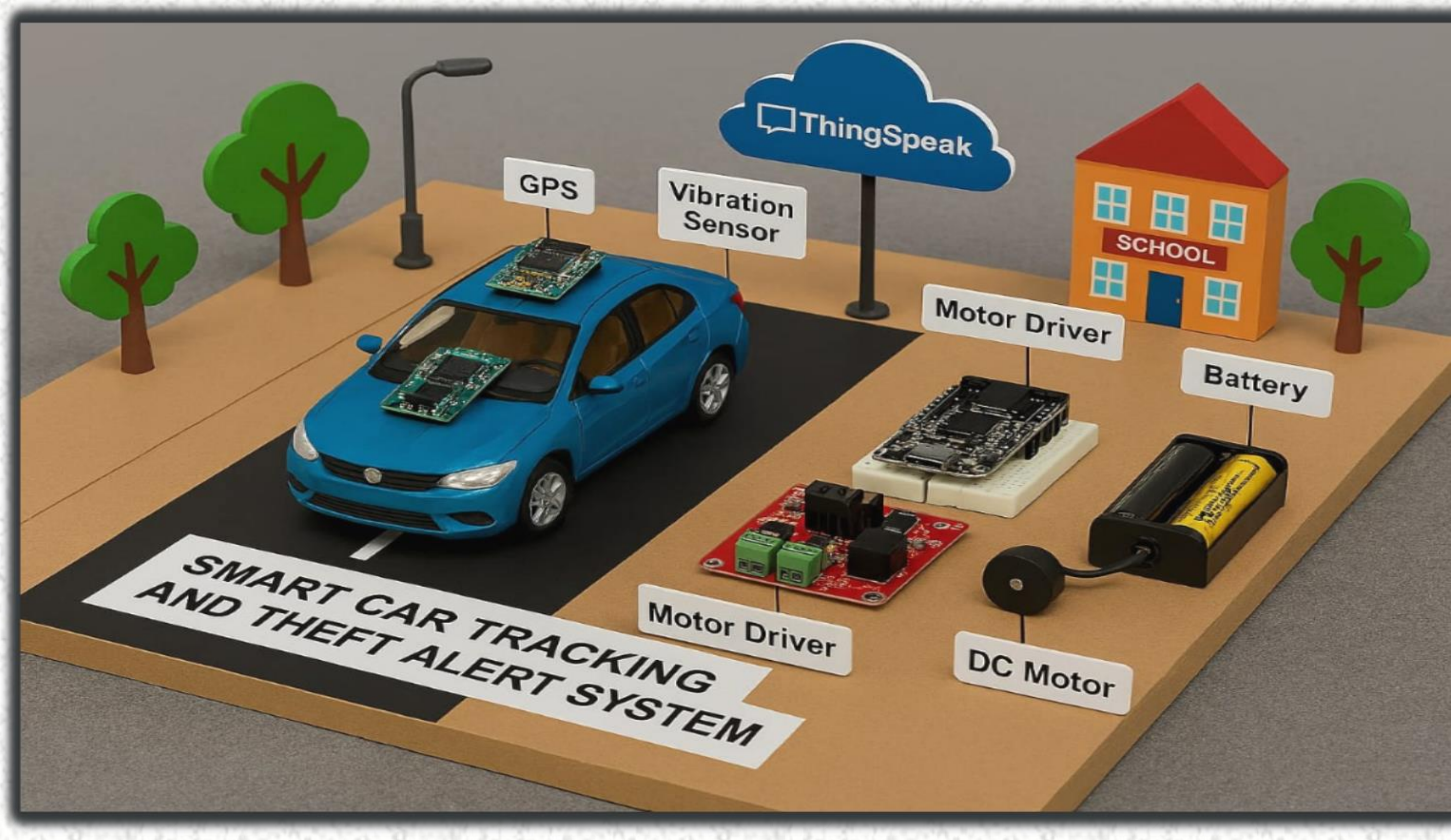
SMART CAR TRACKING AND THEFT ALERT SYSTEM

Embedded and IoT System



23-NTU-CS-1082

23-NTU-CS-1086

Abstract

This project presents a **Smart Car Tracking and Theft Alert System** based on Internet of Things (IoT) technology. The system is designed to enhance vehicle security by integrating **real-time GPS tracking**, **vibration-based theft detection**, and a **geo-fencing alert mechanism**. An **ESP32 microcontroller** acts as the central control unit, while the **Blynk mobile application** provides remote monitoring, control, and notification services. The proposed system allows users to remotely control the vehicle engine, monitor movement, receive instant alerts, and track the vehicle location using cloud platforms.

1. Introduction

Vehicle theft and unauthorized usage are increasing due to the lack of real-time monitoring in conventional security systems. Traditional alarm systems do not provide remote access or live location tracking.

The **Smart Car Tracking and Theft Alert System** overcomes these limitations by using IoT technologies to provide **remote vehicle control**, **live tracking**, and **instant alert notifications** through a smartphone application.

2. Objectives of the Project

The main objectives of this project are:

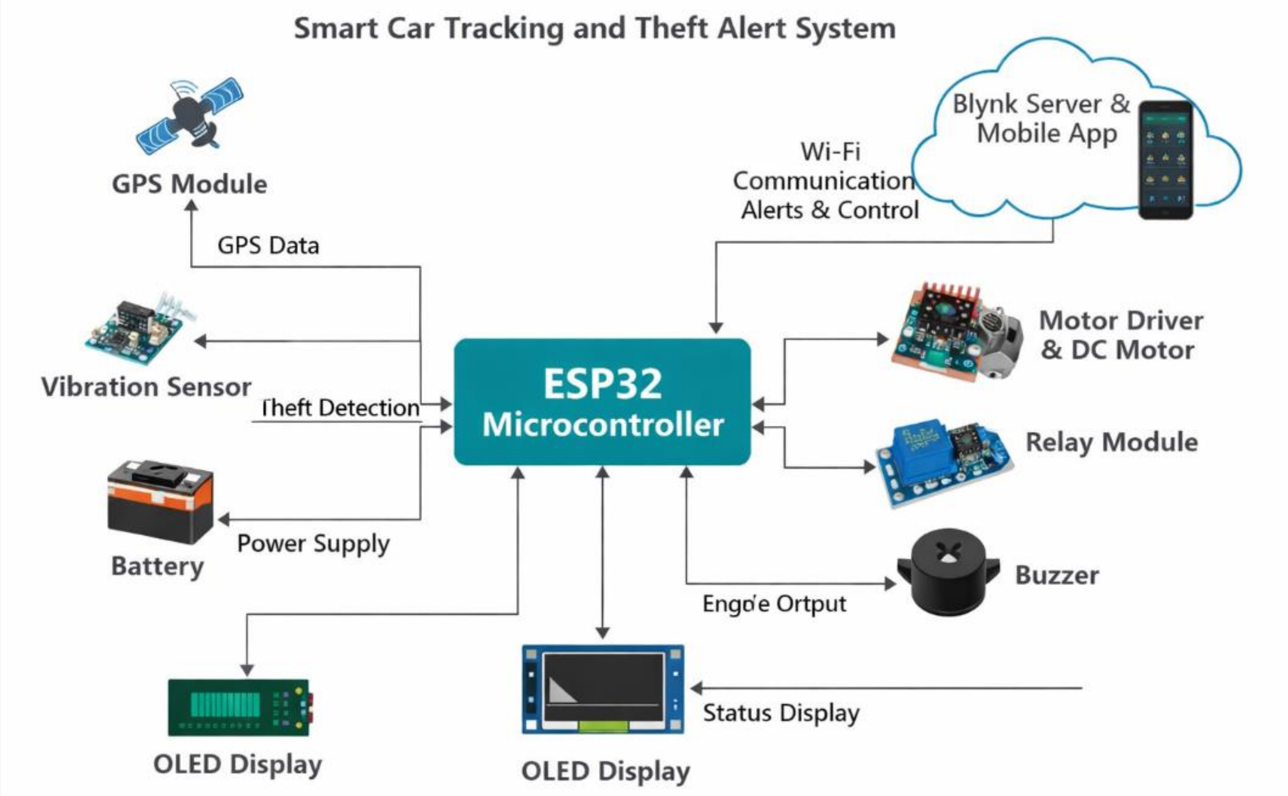
- To track the **real-time location** of a vehicle using GPS
- To detect **theft attempts** using a vibration sensor
- To implement a **geo-fence-based** security system
- To allow remote **engine locking** and **unlocking**
- To provide instant **alerts** via a mobile application
- To enhance **vehicle safety** using an audible alarm system

3. System Overview

The system is built around the **ESP32 microcontroller**, which integrates all sensors, actuators, and communication modules.

Using **WiFi connectivity**, the ESP32 communicates with the **Blynk cloud server** and **ThingSpeak platform**.

3.1 Block Diagram of Smart Car Tracking and Theft Alert System



System Workflow:

1. ESP32 connects to WiFi and Blynk
2. GPS module provides live location data
3. System continuously monitors vibration and location
4. Alerts are triggered only in **ARM mode**
5. User controls the system remotely using Blynk

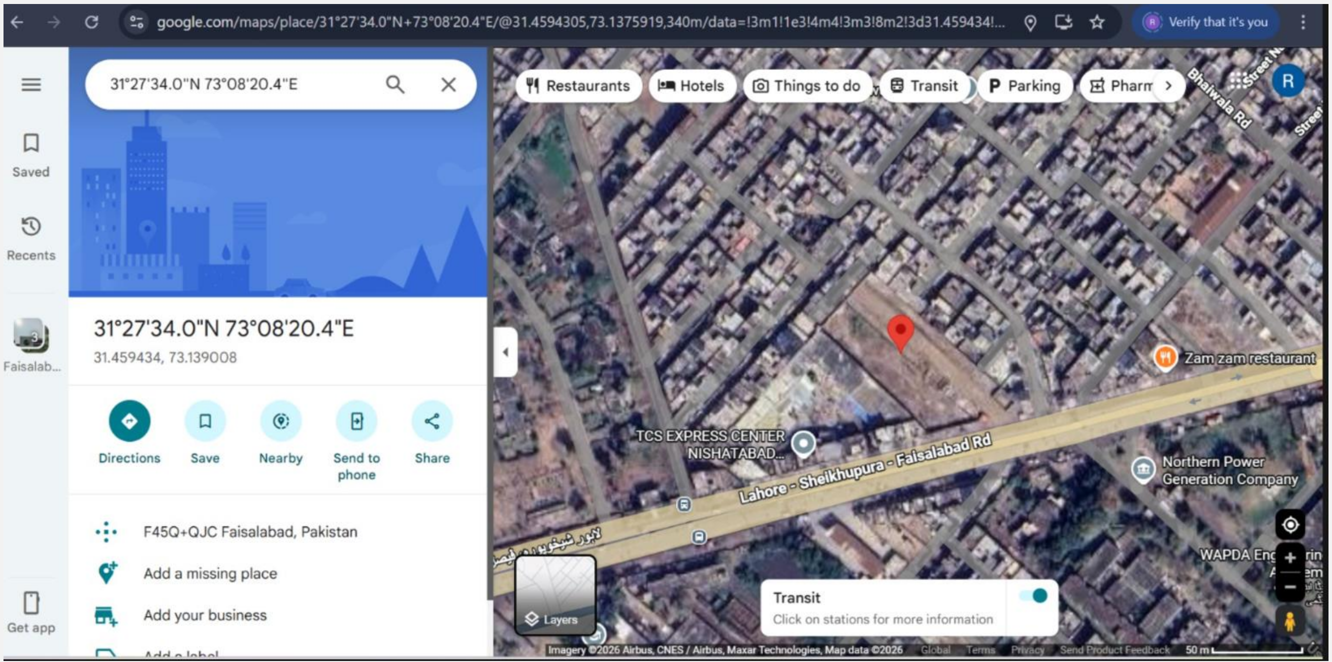
4. Key Features of the System

4.1 GPS-Based Vehicle Tracking

The GPS module continuously collects latitude and longitude data of the vehicle.

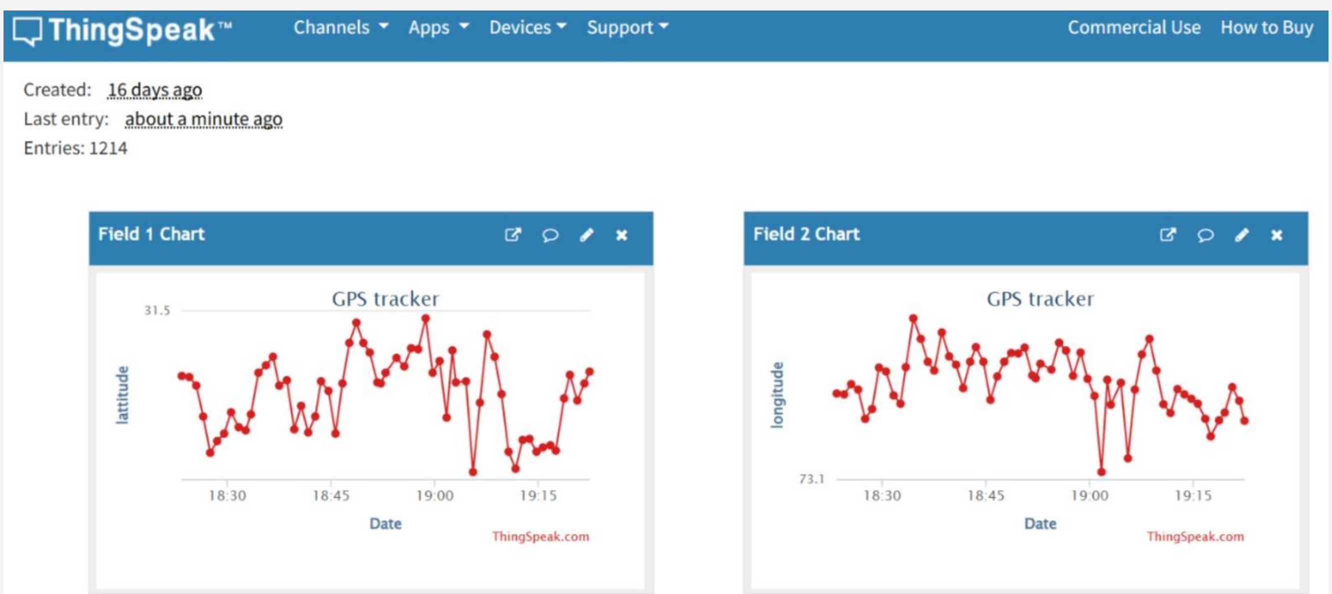
- Live GPS data is processed using the TinyGPS++ library
- Location is displayed on the **OLED** screen
- Last known location is stored if GPS signal is lost
- GPS data is uploaded to **ThingSpeak cloud**

The real-time location of the vehicle displayed on Google Maps using GPS latitude and longitude data transmitted via the IoT system.



4.2 GPS Latitude and Longitude Data on ThingSpeak Cloud

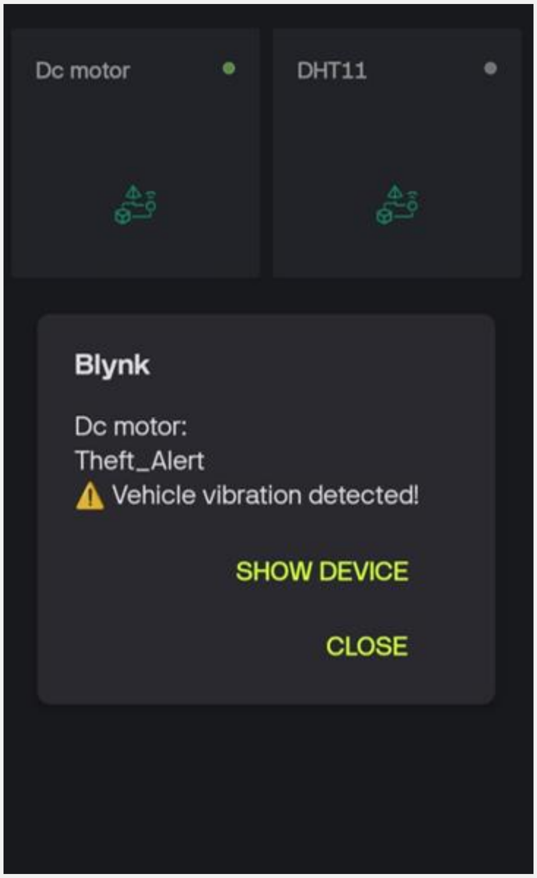
The latitude and longitude values plotted over time on the ThingSpeak cloud platform, demonstrating continuous GPS data logging.



4.3 Theft Detection Using Vibration Sensor

A vibration sensor is used to detect unauthorized movement such as lifting or forced vehicle motion.

- Works only when the system is **ARMED**
- Triggers buzzer alarm
- Sends instant notification to Blynk
- Prevents repeated alerts until vibration stops

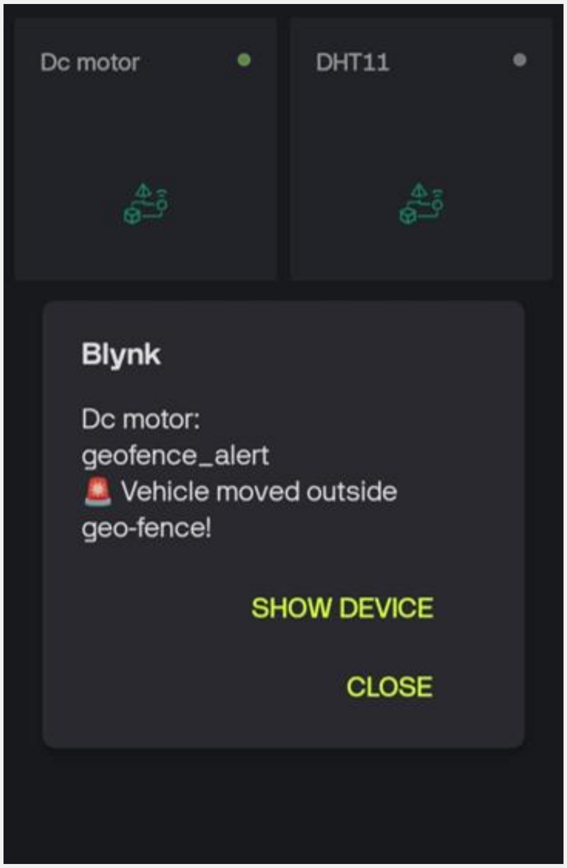


This notification shows up in the Blynk app when the vibration sensor feels any movement while the system is in ARM mode. It warns the user quickly about possible theft or touching of the vehicle. At the same time, the buzzer turns on to give a loud sound alert.

4.4 Geo-Fencing Alert Mechanism

A geo-fence is defined by setting a **center latitude**, **center longitude**, and **radius**.

- Distance is calculated using the Haversine formula
- If the vehicle moves outside the defined area:
    - Geo-fence alert is triggered
    - Buzzer is activated
    - Notification is sent to the user
- Alerts stop automatically when the vehicle re-enters the area



This alert is shown when the vehicle goes outside the geo-fence limit. The system sends a notification to the user and turns on the buzzer to warn about the movement.

4.5 Remote Engine Locking System

The engine locking mechanism is implemented using a **relay module**.

- Controlled via Blynk mobile application
- When locked:
    - Motor operation is disabled
    - Vehicle movement is stopped
- Enhances theft prevention by immobilizing the vehicle

4.6 Motor Control System

A motor driver controls the DC motor to simulate vehicle movement.

- Forward and backward control using Blynk buttons
- Motor operates only when engine is unlocked
- Ensures safe and controlled movement

4.7 ARM / DISARM Security Mode

The ARM/DISARM mode prevents false alerts.

- **ARMED mode**:
    - Vibration detection active
    - Geo-fence monitoring active
- **DISARMED mode**:
    - All alerts disabled
    - Buzzer turned off
- Controlled via Blynk or Serial Monitor

5. Hardware Components

Component	Description
ESP32	Main controller with WiFi and Bluetooth for data processing.
GPS Module	Provides real-time vehicle location tracking via satellites.
Vibration Sensor	Detects physical tampering or impact for theft detection.
Motor Driver	Controls the speed and direction of the DC Motor.
DC Motor	Used for movement simulation or automated mechanisms.
Relay Module	Acts as an electronic switch to lock/unlock the engine (Ignition Cut).
Buzzer	Provides an audible alarm during unauthorized access.
OLED Display	Shows system status, coordinates, and alert messages.
Battery	Portable power supply for the entire circuit.
Breadboard	Used for circuit prototyping without soldering.
Jumper Wires	Connects all components to the ESP32 and power source.

6. Software Tools and Platforms

- Arduino IDE
- Blynk IoT Platform
- ThingSpeak Cloud
- TinyGPS++ Library

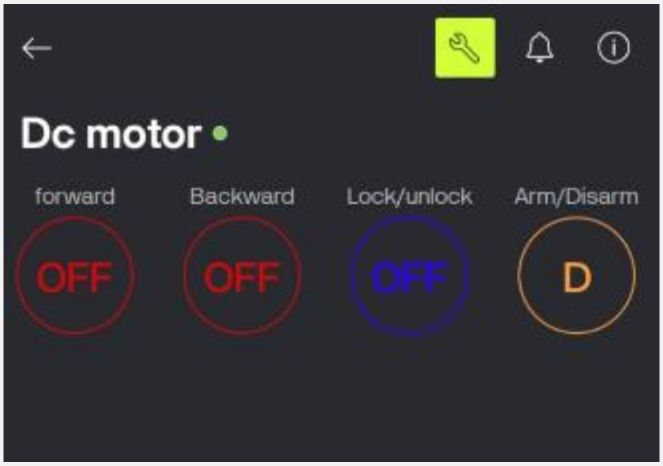
- Adafruit OLED Libraries

7. Blynk Mobile Application Configuration

Virtual Pin	Function
V1	Motor Forward
V2	Motor Backward
V3	Engine Lock / Unlock
V4	ARM / DISARM System

7.1 Blynk Mobile Application Interface for Motor and Engine Control

The Blynk mobile application interface is used to control the DC motor movement, engine lock, and ARM/DISARM security modes remotely.



8. Pin Configuration and Hardware Interface

This section explains how each hardware component is connected to the ESP32.

8.1 ESP32 Pin Assignment Table

Component	ESP32 Pin	Pin Type	Function
Motor Driver IN1	GPIO 4	Output	Motor forward control
Motor Driver IN2	GPIO 18	Output	Motor backward control
Relay Module	GPIO 33	Output	Engine lock/unlock
Vibration Sensor	GPIO 27	Input	Theft detection

Buzzer	GPIO 26	Output	Alarm activation
GPS RX	GPIO 16	UART RX	Receives GPS data
GPS TX	GPIO 17	UART TX	Transmits GPS data
OLED SDA	SDA	I2C	Display data
OLED SCL	SCL	I2C	Display clock
Power Supply	VIN / 5V	Power	Battery input
Ground	GND	Ground	Common ground

8.2 Pin Diagram Explanation

- **GPIO 4 and GPIO 18** control the motor direction
- **GPIO 33** controls the relay for engine locking
- **GPIO 27** monitors vibration signals
- **GPIO 26** activates the buzzer during alerts
- GPS module communicates via **UART pins**
- **OLED display** uses I2C protocol
- All components share a common ground

8.3 Logical Pin Diagram

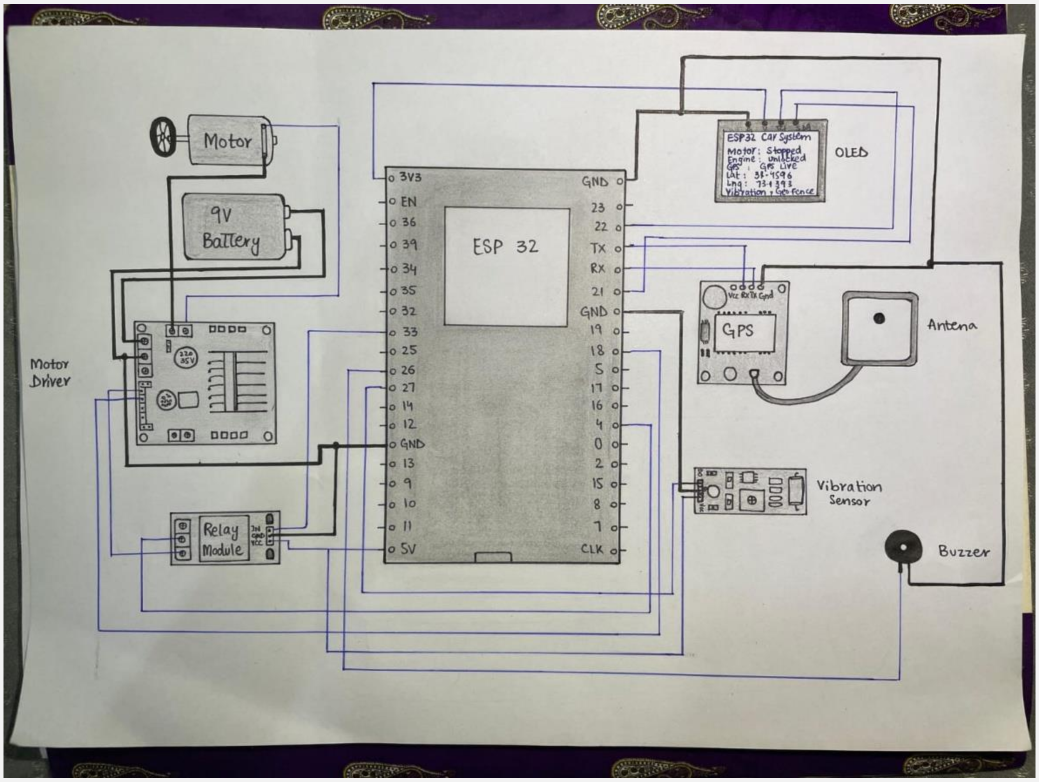
ESP32 Connections:

ESP32 Pin	Component	Pin Function
GPIO 4	Motor Driver	IN1 (Direction Control)
GPIO 18	Motor Driver	IN2 (Direction Control)
GPIO 33	Relay Module	Signal Pin (Engine Lock/Unlock)
GPIO 27	Vibration Sensor	Digital Input (Theft Detection)
GPIO 26	Buzzer	Signal Pin (Alarm Output)
GPIO 16	GPS Module	TX (Data Transmission)
GPIO 17	GPS Module	RX (Data Reception)
GPIO 21 (SDA)	OLED Display	I2C Data Communication
GPIO 22 (SCL)	OLED Display	I2C Clock Communication

VIN	Battery	Power Input (5V)
GND	Battery	Common Ground

8.4 Circuit Layout

This diagram illustrates the full wiring layout of the system. It shows how each component communicates with the ESP32 using GPS, OLED, Motor Driver, Relay Module, Vibration Sensor, and Buzzer.



9. Hardware implementation

This diagram shows the hardware prototype. ESP32 is the main controller connecting GPS, vibration sensor, relay, motor driver, buzzer, and OLED. GPS track's location, vibration sensor detects theft, relay locks the engine, motor moves the vehicle, and buzzer gives alerts. All components are powered by a 5V battery on a breadboard.

10. System Operation (Step-by-Step)

- ESP32 powers on and connects to WiFi
- Blynk server connection is established
- GPS module starts location tracking
- System waits in DISARM mode
- User arms the system via Blynk
- Vibration or geo-fence breach triggers alerts
- Buzzer and notifications are activated
- User can lock engine or stop motor remotely

11. Alerts and Notifications

- **Vibration Alert**: Unauthorized movement detected
- **Geo-Fence Alert**: Vehicle moved outside defined area
- Alerts are sent instantly to the user via the Blynk mobile application.

12. Advantages of the System

- Real-time tracking
- Remote vehicle control
- Strong theft prevention
- Low-cost IoT solution
- Easy mobile interface
- Scalable design

13. Limitations

- Internet connection required
- GPS accuracy depends on signal strength
- Prototype implementation (not a real vehicle ECU)

14. Future Enhancements

- SMS alert integration
- Live map tracking
- Camera-based monitoring
- Speed and fuel tracking
- Full vehicle ECU integration

15. Conclusion

The **Smart Car Tracking and Theft Alert System** provides a reliable and intelligent vehicle security solution using IoT technologies. By combining GPS tracking, theft detection, geo-fencing, and remote engine control, the system enhances vehicle safety and user convenience. The project demonstrates the effective use of ESP32, sensors, and cloud platforms in real-world security applications.

16. GitHub link:

https://github.com/Rameen-Fatima67/Embedded-iot--23-NTU-CS-1086-Rameen