Design and implementation of Hand Gesture Controlled Car with Remote Surveillance and Navigation System

Abinash Bhujel*, Abinash Dewan, Arun Rai, Bishal Sharma, Preya Gurung
Department of Electronics and Communication Engineering, CCCT Chisopani Sikkim 737121

***Abstract—*This project presents the design and development of a hand gesture-controlled car equipped with remote surveillance and navigation capabilities. The system is divided into a transmitter and a receiver unit. In the transmitter, an ESP8266 module reads real-time motion data from an MPU6050 sensor, which captures hand gestures in terms of pitch and roll. This data is wirelessly transmitted using the NRF24L01 module. The receiver section is powered by two Arduino boards: an Uno controls four DC motors via a motor driver for movement based on received gestures, while a Nano handles GPS and GSM modules for navigation and location tracking. For surveillance, an ESP32-CAM module streams live video. The entire receiver setup, including navigation and surveillance components, operates on a single 12V power supply. The system ensures movement accuracy using Mecanum wheels and features intelligent communication loss handling with buzzer alerts. The developed prototype was tested under various real-world conditions, demonstrating precise control, efficient surveillance, and reliable tracking. This hands-free interface can serve critical roles in rescue operations, military applications, and remote patrolling. Its modular and scalable design provides a strong foundation for future enhancements, such as AI-based obstacle detection, autonomous navigation, or integration with IoT networks.**

***Keywords—*Hand gesture, Arduino uno, ESP8266, Robotic car, Navigation, GPS and GSM**

I. INTRODUCTION

The evolution of embedded systems and wireless communication has significantly contributed to the development of intelligent robotic platforms. Conventionally, robotic vehicles have relied on physical controllers or smartphone-based applications to manage navigation, which often introduces latency or demands constant visual attention. As a result, gesture-based control mechanisms are gaining traction as a more seamless and intuitive alternative for human-machine interaction.

This work proposes a gesture-controlled robotic car that combines navigation, surveillance, and communication into one functional system. Unlike prior models that separate control and tracking functionalities or depend solely on manual input, this system integrates multiple modules for real-time performance and versatility. The gesture interface is implemented using an MPU6050 sensor paired with an ESP8266 microcontroller on the transmitting side, translating hand movements into directional commands. At the receiving end, an Arduino Uno governs motor control and communication with the NRF24L01 module, while an Arduino Nano handles the GSM and GPS components for remote location tracking. Furthermore, an ESP32-CAM module is utilized to provide live video streaming, enhancing the surveillance capability of the system.

This integrated approach sets the system apart from traditional robotic designs by enabling hands-free control, real-time feedback, and location-aware operation. The modular nature of the hardware also allows for easy adaptation to various applications, including search and rescue missions, hazardous area inspection, and unmanned patrol systems. In doing so, the project addresses key limitations such as restricted control range, delayed response, and lack of environmental awareness found in earlier systems.

II. LITERATURE REVIEW

In recent years, gesture-based robotic systems and wireless control have gained significant traction, particularly for applications involving navigation and surveillance. Various studies have contributed to this domain, each focusing on specific functionalities.
Rathod and Jaiswal [1] developed a gesture-controlled robot using the MPU6050 sensor interfaced with an Arduino Uno. Their design offered basic movement control but lacked any surveillance or tracking capability, thus limiting practical applications.

Dhumal and Dhone [2] extended the idea by integrating both the MPU6050 and flex sensors to capture complex hand gestures. With NRF24L01 for wireless transmission, their system demonstrated improved responsiveness. However, real-time tracking and visual monitoring were absent.

Parate and Bhosale [3] proposed a similar wireless robotic vehicle utilizing an accelerometer-based gesture control approach. While their method enabled intuitive movement, it still did not address the need for GPS or GSM modules for location tracking or message alerts.

In a separate line of work, Sharma and Agarwal [4] presented a GPS and GSM-based vehicle tracking system focused on transportation monitoring. Though efficient in real-time location updates via SMS, it lacked mobility and control capabilities, which are essential in dynamic robotic applications.

Gupta and Jaiswal [5] explored surveillance through the ESP32-CAM module, enabling wireless video streaming over a local network. Their implementation offered effective monitoring but remained stationary, missing out on mobile robotic integration.

Patil and Kale [6] designed a vehicle tracking model using IoT concepts with GPS and GSM support. Their primary objective was to enhance vehicular security, but gesture-based control and real-time surveillance were not considered.

Shaikh and Pathan [7] demonstrated an IoT-based gesture-controlled robot using the ESP8266 and MPU6050 modules. Although their robot offered wireless control, it didn't include GPS or video streaming capabilities, leaving room for a more integrated solution.

Collectively, these contributions emphasize the growing interest in gesture, tracking, and surveillance technologies, but they fall short of combining all three features into a cohesive system. The proposed project distinguishes itself by integrating gesture-based control via MPU6050, real-time location tracking using GPS and GSM, and live surveillance through ESP32-CAM, forming a complete and versatile robotic platform.

Summary table

Sl. no	Title of the paper /author	Technology used	Key finding	Gap
1	Gesture controlled robot-Rathod &Jaiswal	Arduino uno, MPU6050	Implemented Hand gesture Robotic control	No GPS, GSM, or camera
2	Wireless Gesture Robot-Dhumal &Dhone	Arduino, MPU6050, Flex sensor, NRF240L1	Wireless robotic control using hand and flex sensor input	Lacks Tracking and real-time monitoring
3	Accelerometer based robot-Parate &Bhosale	Accelerometer, Rf module	Enabled movement via accelerometer	No GPS or Surveillance capability
4	Vehicle tracking system-Sharma & Agrawal	GPS, GSM, Arduino	Tracked real - time location via SMS	No robotic control or live monitoring
5	ESP32-CAM based Surveillance-Gupta & Jaiswal	ESP32-CAM	Provide live video streaming over Wi-Fi	No mobility or gesture - based control
6	IOT vehicle tracking-Patil& Kale	GPS, GSM, IOT	Vehicle location monitoring for safety	Does not include gesture or camera integration
7	Gesture controlled IOT robot-Shaikh& Pathan	ESP8266, MPU600	IOT-enabled hand gesture robot	Lacks GPS and Real-time Video feed

III. SYSTEMMODEL/METHODOLOGY

And Design

3.1 system model overview

In this project, we create a robotic car that one can control with hand gestures from a distance. The system can be broken down into two primary sections:

Controller (Transmitter Section): Includes ESP8266 microcontroller, gyroscope sensor (MPU6050) and NRF24L01 transmitter module.

Robotic Car (Receiver Section): Contains Arduino UNO, NRF24L01 receiver, motor driver module (L298N), motors, GSM module, GPS module, wireless camera, power supply and other components.

The system uses hand gesture recognition via a gyroscope sensor, which is relayed wirelessly through the NRF24L01. For real-time surveillance, video streaming is done through the camera, while communication and location tracking are enabled through the GSM and GPS modules.

3.2 Methodology

The system functions as follows:

1. Gesture Recognition:

Placement of the user's hand is detected by the MPU6050 gyroscope sensor.

When a user tilts their hand forward, the car is programmed to move in the same direction.

Car moves in reverse when users tilt their hand backwards.

A leftward tilt will instruct the car to make a left U-turn.

A rightward tilt will prompt the car to make a right U-turn.

2. Signal Processing:

The command is sent through an NRF24L01 wireless transmitter after the gesture command is identified, and the hand movement is captured by the ESP8266 microcontroller.

3. Communication Via Wireless:

The commands are received by the car's NRF24L01 receiver.

4. Motion Control:

The car moves in accordance with the signals that are processed by the Arduino UNO and sent to the motors via the L298N motor driver.

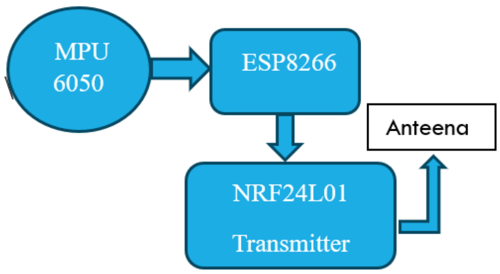
5. Real-time monitoring: The car's wireless camera transmits live video to the user's computer or smartphone.

6. Tracking and Messaging Locations:

The GPS module continuously monitors the vehicle's current location. On request or in an emergency, the GSM module can text the user the location of the vehicle.

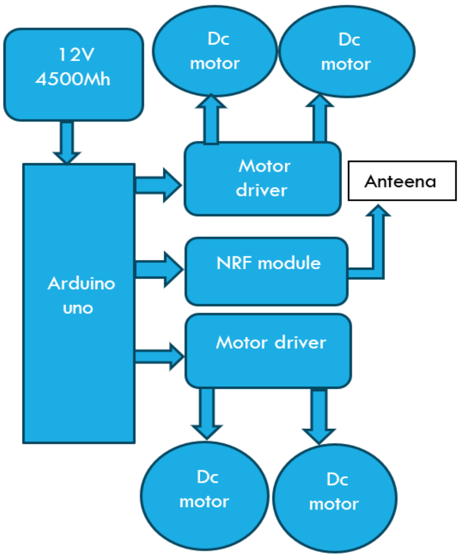
Block diagram:

Fig.1 Transmitter Block



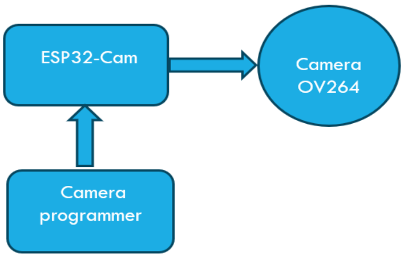
The transmitting part of the system is worn by the user and mainly handles gesture detection. A gyroscope sensor (MPU6050) reads the orientation of the hand—such as tilting forward, backward, or to the sides. This data is then sent to the ESP8266 microcontroller, which processes the gesture into a signal. Once processed, the command is wirelessly transmitted using the NRF24L01 module to the receiving unit on the robotic vehicle.

Fig.2 Receiver Block



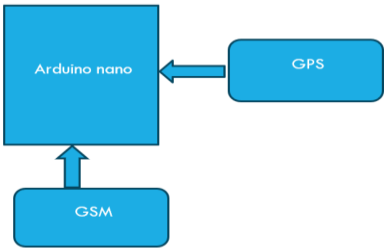
The receiving part is placed on the vehicle. It begins with the NRF24L01 module, which captures the wireless signals sent from the transmitter. These signals are passed to the Arduino UNO, which interprets the gesture commands. Based on the input, the Arduino controls the motor driver module (L298N) to move the vehicle forward, backward, or turn left or right by powering the DC motors accordingly.

Fig.3Surveillance Block



For the surveillance feature, the system uses an ESP32-CAM module, which has a small built-in camera called the OV2640. This setup allows the car to stream live video over Wi-Fi. Once connected to a network, the ESP32-CAM creates a video link that the user can open on a mobile phone or computer browser. This way, the user can see what's around the vehicle in real time, without needing extra screens or equipment. It's a simple and low-cost way to add live monitoring to the system.

Fig.4 Navigation Block



The navigation part of the system helps track the vehicle's location. Since the main Arduino UNO was running out of pins, an Arduino Nano is used specifically for this block. It handles both the GPS module, which gets the current location, and the GSM module, which sends that location to the user via SMS when needed. The Nano reads data from the GPS, processes it, and uses the GSM to send the coordinates to a mobile number. This setup keeps the navigation system separate and ensures smooth communication without pin conflicts.

CIRCUIT DIAGRAM

Fig 5.1 Transmitter

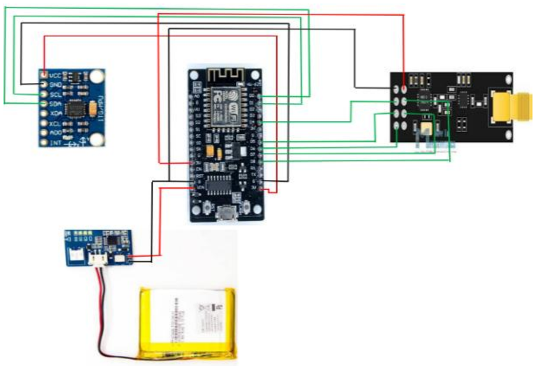
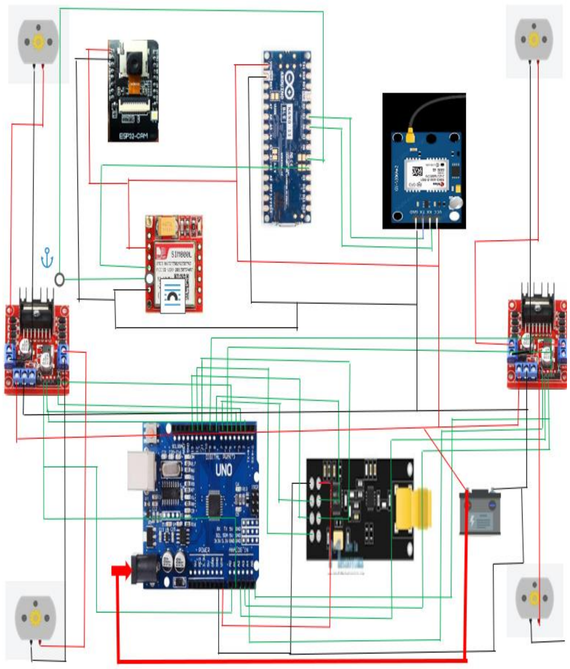


Fig 5.2 Receiver



Circuit Description

This project is divided into four key sections: the transmitting unit, receiving unit, surveillance system, and navigation system. Each section plays a vital role in ensuring the smooth functioning of a wireless gesture-controlled car capable of live video streaming and GPS tracking.

1. Transmitting Unit (Hand Gesture Control using ESP8266 & MPU6050)

The transmitting section uses an ESP8266 NodeMCU board as the central controller. Instead of traditional joystick or smartphone control, an MPU6050 sensor is used to capture hand movements (tilt in pitch and roll axes). These readings are interpreted and transmitted wirelessly via an NRF24L01 module.

The MPU6050 communicates with the ESP8266 using the I2C protocol (SDA and SCL pins).

The NRF24L01 is connected via SPI and transmits the orientation data to the receiving unit.

Power is supplied using a 9V battery stepped down to 3.3V suitable for the ESP8266 and NRF module.

This setup forms a compact, gesture-based wireless remote controller.

2. Receiving Unit (Motor Control using Arduino UNO & NRF24L01)

On the receiver side, an Arduino UNO is used to process incoming signals from the transmitter via an NRF24L01 receiver module. Based on the pitch and roll values received, the Arduino controls four DC motors connected in a Mecanum wheel configuration through L298N motor drivers.

The motor drivers receive commands from the Arduino to control direction and speed.

Each motor driver is powered by a 12V rechargeable battery and is capable of bi-directional control for smooth movement (forward, backward, left, right, and diagonal).

The entire receiver circuit shares a common ground to ensure proper communication and power distribution.

3. Surveillance System (Live Video using ESP32-CAM)

For real-time visual monitoring, an ESP32-CAM module with an integrated OV2640 camera is added to the system. This module connects to a Wi-Fi network and streams video to a mobile device or browser.

The ESP32-CAM is powered using a 5V supply, stepped down from the main 12V battery through a buck converter.

It operates independently and does not interfere with motor control, ensuring a reliable live feed during car operation.

4. Navigation System (GPS & GSM with Arduino Nano) Navigation and location tracking are handled by a separate Arduino Nano, which interfaces with a Neo-6M GPS module and a SIM800L GSM module.

The GPS module provides real-time coordinates of the car.

These coordinates are sent via SMS using the GSM module to a preconfigured phone number.

Both modules are connected via serial communication, and the Arduino Nano handles transmission.

Power is supplied through the same 12V battery, regulated appropriately to 5V for stable operation.

Power Management

A single 12V rechargeable battery powers the entire receiving setup:

It directly powers the motor drivers and the Arduino UNO via Vin.

A buck converter steps down the voltage to 5V for the ESP32-CAM, Arduino Nano, GPS, and GSM modules.

Proper grounding is ensured across all components to prevent communication issues.

Overall Function

The user wears the transmitter (ESP8266 + MPU6050), which detects hand gestures and sends data wirelessly. The receiver interprets this data to control the car's movement. Meanwhile, the ESP32-CAM streams live video, and the Arduino Nano handles location tracking. All components work in sync to deliver an intelligent, gesture-controlled robotic system with advanced features like navigation and surveillance.

4. Experimental conduction

The experimental phase of this project was structured into two key components: the transmitter and the receiver sections, each developed and tested to ensure effective communication, mobility, and surveillance.

A. Transmitter Section

The transmitter was built using the ESP8266 NodeMCU microcontroller and the MPU6050 accelerometer-gyroscope sensor. The MPU6050 was used to detect hand gestures, translating the orientation of the user's hand into directional control commands. These commands were processed by the ESP8266 and transmitted wirelessly through the NRF24L01 module. The device was powered via a 5V USB source, and calibration of gestures was conducted during testing to ensure accurate and consistent responses.

B. Receiver Section

The receiver was segmented into three subsystems: motion control, navigation, and surveillance.

For motion control, an Arduino Uno processed data from the NRF24L01 and operated four DC motors arranged in a mecanum wheel setup via the L298N motor driver, enabling omnidirectional movement.

The navigation unit, managed by an Arduino Nano, used a Neo-6M GPS module and a SIM800L GSM module to send real-time location coordinates through SMS.

The surveillance system employed an ESP32-CAM module for live video streaming over Wi-Fi. The module was programmed to connect with a mobile browser, providing real-time visual feedback.

C. Power Supply Configuration

A single 12V rechargeable battery powered all components in the receiver unit. Step-down voltage regulators were integrated to safely supply 5V or 3.3V to the Arduino boards, GPS, GSM, and ESP32-CAM modules.

D. Testing Procedure

The entire system was tested in an open environment. Gesture commands were mapped and verified for real-time vehicle response, GPS data was validated using SMS location tracking, and the ESP32-CAM stream was monitored using a smartphone browser. Results confirmed the system's ability to respond accurately and function reliably under various conditions.

Hardware Components Used:

Transmitter Section:

ESP8266 NodeMCU: Main controller that processes gesture data.

MPU6050 Sensor: Detects hand movements using acceleration and rotation.

NRF24L01 Module: Sends gesture signals wirelessly to the receiver.

Battery Pack: Powers the transmitter system.

Receiver Section:

Arduino Uno: Controls the motor driver and car movement.

Arduino Nano: Handles GPS and GSM module operations.

L298N Motor Driver Module: Drives the four DC motors.

4 × DC Motors with Mecanum Wheels: Enable movement in all directions.

NRF24L01 Module: Receives gesture signals from the transmitter.

ESP32-CAM Module: Streams live video over Wi-Fi.

NEO-6M GPS Module: Tracks the car's location.

SIM800L GSM Module: Sends the GPS coordinates via SMS.

Battery Pack: Powers the whole receiver setup.

Software Tools Used:

Arduino IDE: Used to write and upload code to all microcontrollers including ESP8266, ESP32-CAM, Uno, and Nano.

Tinker cad: Used to simulate the circuit connections during the development phase.

Google Maps: Used to view the car's location by opening GPS coordinates.

Serial Monitor: Used for testing and debugging sensor and module outputs.

IV. RESULT AND DISCUSSION

The implemented system successfully achieved its objective of gesture-based control for a robotic car integrated with surveillance and navigation features. The car responded accurately to hand movements (pitch and roll) detected by the MPU6050 sensor. The gesture signals transmitted via the NRF24L01 module were reliably received by the Arduino UNO-based control system.

Key observations include:

Motion Response: The car exhibited precise Mecanum wheel movement in all four directions (forward, backward, left, right) based on hand tilt. Diagonal motion was smooth and stable.

Communication Range: Stable wireless communication was maintained up to ~20 meters indoors using the NRF24L01 module with PA+LNA.

Surveillance: The ESP32-CAM module provided live video streaming over Wi-Fi with minimal delay (~1 second), offering real-time visibility of the car's surroundings.

Navigation: The GPS module accurately tracked the car's location, and the GSM module successfully transmitted GPS coordinates via SMS upon request.

Power Supply: A single 12V battery effectively powered all modules using appropriate voltage regulation, with no overheating or power drops observed during testing.

The integration of all systems into one platform demonstrated both technical feasibility and functional performance.

Comparative Discussion

To evaluate the effectiveness of the proposed system, it was compared with conventional control and surveillance approaches. The table below highlights the key differences:

Compared to other systems, this project offers a more interactive, intuitive, and autonomous experience. While traditional RC systems rely on manual input and lack intelligence, this system utilizes natural human gestures for control, integrates a live feed for surveillance, and provides real-time location tracking—making it suitable for modern smart robotic applications, especially in areas like search-and-rescue, surveillance, and remote inspection.

V. CONCLUSION

In this project, a smart robotic car was designed and developed to operate using hand gesture control, coupled with navigation and surveillance capabilities. The system integrates multiple components including the MPU6050 sensor for gesture detection, ESP8266 for transmitting data wirelessly, NRF24L01 for reliable communication, Arduino UNO and Nano for processing at the receiver, ESP32-CAM for live video feed, and GPS-GSM modules for real-time location tracking and SMS alerts. Powered by a 12V battery and carefully regulated, the entire setup demonstrated smooth performance, reliable communication, and multi-functional coordination across modules.

The results showed that the robotic car accurately followed user hand movements, provided effective real-time surveillance, and successfully transmitted GPS location via GSM during testing. The use of Mecanum wheels enabled precise and multidirectional movement, enhancing the flexibility of the robot in constrained or dynamic environments. This work demonstrates a holistic and modular approach to robotic system design, where control, navigation, and monitoring are seamlessly integrated into a single unit. The success of this system highlights the potential for its deployment in real-world applications such as remote surveillance, search and rescue, or hazardous area inspection.

VI. FUTURE SCOPE

The current system opens several avenues for further development:

Autonomous Mode: Integration of AI-based obstacle detection and path planning using sensors like ultrasonic, LiDAR, or computer vision could allow autonomous navigation.

Voice Command Integration: Adding voice recognition could offer an additional mode of control, increasing user accessibility.

Cloud-Based Monitoring: GPS and video feed can be integrated with IoT platforms like Blynk or Firebase for cloud-based data logging and remote access from anywhere.

Security Enhancements: Facial recognition or intruder detection can be added using the ESP32-CAM for advanced surveillance features.

Range Expansion: Replacing NRF24L01 with long-range communication modules like LoRa or 4G LTE can extend usability over greater distances.

Energy Efficiency: Implementing sleep modes and solar-based charging systems could improve power efficiency for field operations.

Overall, the project forms a strong foundation for future research and innovation in gesture-controlled autonomous robotic systems.

REFERENCE

[1] A. Rathod and A. Jaiswal, “Gesture Controlled Robot using MPU6050,” International Journal of Scientific Research in Engineering and Management, vol. 4, no. 6, pp. 88–92, Jun. 2020.

[2] S. Dhumal and D. Dhone, “Wireless Gesture Controlled Robot using MPU6050 and Flex Sensors,” International Research Journal of Engineering and Technology, vol. 8, no. 4, pp. 1621–1624, Apr. 2021.

[3] K. Parate and P. Bhosale, “Accelerometer Based Wireless Gesture Controlled Robot,” International Journal of Innovative Research in Science, Engineering and Technology, vol. 9, no. 5, pp. 2341–2346, May 2020.

[4] R. Sharma and A. Agarwal, “GPS and GSM Based Vehicle Tracking System,” International Journal of Engineering and Advanced Technology, vol. 8, no. 6, pp. 5532–5535, Aug. 2019.

[5] S. Gupta and R. Jaiswal, “Wireless Surveillance Using ESP32-CAM,” International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 10, no. 3, pp. 1342–1346, Mar. 2021.

[6] A. Patil and S. Kale, “IoT Based Vehicle Tracking Using GPS and GSM,” International Journal of Computer Applications, vol. 182, no. 30, pp. 45–49, Feb. 2022.

[7] A. Shaikh and S. Pathan, “IoT Based Gesture Controlled Robot,” International Journal of Scientific & Engineering Research, vol. 11, no. 7, pp. 1041–1044, Jul. 2020.