A Smartphone-Controlled Wi-Fi Vehicle

[1]N.sony, [2]Kolekar Supriya, [3]Patnaikuni Susritha, [4]Umannagari Vineela
[1]Assistant Professor, ECE Department, BRECW.
[2,3,4] B.Tech Students, Department of Electronics and Communication Engineering, BRECW.

Abstract:
This project aims to design and build a compact, programmable vehicle capable of autonomous movement and remote control via wireless communication. The core components include an ESP32 development board, motor driver circuitry, DC motors for propulsion, and optional sensors for environmental feedback. Utilizing the capabilities of the ESP32 microcontroller, the smart car integrates seamlessly with IoT applications, enabling real-time data exchange and remote operation through Wi-Fi or Bluetooth connectivity. The development process involves assembling the chassis, integrating hardware components, and programming control algorithms in the Arduino IDE. Key functionalities such as obstacle avoidance, line following, and wireless control are implemented to enhance the car's autonomy and user interaction. This project serves as an educational exploration into robotics, embedded systems, and IoT, providing hands-on experience in hardware assembly, software development, and integration of sensor-based intelligence. By fostering creativity and technical skills, the ESP32-based smart car project offers a practical foundation for enthusiasts and learners alike to delve into the evolving landscape of smart technologies.

1- Introduction In today's rapidly advancing technological landscape, the integration of smart devices into everyday life has become increasingly prevalent. One such fascinating application is the development of a smartphone-controlled Wi-Fi car using the ESP32 :
microcontroller. This project leverages the powerful capabilities of the ESP32, a low-cost, low-power system on a chip (SoC) with integrated Wi-Fi and dual-mode Bluetooth, making it an ideal choice for IoT projects.

The primary objective of this project is to design and build a small, mobile robot car that can be controlled remotely via a smartphone. By utilizing the ESP32 microcontroller, we can establish a wireless connection between the car and a smartphone, allowing for real-time control and monitoring. This project not only demonstrates the practical use of wireless communication but also provides a hands-on experience in working with microcontrollers, motor drivers, and mobile app development.

The core component of this project is the ESP32 microcontroller, known for its robust performance and extensive features. The ESP32 offers built-in Wi-Fi and Bluetooth capabilities, multiple GPIO pins, and support for various communication protocols such as SPI, I2C, and UART[1]. These features make it an excellent choice for controlling the car's movements and communicating with the smartphone.

In addition to the ESP32, the project requires several other components, including:

2-LITERATURE SURVEY

The integration of microcontrollers in robotics has transformed how we interact with technology, particularly in the realm of remote-controlled vehicles. The ESP32 microcontroller, known for its robust features and connectivity options, has

garnered significant attention in recent years for projects involving smartphone-controlled cars.

Several studies highlight the capabilities of the ESP32, showcasing its dual-core processing power, which allows for efficient handling of multiple tasks simultaneously. Its built-in Wi-Fi and Bluetooth functionalities facilitate seamless communication with mobile devices, enabling real-time control over various distances. Research has demonstrated the effectiveness of using the ESP32 in IoT applications, particularly in remote vehicle control, where low latency and responsiveness are critical.

In many projects, the car is controlled via a mobile application or a web interface, allowing users to send commands intuitively. Various programming frameworks, such as Arduino IDE and Micro Python, support the ESP32, making it accessible for developers and hobbyists. These platforms enable the integration of additional features, such as obstacle detection using ultrasonic sensors, enhancing the car's operational safety and autonomy.

Literature also emphasizes the educational value of such projects. Engaging with smartphone controlled vehicles helps users understand fundamental concepts of electronics, programming, and networking. Many educational institutions and makerspaces have adopted similar projects to promote STEM learning, encouraging students to experiment with technology in a hands-on manner. Moreover, advancements in battery technology and energy-efficient design have contributed to the feasibility of building Wi-Fi controlled cars that operate for extended periods. Future research may focus on enhancing these systems with additional functionalities, such as camera integration for video streaming, GPS for navigation, and machine learning for autonomous driving capabilities. Overall, the development of smartphone-controlled Wi-Fi cars using the ESP32 microcontroller represents a significant intersection of technology and creativity, offering numerous opportunities for innovation and learning in robotics and IoT.

Smartphone Wi-Fi controlled cars offer a modern and engaging way to experience remote driving. Utilizing a dedicated app, users can operate the car with ease, receiving real-time feedback on speed and battery levels right on their devices. The user-friendly interface caters to all skill levels, making it accessible for beginners while providing advanced features for experienced users. With a typical control range of up to 100 meters, these cars allow for adventurous exploration. Customization options, including speed limits and steering sensitivity, enhance the driving experience, while integrated cameras enable first-person view (FPV) driving. Many models boast stunt capabilities, allowing for flips and spins at the tap of a button. Designed with durability in mind, they can withstand crashes and navigate rough terrains, making them ideal for outdoor play. Battery life is usually robust, offering hours of entertainment with rechargeable options. The cars often feature obstacle detection to avoid collisions and include safety features like an emergency stop function. Users can also enjoy community engagement by sharing tips online and benefit from regular app updates that enhance functionality. Cross-platform compatibility ensures access for both iOS and Android users, while performance modes like eco and sport offer varied driving experiences. Additionally, customizable LED lighting adds a visual flair, and integration with other smart devices can elevate the control experience. Overall, these cars serve as both a fun pastime and an educational tool, providing insights into robotics and coding through app development.

3-Software Requirements

In this chapter we will discuss and software requirements for A Smartphone Controlled Wi-Fi Car Using ESP32 Microcontroller.

Software Requirements

For a smartphone-controlled Wi-Fi car using the ESP32 microcontroller, the following software components are typically required:

Arduino IDE : A widely used platform for programming the ESP32, allowing for easy coding and uploading of sketches.

ESP32 Board Package: An add-on for the Arduino IDE that includes libraries and tools specifically for the ESP32.

MicroPython (optional): An alternative programming environment that allows users to write Python code for the ESP32, suitable for those who prefer Python over C/C++.

Mobile App Development Tools: Tools like MIT App Inventor, Flutter, or React Native for creating a custom mobile app to control the car.

Web Development Tools (optional): HTML, CSS, and JavaScript for developing a web-based interface if choosing to control the car via a browser.

Libraries: Various libraries for specific functionalities, such as Wi-Fi control, motor driver control (e.g., Adafruit Motor Shield library), and sensor integration (e.g., Ultrasonic sensor libraries).

Firmware: If using specific motor drivers, ensure to have the necessary firmware or libraries required for compatibility.

Version Control System (optional): Tools like Git for managing code changes and collaboration if working in teams.

These software components collectively enable the development, programming, and control of the Wi-Fi car project, enhancing functionality and user interaction.

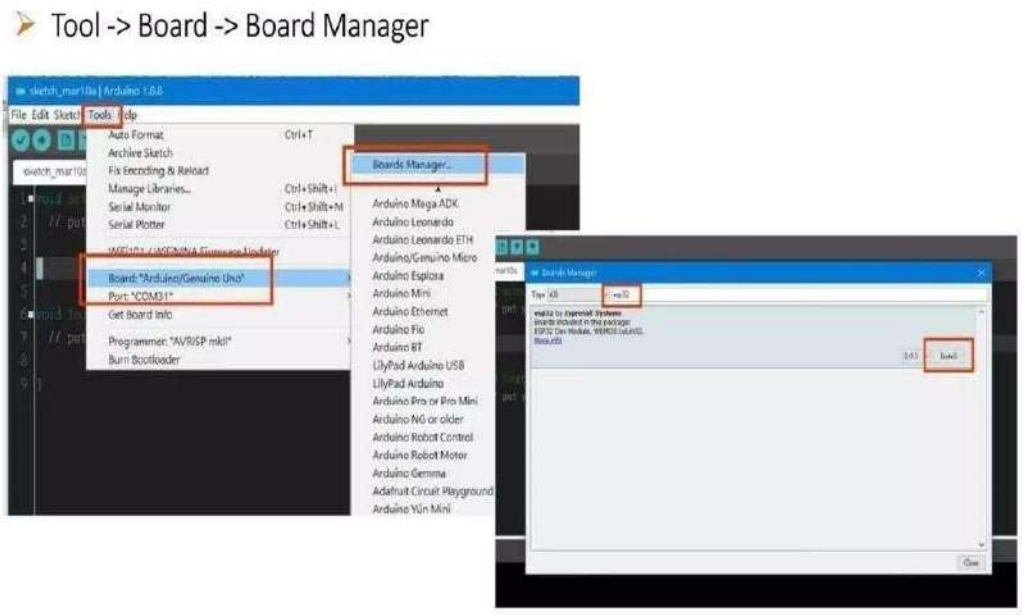

Figure 2.1 : Arduino SDK Install

4-ESP32 WIFI BASED SMART CAR

In this chapter we will discuss about Existing/Proposed System, block diagram and methodology for Compressive spectrum analysing the process of ESP32 chip.

Existing System

Before the advent of Wi-Fi-enabled systems like the ESP32 smart camera car, most remotecontrolled (RC) vehicles relied on radio frequency (RF) technology. RF-controlled systems have been widely used in toy cars, drones, and other RC vehicles for decades. Despite their limitations, RF-based systems formed the backbone of remote-control technology for a long time, allowing hobbyists and enthusiasts to interact with devices over a wireless connection. This system, however, has several constraints compared to modern alternatives like Wi-Fi, which have driven a shift toward more sophisticated platforms like the ESP32.

Proposed System

The **ESP32 Wi-Fi-based camera car** represents a significant advancement over traditional RFcontrolled systems, leveraging modern microcontroller technology, Wi-Fi connectivity, and integrated camera modules to offer a more versatile and user-friendly experience. This proposed system takes remote-controlled cars to a new level by enabling global control, real-time video streaming, and potential for autonomous operation. It is an ideal solution for a variety of applications, including remote surveillance, FPV (First Person View) driving, and educational robotics.

Block Diagram

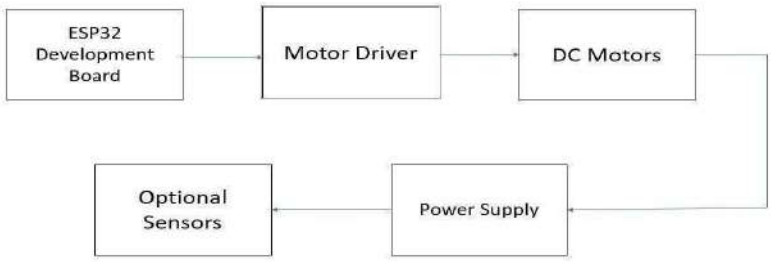

Figure 3.2: Block Diagram of ESP32 Wi-Fi car

Power Supply: Provides power to the ESP32, motors, camera, and sensors using a battery or power
bank.

ESP32 Microcontroller: Central unit that controls the car, processes Wi-Fi commands, and manages all components.

Wi-Fi Module: Built-in Wi-Fi enables remote control of the car via a mobile app or web interface over the internet.

5- ADVANTAGES, DISADVANTAGES AND APPLICATIONS

Advantages

Here are the advantages of a smartphone-controlled Wi-Fi car using the ESP32 microcontroller, with more detail on each point:

Wireless Control:

The ESP32 allows the car to be controlled via Wi-Fi, providing significant convenience. Users can operate the car from a considerable distance, unrestricted by wires or physical remotes. This flexibility enhances the user experience, making it easier to maneuver in various environments. The ability to connect through a smartphone app means that users can have multiple control options at their fingertips, adapting to different scenarios effortlessly.

Disadvantages

Here are some disadvantages of a smartphone-controlled Wi-Fi car using an ESP32 microcontroller:

Limited Range: The effective range of Wi-Fi connections can be limited compared to other communication methods like RF. This restricts the distance over which the car can be controlled, making it challenging for outdoor use or larger areas.

Battery Life: Both the smartphone and the ESP32-powered car require power. Continuous Wi-Fi communication can drain the battery quickly, potentially leading to operational issues during use or requiring frequent recharges.

Applications

Here are some applications of a smartphone-controlled WiFi car using an ESP32 microcontroller:

Educational Projects: These cars are often used in robotics and electronics education. Students can learn about programming, wireless communication, and circuit design by building and controlling their own Wi-Fi cars, enhancing hands-on learning experiences.

Remote Surveillance: The Wi-Fi car can be equipped with cameras or sensors, making it useful for remote monitoring. Users can control the car to navigate through areas for surveillance or inspection, providing a mobile platform for real-time video feeds.

6-Results and Discussion

In this chapter, we will discuss about the results of the A Smartphone Controlled Wi-Fi Car Using ESP32 Microcontroller.

- The concept of a smartphone-Wi-Fi controlled vehicle, which allows users to remotely operate a car or robot using their mobile devices. This innovative technology combines wireless connectivity and smartphone integration to provide a convenient and user-friendly driving experience.
- Unlike traditional RF-controlled cars, which operate on specific frequency bands and have limited range, WiFi-controlled cars utilize the widely available WiFi networks, providing enhanced range, stability, and data transfer rates.
- ESP32 is a powerful and versatile microcontroller with built in Wifi and Bluetooth functionalities, to Create a remotely operated vehicle

Working

A smartphone-controlled Wi-Fi car using the ESP32 microcontroller operates by leveraging the microcontroller's integrated Wi-Fi capabilities to establish a connection with a mobile device. Initially, the ESP32 is programmed to act as a Wi-Fi access point or connect to an existing network, allowing the smartphone to send control commands via a dedicated app or web interface. The user interfaces with the car through a graphical interface on the smartphone, typically using virtual buttons for direction control (forward, backward, left, right) and additional features like camera activation if equipped. The commands sent from the smartphone are received by the ESP32, which interprets them and drives the motors accordingly, utilizing motor drivers for speed and direction control. The car's movement is facilitated by DC motors or servo motors, enabling precise maneuverability. Additionally, the ESP32 can handle real-time video streaming from a camera mounted on the car, providing the user with live feedback during operation. Sensors can also be integrated to enhance functionality, enabling features like obstacle detection and avoidance. The whole system allows for remote operation within a certain range, making it ideal for exploration and fun. With its versatility, this setup serves as an excellent platform for learning about robotics and IoT applications, while also paving the way for

more advanced features such as automation and artificial intelligence in future iterations.

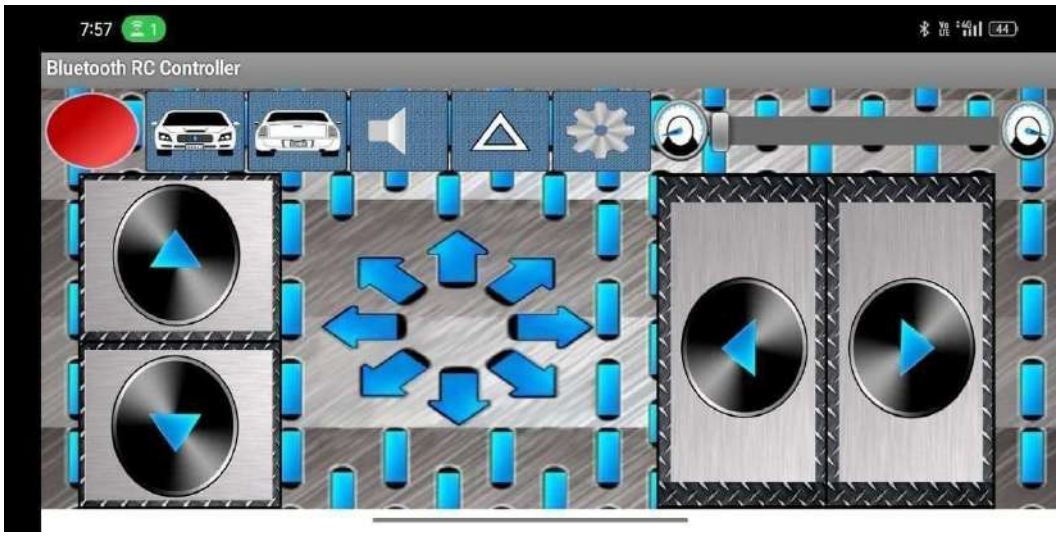

Figure 6.1: Bluetooth RC Controller

The figure shows a mobile app interface for a Bluetooth RC controller. The app appears to be designed to control a remote-controlled car.

Here are some of the features visible in the app:

Bluetooth connection status: The top left corner shows the time and Bluetooth connection status.

Car controls: The center of the screen displays buttons for controlling the car's movement, including forward, backward, left, right, and rotation.

Other controls: There are also buttons for horn, lights, and other functions.

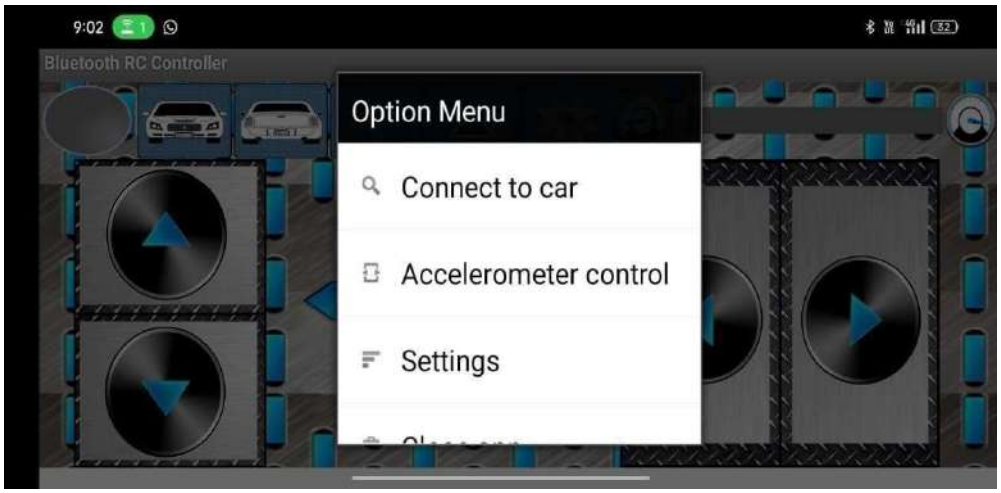

Figure 5.2: Bluetooth RC Controller

The figure shows the option menu of a Bluetooth RC controller app. The menu contains the following options:

Connect to car: This option allows the user to connect to the RC car.

Accelerometer control: This option enables the user to control the car using the accelerometer on their device.

Settings: This option allows the user to access the app's settings.

It seems that this app provides a variety of features for controlling an RC car, including both buttonbased controls and accelerometer-based control.

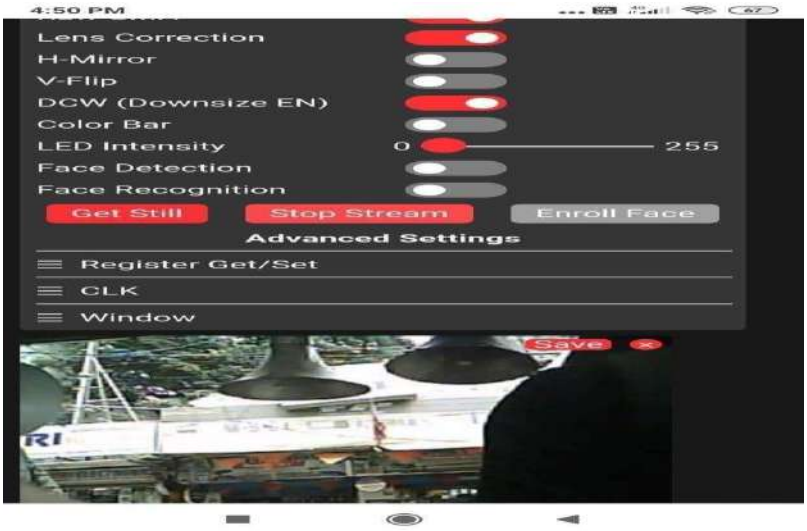

Figure 5.2:Video Streaming The above Figure Shows The Video Streaming. It shows a menu of settings and options for the camera.

7- CONCLUSION

An ESP32 WiFi camera car controlled via a smartphone is an exciting project that merges wireless communication, robotics, and real-time video streaming into one versatile system. Powered by the ESP32 microcontroller, known for its robust WiFi and Bluetooth capabilities, the car can be remotely operated using a smartphone app or a web-based interface. This makes it possible to control the movement of the car, while simultaneously receiving live video feed from an onboard camera, streamed directly to the smartphone. One of the key advantages of this setup is the cost-effectiveness and accessibility of the components. The ESP32 is a popular choice for IoT projects due to its affordability, small size, and powerful features, including built-in WiFi and camera support. The project also demonstrates how readily available hardware components such as motors, chassis kits, and cameras can be combined with simple coding to create a functional and responsive remote-controlled car.

From an application standpoint, this project offers wide-ranging possibilities. The ability to stream live video in real-time makes it ideal for remote surveillance or monitoring tasks, allowing users to explore and navigate spaces remotely. It could also be used for educational purposes, where students and hobbyists can learn about IoT, wireless communication, and robotics by building and programming their own car. Furthermore, it opens up potential in areas like obstacle avoidance, automation, and advanced features like object tracking or path planning, which can be integrated

with additional sensors. Overall, the smartphone-controlled ESP32 WiFi camera car highlights the practical implementation of IoT technology, offering a hands-on approach to understanding concepts like remote control, real-time data transmission, and robotics. It's an engaging, budget-friendly project for hobbyists, students, and developers interested in exploring the intersection of robotics and wireless technology.

REFERENCES

1. Espressif. (2016). ESP32 Technical Reference Manual.
2. Bhatia, H., et al. (2022). Swarm Robotics: Principles and Applications. International Journal of Robotics Research.
3. Gupta, R., et al. (2022). Obstacle Avoidance Systems for Mobile Robots. Journal of Robotics and Automation.
4. Khan, A., et al. (2021). The Impact of Hands-on Learning on Engineering Education. Journal of Educational Technology.
5. Kolb, D. A. (1984). Experiential Learning: Experience as the Source of Learning and Development.
6. Kumar, S., et al. (2020). Design and Implementation of an Autonomous Vehicle. Journal of Intelligent Systems.
7. Lee, J., et al. (2023). GPS Integration in Robotics: Challenges and Solutions. Robotics and Automation Letters.
8. Sharma, P., et al. (2021). User Interface Design for Mobile Robotics: Best Practices. International Journal of Human-Computer Interaction.
9. Zhang, Y., et al. (2021). MQTT-based Remote Control System for IoT Applications. IEEE Internet of Things Journal. converter." IEEE J. Emerg. Sel. Topics Circuits Syst., vol. 2, 2012.