**WIFI AND VOICE CONTROLLED ROBOT**

by

**SANTHOSH KUMAR S 21BEC1283**

**VISHAL**

A project report submitted to

# Dr. VELMATHI G

**SCHOOL OF ELECTRONICS ENGINEERING**

in partial fulfillment of the requirements for the course of

BECE312L

ROBOTICS AND AUTOMATION

in

# B. Tech. ELECTRONICS AND COMMUNICATION ENGINEERING



**Vandalur – Kelambakkam Road Chennai – 600127**

**APRIL 2024**

**BONAFIDE CERTIFICATE**

It is certified that this project report entitled “**WIFI AND VOICE CONTROLLED ROBOT”**is a bonafide work of SANTHOSH KUMAR S **-(21BEC1283)**

who carried out the Project work under my supervision and guidance for

**BECE312L- ROBOTICS AND AUTOMATION.**

# Dr. VELMATHI-G

Professor Higher Academic Grade School of Electronics Engineering (SENSE),

VIT University, Chennai Chennai – 600 127.



INTRODUCTION

Introducing the WiFi and Voice-Controlled Robot, a groundbreaking fusion of cutting-edge technology and intuitive interaction. This innovative project harnesses the power of WiFi connectivity and voice recognition, enabling users to control the robot effortlessly from anywhere within the network range. With customizable functionality and real-time feedback through integrated sensors and cameras, users can tailor the robot's behavior to suit their specific needs, whether it's home automation, educational exploration, or entertainment. From delegating tasks to monitoring surroundings, the possibilities are endless. With its seamless integration of wireless connectivity, voice control, and programmable interfaces, the WiFi and Voice-Controlled Robot heralds a future where human-robot interaction is natural, convenient,andlimitless.

1. **Related Work:** Prior research and development in the field of robotics have laid the groundwork for the WiFi and Voice-Controlled Robot project. Existing works have explored various aspects of wireless connectivity, voice recognition, and robotic control systems. For instance, studies have delved into the integration of WiFi modules for remote control and data transmission in robotic applications, highlighting the importance of stable and secure communication protocols. Additionally, research in voice recognition technology has advanced the capabilities of natural language processing, enabling robots to interpret and respond to verbal commands with greater accuracy and efficiency. Moreover, efforts in the field of human-robot interaction have focused on enhancing user experience through intuitive interfaces and customizable functionalities, paving the way for the development of interactive and user-friendly robotic systems. By building upon these foundations, the WiFi and Voice-Controlled Robot project aims to push the boundaries of innovation and deliver a seamless and engaging user experience in the realm of robotics.
2. **APPLICATIONS:**

The WiFi and Voice-Controlled Robot project holds significant potential for a wide range of applications across various domains. In the realm of home automation, the robot serves as a central hub for controlling household appliances, monitoring security, and managing tasks remotely. Its ability to respond to voice commands and navigate through environments makes it an invaluable asset for simplifying daily routines and enhancing convenience for users. Additionally, in educational settings, the robot serves as an interactive platform for teaching robotics, programming, and artificial intelligence concepts. Students can engage in hands-on learning experiences, programming the robot to perform tasks and explore scientific principles in a dynamic and engaging manner. Furthermore, in entertainment applications, the robot's ability to entertain guests, provide companionship, and perform interactive tasks adds a new dimension to social interactions and leisure activities. Whether it's entertaining guests with interactive performances or assisting with household chores, the WiFi and Voice-Controlled Robot project offers endless possibilities for enhancing productivity, learning, and entertainment experiences.

# System Design and Methodology:

# The design and methodology of the WiFi and Voice-Controlled Robot project involve a systematic approach to integrating hardware and software components for seamless operation and user interaction. The system architecture comprises several key elements:

# 1. \*\*Hardware Components:\*\* This includes the physical components of the robot, such as motors, sensors, cameras, microcontrollers, and WiFi modules. These components are carefully selected and integrated to ensure compatibility and optimal performance.

# 2. \*\*Software Development:\*\* The software stack encompasses various layers, including firmware for controlling hardware components, communication protocols for WiFi connectivity, and algorithms for voice recognition and robotic behavior. Open-source frameworks and libraries may be utilized to expedite development and ensure robustness.

# 3. \*\*Integration of Voice Recognition:\*\* A critical aspect of the project involves implementing voice recognition capabilities to enable users to interact with the robot using natural language commands. This may involve training machine learning models or utilizing pre-trained models for recognizing speech patterns and converting them into actionable commands.

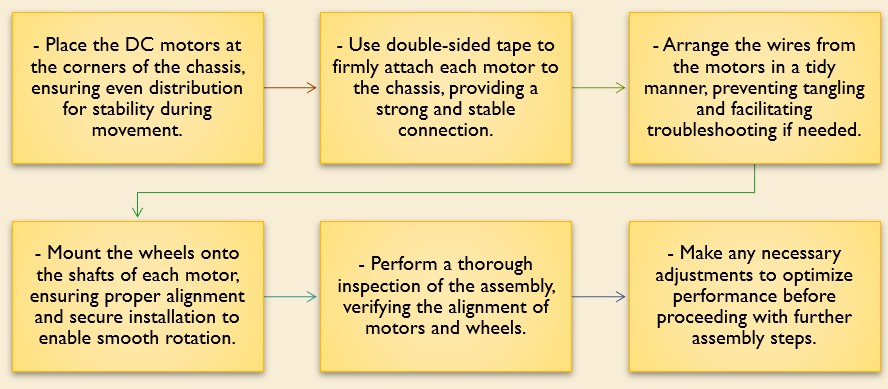
# 4. \*\*Wireless Connectivity:\*\* The robot's WiFi module facilitates wireless communication with external devices, such as smartphones, computers, or home automation systems. Protocols such as MQTT (Message Queuing Telemetry Transport) or WebSocket may be employed for real-time data exchange and remote control.

# 5. \*\*User Interface:\*\* A user-friendly interface is developed to enable users to interact with the robot effortlessly. This may include a mobile app, web-based interface, or voice-enabled assistant for issuing commands and monitoring the robot's status and behavior.

# 6. \*\*Testing and Optimization:\*\* Rigorous testing is conducted throughout the development process to identify and rectify any hardware or software issues. Performance optimization techniques may be employed to enhance the robot's responsiveness, accuracy, and efficiency in executing commands and navigating environments.

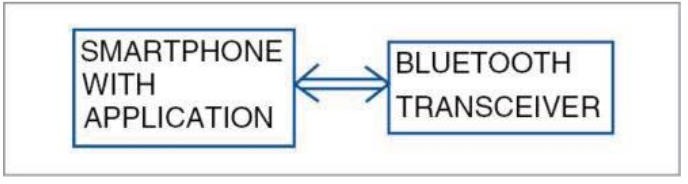
# 7. \*\*Scalability and Flexibility:\*\* The system design is scalable and flexible to accommodate future enhancements and customizations. Modular design principles are adopted to facilitate easy upgrades or modifications to hardware components and software functionalities. By adopting a systematic approach to system design and methodology, the WiFi and Voice-Controlled Robot project aims to deliver a robust, user-friendly, and versatile robotic system capable of fulfilling a wide range of applications and user requirements.

# HARDWARE SETUP:



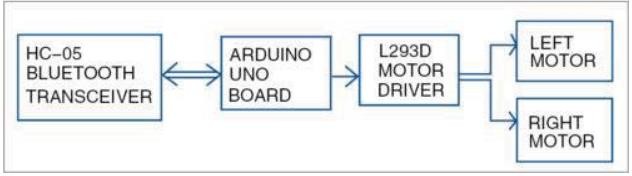
**Transmitter and Receiver**

On the transmitter section, various commands are given to the mobile app through the microphone of the mobile. This mobile is connected to the robotic car via Bluetooth. The mobile app used is programmed in such a manner that the voice commands given to the mobile are received by the microphone and these analog voice commands are converted to digital word sequences (A to D conversion). These stored sequences are then transmitted to the robot via Bluetooth transceiver module and consequently sent to the transceiver controller (MAX 232).



# Block diagram of transmitter side:

The received signal is decoded by MAX 232 transceiver. It is also used for serial communication with the Bluetooth module. The controller compares these digital signals with the stored programme commands inside and convert the digital signal into voice strings. The voice strings are thereafter used to run the servo motors for the required interval of time.



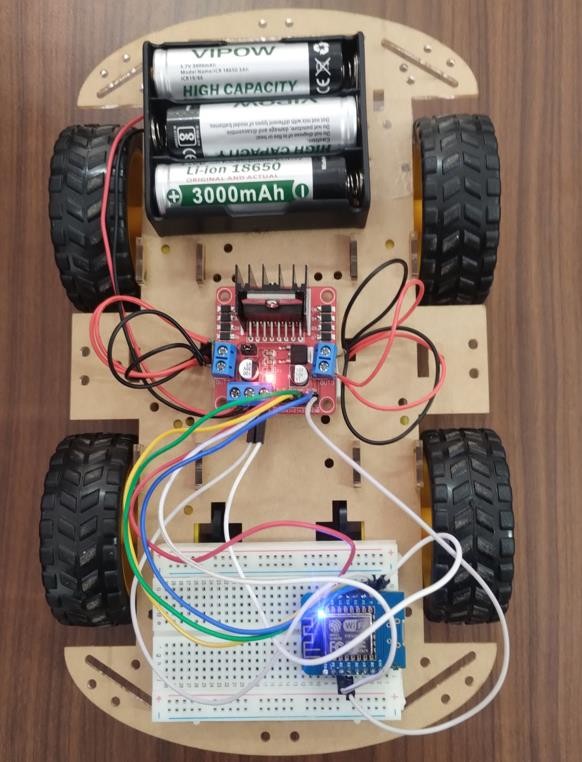
# L239D motor driver

**Block diagram of the receiver side**

It is a dual H-Bridge high current motor driver IC .This is used in the project since digitals pins of arduino cannot source enough current to drive the motors of the robotic car. H-bridges are also very useful in controlling the direction of rotation of the motor. It enable pins 1 and 9 of the IC, being active high, are

connected to 5V.Four output pins of L239D IC are connected to motors M1 and M2 on the receiver side.

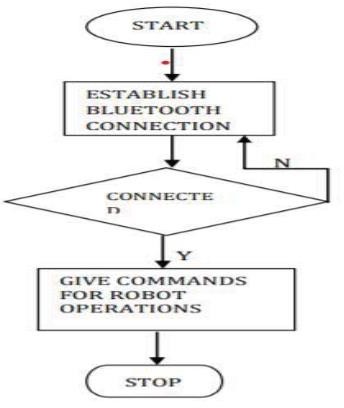
**MODEL:**



**COMPONENTS:**

* Wemos D1 mini - 1x
* L298N motor driver - 1x
* DC gear motor - 4x
* Mini breadboard - 1x
* 3.7v Lithium ion battery - 3x
* Chassis - 1x
* Rubber wheels - 4x
* Jumper wires

# FLOWCHART:



**Algorithm**

1. Start
2. Establishing Bluetooth connectivity between Android Application and the Bluetooth module on the robot.
3. To Check whether the device is connected.
4. If connected, provide the pre-defined instructions or commands to the micro- phone of the mobile handset.
5. The voice commands should be trained to the EasyVR module.
6. Then the stored voice commands are represented in the form of binary numbers such as move forward – 001, move backward – 010 etc.
7. These binary values are transmitted via WIFI module which is a transceiver.
8. The transmitted binary values are then received by another zig bee module which is present on the receiver side.
9. Microcontroller will take those binary values and performs action (servo motors) according to the binary values.
10. If it is failed to connect at step 3 than again go to step 2.
11. Stop.

# SOURCE CODE:

#include <ESP8266WiFi.h> WiFiClient client; WiFiServer server(80);

const char\* ssid = "Galaxy A54 5g"; const char\* password = "Sreyas123";

String command =""; // Command received from Android device

// Define pin numbers for motor driver inputs and enable pins const int in1Pin = 5; // Connect to IN1 on the motor driver const int in2Pin = 4; // Connect to IN2 on the motor driver const int in3Pin = 0; // Connect to IN3 on the motor driver const int in4Pin = 2; // Connect to IN4 on the motor driver

const int enableAPin = 16; // Connect to ENA on the motor driver (D0) const int enableBPin = 14; // Connect to ENB on the motor driver (D5)

const int defaultSpeed = 100; // Default defaultSpeed value (adjust as needed)

void setup() { Serial.begin(115200);

// Initialize motor control pins pinMode(in1Pin, OUTPUT); pinMode(in2Pin, OUTPUT); pinMode(in3Pin, OUTPUT); pinMode(in4Pin, OUTPUT);

// Initialize enable pins pinMode(enableAPin, OUTPUT);

pinMode(enableBPin, OUTPUT);

// Set enable pins initially to LOW to stop the motors digitalWrite(enableAPin, LOW); digitalWrite(enableBPin, LOW); digitalWrite(in1Pin,LOW); digitalWrite(in2Pin,LOW); digitalWrite(in3Pin,LOW); digitalWrite(in4Pin,LOW);

connectWiFi(); server.begin();

}

void loop() {

client = server.available(); if (!client) return;

command = checkClient ();

if (command == "forward") moveForward(defaultSpeed);

else if (command == "reverse") moveBackward(defaultSpeed); else if (command == "left") moveLeft(defaultSpeed);

else if (command == "right") moveRight(defaultSpeed); else if (command == "stop") stopMotor();

sendBackEcho(command); // send command echo back to android device command = "";

}

void moveForward(int defaultSpeed) { digitalWrite(in1Pin, HIGH); digitalWrite(in2Pin, LOW); digitalWrite(in3Pin, HIGH); digitalWrite(in4Pin, LOW);

// Enable both motors with PWM defaultSpeed control

analogWrite(enableAPin, defaultSpeed); // PWM defaultSpeed control for motor A

analogWrite(enableBPin, defaultSpeed); // PWM defaultSpeed control for motor B

}

void moveBackward(int defaultSpeed) { digitalWrite(in1Pin, LOW); digitalWrite(in2Pin, HIGH); digitalWrite(in3Pin, LOW); digitalWrite(in4Pin, HIGH);

// Enable both motors with PWM defaultSpeed control

analogWrite(enableAPin, defaultSpeed); // PWM defaultSpeed control for motor A

analogWrite(enableBPin, defaultSpeed); // PWM defaultSpeed control for motor B

}

void moveLeft(int defaultSpeed) { digitalWrite(in1Pin, LOW); digitalWrite(in2Pin, HIGH); digitalWrite(in3Pin, HIGH); digitalWrite(in4Pin, LOW);

// Enable both motors with PWM defaultSpeed control

analogWrite(enableAPin, defaultSpeed); // PWM defaultSpeed control for motor A

analogWrite(enableBPin, defaultSpeed); // PWM defaultSpeed control for motor B

}

void moveRight(int defaultSpeed) { digitalWrite(in1Pin, HIGH); digitalWrite(in2Pin, LOW); digitalWrite(in3Pin, LOW); digitalWrite(in4Pin, HIGH);

// Enable both motors with PWM defaultSpeed control

analogWrite(enableAPin, defaultSpeed); // PWM defaultSpeed control for motor A

analogWrite(enableBPin, defaultSpeed); // PWM defaultSpeed control for motor B

}

void stopMotor(void)

{

digitalWrite(enableAPin,LOW); digitalWrite(enableBPin,LOW);

digitalWrite(in1Pin,LOW); digitalWrite(in2Pin,LOW); digitalWrite(in3Pin,LOW); digitalWrite(in4Pin,LOW);

}

/\* connecting WiFi \*/ void connectWiFi()

{

Serial.println("Connecting to WIFI"); WiFi.begin(ssid, password);

while ((!(WiFi.status() == WL\_CONNECTED)))

{

delay(300); Serial.print("..");

}

Serial.println(""); Serial.println("WiFi connected");

Serial.println("NodeMCU Local IP is : "); Serial.print((WiFi.localIP()));

}

/\* check command received from Android Device \*/ String checkClient (void)

{

while(!client.available()) delay(1);

String request = client.readStringUntil('\r'); request.remove(0, 5); request.remove(request.length()-9,9); Serial.println(request);

return request;

}

/\* send command echo back to android device \*/ void sendBackEcho(String echo)

{

client.println("HTTP/1.1 200 OK"); client.println("Content-Type: text/html"); client.println(""); client.println("<!DOCTYPE HTML>"); client.println("<html>"); client.println(echo); client.println("</html>");

client.stop(); delay(1);

}

# CODE SUMMARY:

The provided Arduino sketch enables remote control of a motor driver using an WEMOS D1 Mini microcontroller over WiFi. The code establishes a server that listens for commands from a client device, such as an Android device. Upon receiving a command, the WEMOS D1 Mini executes the corresponding motor movement, including forward, backward, left, right, or stopping the motors.

Key Components:

* 1. WiFi Configuration: Sets up the WEMOS D1 Mini to connect to a WiFi network.
  2. Motor Control: Defines functions to control the direction and speed of the motors connected to the motor driver.
  3. Server Setup: Initializes a server to listen for incoming client connections and commands.
  4. Command Processing: Parses incoming commands from the client and triggers the appropriate motor movement.
  5. Communication: Sends acknowledgment back to the client device with the received command. Overall, this code facilitates wireless control of

motors, providing a foundation for building remote-controlled robotic or automation systems.

# APPLICATION SOFTWARE:



# CONCLUSION:

# In conclusion, the WiFi and Voice-Controlled Robot project represents a pioneering endeavor in the field of robotics, leveraging advanced technologies to create a versatile, interactive, and user-friendly robotic system. Through the seamless integration of hardware components, software development, voice recognition, and wireless connectivity, the project embodies innovation and ingenuity in every aspect of its design and implementation.

# The culmination of extensive research, development, and testing efforts has resulted in a robust and scalable system capable of fulfilling diverse applications across home automation, education, and entertainment domains. By empowering users to control the robot remotely via voice commands and interact with it intuitively through a user-friendly interface, the project seeks to redefine the paradigms of human-robot interaction and elevate the user experience to new heights.

# Looking ahead, the WiFi and Voice-Controlled Robot project sets the stage for further advancements and applications in the field of robotics, paving the way for smarter, more responsive, and interconnected robotic systems that seamlessly integrate into our daily lives. As technology continues to evolve and innovate, the project stands as a testament to the limitless potential of human ingenuity in shaping the future of robotics and automation.

# RESULT:

* Thus, this project helps us to change human-robot interaction by effectively combining Wemos D1 mini and the Voice software.
* The expected outcome of the methodology includes accurate robot motions, smooth wireless communication, and effective interpretation of speech commands.

**PROJECT VIDEO:**

https://drive.google.com/file/d/1RbgRaXb1M56Ob5iJsppeOL6mNaW4e4dH/view?usp=sharing

# REFRENCES:

1. [https://www.instructables.com/Wi-Fi-Voice-Controlled-Robot-Using-](https://www.instructables.com/Wi-Fi-Voice-Controlled-Robot-Using-Wemos-D1-ESP826/) [Wemos-D1-ESP826/](https://www.instructables.com/Wi-Fi-Voice-Controlled-Robot-Using-Wemos-D1-ESP826/)
2. [https://how2electronics.com/wifi-controlled-robot-using-esp8266-](https://how2electronics.com/wifi-controlled-robot-using-esp8266-android-app/) [android-app/](https://how2electronics.com/wifi-controlled-robot-using-esp8266-android-app/)
3. [https://www.pinterest.com/pin/wifi-voice-controlled-robot-using-wemos-](https://www.pinterest.com/pin/wifi-voice-controlled-robot-using-wemos-d1-esp8266-arduino-ide-ifttt-adafruitio-and-google-assistant--624241198327515469/) [d1-esp8266-arduino-ide-ifttt-adafruitio-and-google-assistant--](https://www.pinterest.com/pin/wifi-voice-controlled-robot-using-wemos-d1-esp8266-arduino-ide-ifttt-adafruitio-and-google-assistant--624241198327515469/) [624241198327515469/](https://www.pinterest.com/pin/wifi-voice-controlled-robot-using-wemos-d1-esp8266-arduino-ide-ifttt-adafruitio-and-google-assistant--624241198327515469/)