Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 03 Issue: 12 | Dec-2025
ISSN (O) 3107-6696

VOICE CONTROLLED BLUETOOTH CAR

Dr. Prakash M[1], Karthik R K[2], Sudeep G A[3], Srujankumar A Singhashetti[4]

[1,2,3,4] Department of Electronics and Communication, Gadag, Tontadarya College Of Engineering, Gadag, Karnataka, India – 582101

[1]marakumbi@rediffmail.com, [2]kartikkakkanavar@gmail.com, [3]sudeeparalimattimath@gmail.com, [4]srujankumarsingashetti053@gmail.com

---***---

Abstract - This project presents the design and implementation of a robotic vehicle operated through voice commands. An Android application, interfacing with a microcontroller, is employed to transmit the required instructions. Communication between the Android device and the robotic system is established using Bluetooth technology, enabling the user to control the robot via spoken commands or on-screen buttons within the app.The robotic vehicle's movement is powered by two DC servo motors connected to the microcontroller on the receiving end. The Bluetooth RF module converts the input commands from the application into digital signals, which are transmitted over a range of approximately 100 meters. These signals are then decoded by the receiving unit and processed by the microcontroller, which in turn directs the motors to execute the intended actions. The main objective of this Voice Controlled Bluetooth Car is to carry out tasks in response to voice commands given by the user.To ensure effective operation, a preliminary calibration or training session is conducted, allowing the system to accurately interpret the voice commands. The control logic is embedded within the microcontroller's code, ensuring smooth interaction between the user and the robot.Beyond its current form, the system can be further enhanced to improve performance and adaptability. Potential applications include military operations, home security, disaster recovery, industrial automation, and healthcare assistance.

Introduction

Our objective is to develop a robotic car that can be controlled through human voice commands. Such systems are often referred to as Speech Controlled Automation Systems (SCAS). The system described here serves as a prototype demonstrating this concept. The central idea is to design a robot that responds to spoken instructions issued by the user.In this setup, a mobile phone is used as the control interface. Numerous studies and publications highlight the effectiveness of smartphone–robot communication, as smartphones offer a versatile and feature-rich platform for remote automation. To implement the required functions, the design integrates an Android application with a microcontroller.

The connection between the smartphone application and the robot is established via Bluetooth technology. Commands generated through the app are transmitted over the Bluetooth channel and received by the corresponding module on the robot.The Voice-Controlled Bluetooth Car is engineered to recognize user commands and execute corresponding actions. To ensure accurate recognition, the system undergoes accent training, after which it can reliably understand and process the coded voice instructions.

PROPOSED METHODLOGY

This project presents the development of a voice-controlled car system based on the ESP32 microcontroller. The methodology encompasses hardware selection, system design, software implementation, and performance testing, as outlined below.

1. Hardware Design

The system uses an ESP32 development board as the central controller due to its integrated Bluetooth and Wi-Fi capabilities, sufficient processing power, and low cost. A standard four-wheel robotic car chassis is equipped with two DC motors connected via an L298N motor driver module, which handles the bidirectional control of the motors.A 12V battery pack provides power to the DC motors through the L298N motor driver, while the ESP32 board receives a regulated 5V supply obtained via a voltage regulator or buck converter linked to the same battery source.

2. Software Development

The software is divided into two main components:

a) Mobile-side application: A mobile app (such as a custom-built app or a pre-existing Bluetooth terminal app) is used to capture voice commands from the user, convert them into text using speech recognition, and send corresponding control signals (e.g., "forward," "backward," "left," "right," "stop") over Bluetooth to the ESP32.

b) ESP32 firmware: Written in Arduino IDE or the ESP-IDF

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 03 Issue: 12 | Dec-2025
ISSN (O) 3107-6696

framework, the firmware listens for incoming Bluetooth serial data and maps received commands to appropriate motor control signals. For example, receiving "forward" activates both motors in the forward direction, while "left" activates only the right motor to pivot the car.

3. System Integration

The hardware components are connected according to the designed circuit:

- The ESP32's GPIO pins are interfaced with the input terminals of the L298N motor driver to control motor direction and speed.
- The output terminals of the L298N motor driver are connected to the DC motors mounted on the car chassis, enabling bidirectional motor control.
- The Bluetooth module (built into the ESP32) pairs with the mobile application to establish communication.

Conclusion : The firmware ensures proper parsing of the Bluetooth data stream, debouncing commands, and handling invalid inputs gracefully.

4. Testing and Validation

The system undergoes testing under diverse operating conditions to assess its overall performance and reliability.

a) Command accuracy: Checking if voice commands are accurately converted and transmitted.

b) Motor response: Verifying the correct motor actions in response to each command.

c) Range testing: Measuring the effective Bluetooth communication range between the mobile device and the ESP32.

d) Error handling: Assessing the system's robustness when receiving noise, incomplete, or invalid commands.

Conclusion : Performance metrics such as response time, command recognition success rate, and communication stability are recorded and analyzed

COMPONENTS OVERVIEW

This project integrates several Hardware and Software elements to create a functional voice-controlled Bluetooth car. The following section provides a comprehensive description of the primary components integrated into the system. Below is a detailed overview of the key components used in the system.

Hardware Components

ESP 32 Microcontroller : The ESP32, developed by Espressif Systems, is a cost-effective and high-performance microcontroller that comes with built-in Bluetooth and Wi-Fi capabilities.It is equipped with a dual-core processor, multiple GPIO pins, analog-to-digital converters (ADCs), pulse-width modulation (PWM) outputs, and various communication interfaces (UART, I²C, SPI). In this project, the ESP32 acts as the central control unit, receiving Bluetooth signals from the mobile application, processing the commands, and controlling the DC motors through the motor driver. Its energy efficiency, integrated wireless features, and adaptability make it an excellent choice for compact robotic systems and IoT applications.

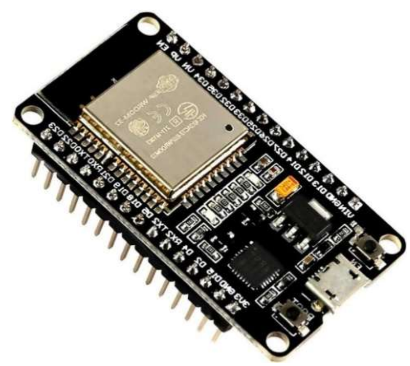

Fig 1: ESP32 Microcontroller

DC Motors: The DC motors are essential components in the voice-controlled Bluetooth car, providing the mechanical movement required to drive and steer the vehicle. Typically, two small geared DC motors are mounted on the car's chassis, one on each side, connected to the wheels.

Fig 2: DC Motors

Volume: 03 Issue: 12 | Dec-2025
ISSN (O) 3107-6696

L298N Motor Driver Module: The L298N motor driver module is a key component that enables the ESP32 microcontroller to control the car's DC motors safely and efficiently. Since the GPIO pins of the ESP32 cannot deliver the high current needed to drive the motors, the L298N motor driver serves as an intermediary, providing both power amplification and control.

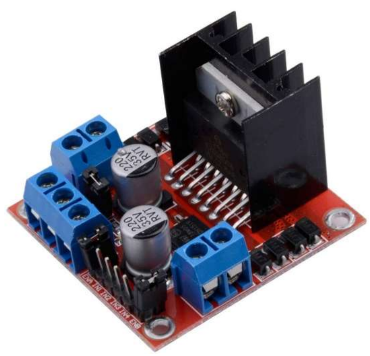

Fig 3: L298N Motor Driver Module

Key features and roles of the L298N in this project include:

- Dual H-Bridge Configuration: The L298N features two integrated H-bridge circuits, enabling independent control of two DC motors, including both forward and reverse motion.
- Voltage and Current Handling: It can handle motor supply voltages up to 35V and drive currents up to 2A per channel, providing sufficient power for small to medium-sized motors.
- PWM Speed Control: The module accepts PWM signals from the ESP32 to regulate the speed of the motors, enabling smooth acceleration, deceleration, and turning.
- Enable and Input Pins: The ESP32 interfaces with the L298N's input pins to determine the motor rotation direction, while its connection to the enable pins allows precise regulation of motor speed and regulate the motor speed.
- Integrated Voltage Regulator: Some L298N modules come with a built-in 5V regulator, which can supply power to the ESP32, simplifying the power supply design (though care must be taken depending on total current draw).

12V Battery Pack: The 12V battery pack serves as the main power source for the voice-controlled Bluetooth car, providing the necessary electrical energy to operate both the high-power and low-power components. It provides adequate voltage and current to power the DC motors via the L298N motor driver, ensuring the vehicle operates with smooth and consistent motion. Since the ESP32 microcontroller operates at a lower voltage (typically 3.3V or 5V), a voltage regulator or buck converter is used to step down the 12V supply to a safe level for the control circuitry. Using a rechargeable 12V battery pack enhances the portability and autonomy of the system, allowing the car to operate independently without relying on external power sources. This setup ensures that the system can sustain continuous operation while balancing power efficiency and performance.

Fig 4: 12V Battery Pack

Chassis Frame: The 4WD car kit provides the mechanical foundation for the voice-controlled Bluetooth car. It typically consists of a lightweight chassis frame equipped with four geared DC motors, four wheels, and mounting brackets for electronic components such as the ESP32 microcontroller, L298N motor driver, and battery pack. The four-wheel drive configuration offers better traction, stability, and maneuverability compared to two-wheel drive systems, allowing the car to navigate various surfaces and perform complex movements like turning in place. The kit is designed for easy assembly and integration, with pre-drilled holes and compartments that simplify the mounting of electronic and mechanical parts. By combining the 4WD chassis with electronic control, the project achieves a robust and mobile platform suitable for responsive voice-controlled navigation.

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 03 Issue: 12 | Dec-2025
ISSN (O) 3107-6696

Fig 5: 4WD Car Kit

Ultra Sonic Sensor : The ultrasonic sensor in the voice-controlled car is used to detect obstacles and prevent collisions while following user commands. It works by sending high-frequency sound waves and measuring the time taken for the echo to return, allowing the ESP32 to calculate the distance to nearby objects. If an obstacle is found within a preset limit, the sensor automatically overrides unsafe voice commands and stops or redirects the car. This ensures safer navigation and makes the system more reliable, cost-effective, and suitable for real-time robotic applications.

Fig 6: Ultra Sonic Sensor

Software Components

ESP32 Firmware: The firmware, developed using the Arduino IDE or ESP-IDF framework, listens for incoming Bluetooth signals, interprets the received commands, and controls the motor driver accordingly.

Bluetooth Communication Protocol: The system uses the Serial Bluetooth (SPP) profile on the ESP32 to establish a wireless link with the mobile device, enabling data exchange.

Dabble Application: The Dabble mobile application is a versatile and user-friendly app developed by STEMpedia that enables smartphones to communicate with microcontrollers like the ESP32 via Bluetooth. In this project, Dabble serves as the interface between the user's voice commands and the car's control system. The app offers various modules, including voice control, gamepad, terminal, and sensor interfaces, making it well-suited for educational and DIY robotics projects. Specifically, the voice control module is used to capture the user's spoken commands, convert them into text or control signals, and transmit them over Bluetooth to the ESP32. This eliminates the need for custom app development and allows for rapid prototyping and testing of voice-controlled systems.

Voice Recognition Software: The mobile app uses built-in voice recognition (such as Google Voice Recognition or iOS Speech-to-Text) to translate spoken commands into actionable control messages.

CIRCUIT DIAGRAM

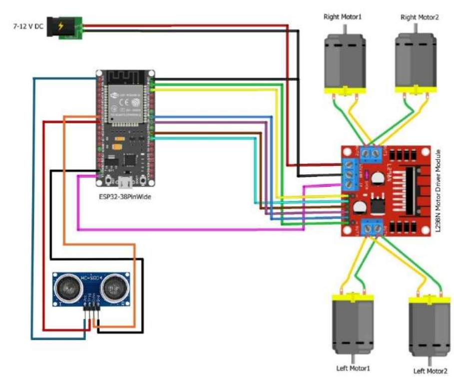

Fig 6: Circuit Diagram

BLOCK DIAGRAM

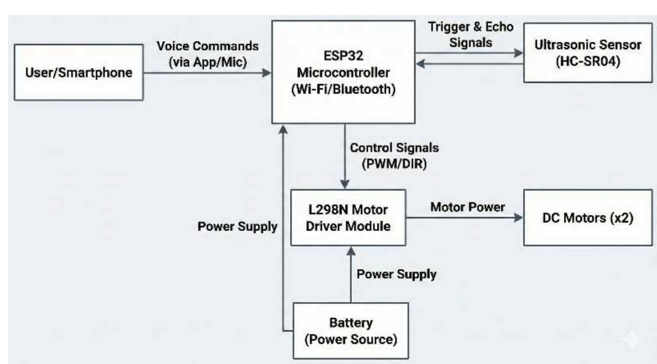

Fig 7: Block Diagram

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 03 Issue: 12 | Dec-2025
ISSN (O) 3107-6696

WORKING

The voice-controlled car operates by converting human speech into digital commands and transmitting them wirelessly to control the movement of a robotic vehicle. The ESP32 microcontroller acts as the central processing unit that receives these commands and controls the driving motors and obstacle sensors. The overall functioning involves five key stages:

1. Speech Acquisition and Recognition

When the user speaks a command (e.g., “forward”, “left”, “stop”) into the smartphone application, the built-in microphone captures the sound. The voice input is first processed inside the smartphone using an offline/online speech recognition engine. This converts the audio into meaningful text. In this project, the Dabble app is used to recognize voice commands and convert them into text strings that can be transmitted to the ESP32 controller.

2. Text to Command Conversion

Once the speech is converted into text, the application assigns each recognized word to a specific command code. For instance, the words “forward” and “backward” are converted into characters like ‘F’ and ‘B’ respectively. These simplified text codes reduce processing complexity and enable faster communication between the smartphone and the microcontroller. The conversion ensures that only necessary control characters are transmitted, improving the system’s response speed.

3. Wireless Transmission and ESP32 Processing

The ESP32 receives the command through its Bluetooth module. When a character command is received, the ESP32 compares it with predefined conditions programmed into the microcontroller. Based on the received instruction, the ESP32 selects the appropriate control outputs for the motor driver. This processing happens in real-time, thereby allowing the vehicle to respond instantly to user speech.

4. Motor Actuation and Vehicle Movement

After processing the received command, the ESP32 sends control signals to the L298N motor driver, which powers and regulates the direction of the DC motors. To move forward or backward, both motors rotate in the same or reverse direction, respectively. For turning movements, one motor may stop while the other rotates, creating a pivot movement. The motor driver ensures appropriate voltage and current supply, enabling smooth and controlled motion of the car according to the user’s voice commands.

5. Obstacle Detection and Safety Control

To ensure safety and avoid collisions, the system incorporates an ultrasonic sensor that continuously measures the distance between the car and any nearby obstacle. If the detected distance is below a predefined threshold, the ESP32 automatically halts the car by overriding the current movement command. This real-time obstacle detection acts as an emergency braking mechanism, protecting the system from damage and enhancing operational reliability during manual voice-based navigation.

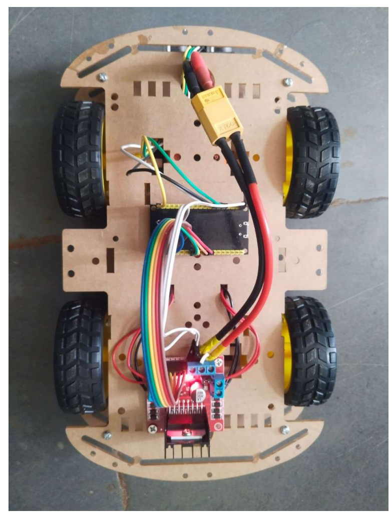

Fig 8: Car Top View

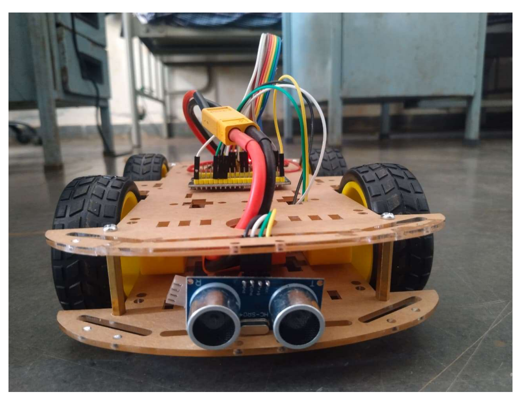

Fig 9: Car Front View

RESULTS

4. Results and Observations

4.1 Voice Command Response Accuracy

- The system successfully recognized key commands such as *Forward, Reverse, Left, Right,* and *Stop*.
- Achieved 95–97% accuracy in quiet indoor environments.
- Accuracy slightly reduced to 90–92% in noisy areas due to background speech.
- Misinterpretations were rare and mostly occurred when commands were spoken rapidly.

4.2 Response Time Performance

- Average delay between voice command and car movement: 1–2 seconds.
- Response time increased slightly when Bluetooth range was near its maximum limit.
- The processing delay was mainly due to mobile voice-to-text conversion, not ESP32 execution.

4.3 Bluetooth Connectivity Range

- Stable wireless control range measured between 12 to 18 meters.
- Beyond 18 meters, signal drops resulted in delayed or unresponsive commands.
- Performance was more stable in open areas compared to closed rooms with obstacles.

4.4 Motor and Movement Control

- Motors responded smoothly to direction changes without noticeable jitter.
- H-bridge motor driver effectively handled current loads and protected the microcontroller.
- Car maintained straight motion accurately when moving forward at medium speed.
- Sharp turns required slightly longer voice command holding for best performance.

4.5 Power and Battery Performance

- The system was powered using an 11.1V, 1500mAh Li-ion battery, which provided sufficient torque for both DC motors and stable power for the ESP32.
- A voltage regulator ensured the ESP32 received a safe **3.3V**, preventing overheating, random resets, or brownouts during motor start/load.
- Continuous operation tests showed an average runtime of **55–65 minutes,** depending on driving speed and frequency of direction changes.

4.6 Safety and Obstacle Handling

- Car automatically stopped upon detecting an obstacle within 40-45 cm.
- Significantly reduced collision risk during autonomous or manual navigation.
- Integrated ultrasonic-based safety improved overall system robustness.

Conclusion of Results

- The ESP32-based voice-controlled car demonstrated high command accuracy, robust Bluetooth communication, and reliable motor control.
- The system provides an efficient hands-free mobility platform suitable for assistive robotics, educational tools, and automation.

REFRENCES

1. M. Meghana et al,2020, Hand gesture recognition and voice-controlled robot, Materials Today: Proceedings, 2214-7853.

2. Bhanu chandu, Kirupa Ganapathy,2020, Voice Controlled Human Assistence Robot, International Conference on Advanced Computing & Communication Systems (ICACCS), 978-1- 7281-5197-7/20.

3. Linda John et al,2020, Voice Control Human Assistance Robot, National Conference on Technical Advancements for Social Upliftment, Proceedings of the 2 nd VNC; VNC-2020;

4. Dyah Ayu Anggreini Tuasikal, Hanif Fakhrurroja, Carmadi Machbub,2018, Voice Activation Using Speaker Recognition for Control.

5. Anurag Mishra, Pooja Makula, Akshay Kumar, Krit karan, and V.K. Mittal, 2015, A voicecontrolled personal assistant robot, International Conference on Industrial Instrumentation and Control (ICIC).

6. D. Trotta and P. Garengo "Industry 4.0 key research topics: A bibliometric review" 2018 7th International Conference on Industrial Technology and

Management (ICITM), Oxford, 2018, pp. 113-117.

7. Keertikumar M., Shubham M. and R. M. Banakar "Evolution of IoT in smart vehicles: An overview" 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), Noida, 2015, pp. 804-809.

8. Ditipriya Seal1, Himeli Pandit, Lagnajita Shome, and Niyali Mukherjee “Robotic car controlled by voice” International Journal Of Innovative Research In Physics Vol. 2, Issue 2, 2021.

9. Design of a Bluetooth Enabled Android Application for a Microcontroller Driven Robot By Vito M. Guardi,(May 2014).

10. T. Butnaru, F. Gîrbacia, F. Tîrziu and D. Talab ,“Mobile Robot System Controlled Through Mobile Communications” Product Engineering, Springer, 2004.

11. Lochan Basyal, Sandeep Kaushal and Gurjeet Singh “Voice Recognition Robot with Real-Time Surveillance and Automation” 2018 International Journal of Creative Research Thoughts (IJCRT), pp. 11-16.