The NEO-6M module typically provides position accuracy of 2.5 meters under open-sky conditions. It supports 50 channels, allowing it to lock onto multiple satellites for faster and more reliable positioning.It also features a data refresh rate of up to 10Hz, ensuring frequent and accurate updates.

4. Use in the Project:

The GPS module helps operators monitor the exact location of the robot during movement. If integrated with a GSM or IoT system in the future, this data could be uploaded to the cloud for live mapping and geofencing applications. It enables mission tracking for security, surveillance, or search-and-rescue operations.

5. Real-World Benefits:

Ensures the robot doesn't get lost during autonomous or remote operations. Enhances situational awareness by allowing users to know where the robot is at all times. Increases safety and operational efficiency in large, outdoor environments.

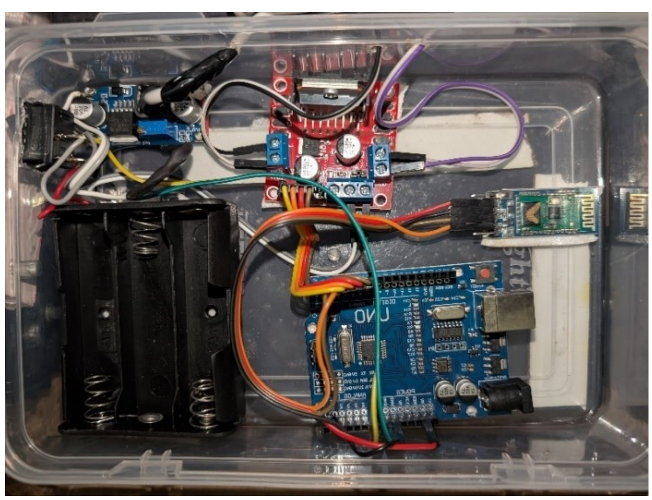


Fig 6.2 GPS Navigation Connection

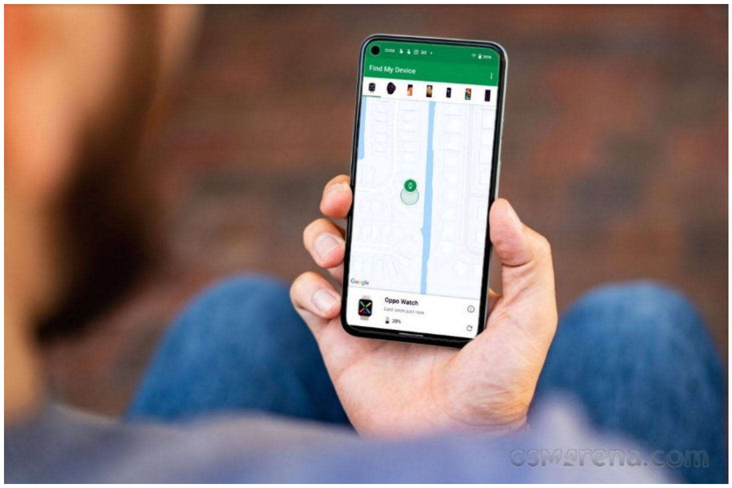


Fig 6.3 Live GPS Tracking

3. Surveillance Monitoring:

One of the core functionalities of the Robotic Surveillance Vehicle is real-time video monitoring, made possible through the integration of a CP Plus camera module. This module enables live streaming and remote visual inspection, transforming the robot into a mobile surveillance system suitable for security, rescue, and inspection tasks.

1. About CP Plus Camera Module:

CP Plus is a trusted brand in surveillance and security solutions. The camera module used in this project is a Wi-Fi or Bluetooth-enabled IP camera, designed for compact installations with high-resolution output. It is compatible with mobile apps like ezyLiv, which allow live video access on smartphones or tablets.

2. Camera Features Relevant to the Project:

High-definition (HD) video streaming ensures clear visuals for identifying objects or threats. Infrared (IR) night vision allows monitoring in low-light or dark environments. Wide-angle lens provides broader field coverage, reducing blind spots during navigation. Real-time video transmission to a mobile device via a secure Wi-Fi or Bluetooth connection. Compact and lightweight design, making it easy to mount on the robot's chassis.

3. Integration with the Robot:

The camera is mounted at the front of the vehicle to give a forward-facing live view. It is powered either by the robot's power system or an independent source. The live video feed is accessible through the CP Plus ezyLiv app, allowing the user to monitor the surroundings from a remote location. The user can also record footage, take snapshots, and adjust camera settings through the app interface.

4. Working Principle:

Once powered on, the camera creates a Wi-Fi hotspot or connects to an existing network. The mobile device pairs with the camera using the app. As the robot moves, the camera streams real-time video to the mobile screen. This live feed helps the operator navigate, detect intrusions, or observe hazards from a distance.

5. Applications and Benefits:

Enhances security monitoring in restricted or dangerous areas. Enables remote inspections in industries, tunnels, construction sites, and disaster zones. Supports search-and-rescue operations by visually locating people or objects in collapsed or inaccessible spaces. Provides visual assistance for manual control, especially during stair climbing or rail navigation.

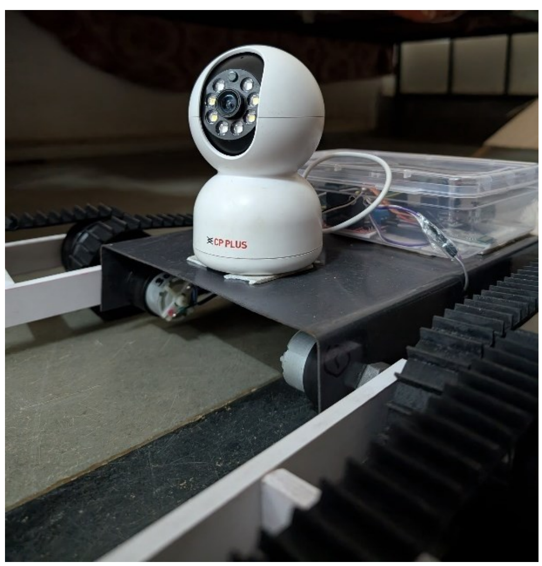


Fig 6.4 Camera connected to the Model

Fig 6.5 Final Project

6.1 Discussion

1. The robotic surveillance vehicle was successfully built using commonly available and low-cost components like Arduino Uno, HC-05, and L298N motor driver.

2. The Arduino Uno effectively handled all control operations, including Bluetooth communication, motor driving, and GPS data processing.

3. Bluetooth module (HC-05) provided smooth and reliable wireless control within a range of approximately 10 meters. Command response was immediate and accurate.

4. The L298N motor driver efficiently managed the direction and speed of the DC motors. Speed adjustments were needed to ensure balanced movement and smooth turning.

5. The GPS module accurately tracked real-time location outdoors. Signal reception was slower indoors, highlighting the typical limitations of GPS systems.

6. The camera module delivered a stable live video feed using mobile apps like ezyLiv, enabling real-time surveillance and remote visual monitoring.

7. The robot was able to move effectively on flat surfaces and handled minor obstacles. Initial testing of the rail/stair climbing mechanism showed potential but needs further improvement for complex terrains.

8. Power management was stable. The 9V battery provided sufficient operation time, although integrating solar or swappable battery systems could further enhance runtime.

9. The modular design of the system allowed easy integration of all components and offers flexibility for future upgrades such as obstacle detection or AI vision.

10. The system serves as a strong foundation for future research and development in areas like autonomous patrolling, IoT integration, and multi-robot collaboration.

CHAPTER 7

CONCLUSION & FUTURE SCOPE

The Robotic Surveillance Vehicle for Monitoring demonstrates a practical and innovative approach to enhancing security and inspection capabilities in environments that are challenging or unsafe for human presence. Through the integration of mobility, wireless communication, real-time video surveillance, and GPS tracking, the project effectively addresses the limitations of traditional surveillance methods, especially in multi-level and rail-based infrastructures. The robot’s ability to climb stairs and navigate rail tracks expands its usability across diverse applications including railway inspection, construction site monitoring, and search-and-rescue missions. The use of an Arduino Uno microcontroller, HC-05 Bluetooth module, L298N motor driver, and CP Plus camera exemplifies how low-cost, open-source components can be combined to build a robust and efficient system. The modular design and scalability of the system allow for easy enhancements, such as incorporating IoT modules for cloud connectivity or additional sensors for environmental monitoring. Moreover, the user-friendly smartphone interface ensures accessibility and ease of use, even for non-technical users.

This project not only showcases the potential of robotics and embedded systems in the surveillance domain but also serves as a foundation for future research and development in autonomous monitoring systems. It highlights the importance of automation in improving safety, efficiency, and operational effectiveness in complex and hazardous environments.

7.1 Future Scope

Future enhancements and expansions can significantly improve the robot's performance, reliability, and range of applications:

1. Autonomous Navigation with AI and Sensors
   Integration of artificial intelligence (AI) and advanced sensors (e.g., LiDAR, ultrasonic, infrared) can enable fully autonomous navigation and obstacle detection, allowing the robot to patrol predefined areas without manual control.
2. IoT and Cloud Connectivity
   Incorporating Internet of Things (IoT) modules can allow real-time data uploads to the cloud, remote diagnostics, and monitoring from any location. Cloud integration also enables long-term data storage and analysis for predictive maintenance and pattern recognition.
3. Enhanced Power Management
   The addition of solar charging units or swappable battery packs can extend operational time and reduce the need for manual recharging, making it more efficient for long-duration deployments.
4. Night Vision and Thermal Imaging
   Upgrading the camera system to include night vision or thermal imaging capabilities will allow the robot to operate effectively in complete darkness or detect heat signatures in search-and-rescue missions.
5. Robust Enclosure and Weatherproofing
   Future iterations can include rugged, waterproof casings to enable all-weather operation, expanding the use-case to outdoor and harsh environments.
6. Voice and Gesture Control
   Incorporating voice or gesture recognition can make the control system more intuitive and user-friendly, especially in constrained or fast-paced environments.
7. Integration with Government and Emergency Systems
   Connecting the system with emergency response networks or transport authorities can help trigger alerts automatically in case of accidents or intrusions.

REFERENCES

[1] Wang, Y., Song, W., Lou, Y., Huang, F., Tu, Z., & Zhang, “Simultaneous location of rail vehicles and mapping of environment with multiple LiDAR’s” IEEE Transactions on Industrial Electronics, vol. 67, no. 5, (2021)

[2] Guillen, S. Cranfield University and Network Rail develop “Autonomous rail inspection and repair system. Clear path Robotics”, Journal of Robotic Systems, 14(4), 231–249, (2021)

[3] Shenhao Technology. “Railway inspection robot takes on maintenance challenges”. IEEE Internet of Things Journal, vol. 9, no. 3, (2022, October 28)

[4] Prof. Najma Siddiqui, Prof. Akil Ahmed " Development of Railway Track Crack Detection System using Arduino" Railway Track Crack Detection System using Arduino, (2022)

[5] Atack, P. R. Japanese railway introduces “Infrastructure robot”. Railway Technology, Springer handbook of robotics (2nd ed.) Springer, (2024, July 4)

APPENDIX

Program

```
#include <SoftwareSerial.h>

int LmotorA=4;
int LmotorB=5;
int RmotorA=6;
int RmotorB=7;
int M3=8;
int M4=9;

int M5=10;
int M6=11;

SoftwareSerial Ser(2,3);

void setup()
{
  pinMode(LmotorA,OUTPUT);
  pinMode(LmotorB,OUTPUT);
  pinMode(RmotorA,OUTPUT);
  pinMode(RmotorB,OUTPUT);
  pinMode(M3,OUTPUT);
  pinMode(M4,OUTPUT);
  pinMode(M5,OUTPUT);
  pinMode(M6,OUTPUT);

  Ser.begin(9600);
  Serial.begin(9600);
```

```
}

void loop()
{
  if(Ser.available())
  {
    char data=Ser.read();
    Serial.print(data);
    if(data=='F')
    {
      Forward();
    }
    else if(data=='B')
    {
      Backward();
    }
    else if(data=='L')
    {
      Left();
    }
    else if(data=='R')
    {
      Right();
    }
    else if(data=='W')
    {
      Clockwise();
    }
    else if(data=='U')
    {
      CounterClockwise();
```

```
  }
  else if((data=='w')||(data=='u'))
  {
   MotorStop();
  }

  else if(data=='X')
  {
   Clockwisewise();
  }
  else if(data=='V')
  {
   CounterClockwisewise();
  }
  else if((data=='x')||(data=='v'))
  {
   MotorStopwise();
  }


  else
  {
   Stop();
  }
 }
}

void Forward()
{
 digitalWrite(LmotorA,HIGH);
 digitalWrite(LmotorB,LOW);
```

```
  digitalWrite(RmotorA,HIGH);
  digitalWrite(RmotorB,LOW);
  Serial.println("F");
}
void Backward()
{
  digitalWrite(LmotorA,LOW);
  digitalWrite(LmotorB,HIGH);
  digitalWrite(RmotorA,LOW);
  digitalWrite(RmotorB,HIGH);
  Serial.println("B");
}
void Right()
{
  digitalWrite(LmotorA,HIGH);
  digitalWrite(LmotorB,LOW);
  digitalWrite(RmotorA,LOW);
  digitalWrite(RmotorB,HIGH);
  Serial.println("R");
}
void Left()
{
  digitalWrite(LmotorA,LOW);
  digitalWrite(LmotorB,HIGH);
  digitalWrite(RmotorA,HIGH);
  digitalWrite(RmotorB,LOW);
  Serial.println("L");
}
void Clockwise()
{
  digitalWrite(M3,LOW);
```

```
  digitalWrite(M4,HIGH);
  Serial.println("CW");
}
void CounterClockwise()
{
  digitalWrite(M3,HIGH);
  digitalWrite(M4,LOW);
  Serial.println("CCW");
}

void MotorStop()
{
  digitalWrite(M3,LOW);
  digitalWrite(M4,LOW);
  Serial.println("Mstop");
}

void Clockwisewise()
{
  digitalWrite(M5,LOW);
  digitalWrite(M6,HIGH);
  Serial.println("CWise");
}
void CounterClockwisewise()
{
  digitalWrite(M5,HIGH);
  digitalWrite(M6,LOW);
  Serial.println("CCWise");
}
```

```
void MotorStopwise()
{
  digitalWrite(M5,LOW);
  digitalWrite(M6,LOW);
  Serial.println("Mstopwise");
}



void Stop()
{
  digitalWrite(LmotorA,LOW);
  digitalWrite(LmotorB,LOW);
  digitalWrite(RmotorA,LOW);
  digitalWrite(RmotorB,LOW);
  Serial.println("S");
}
```