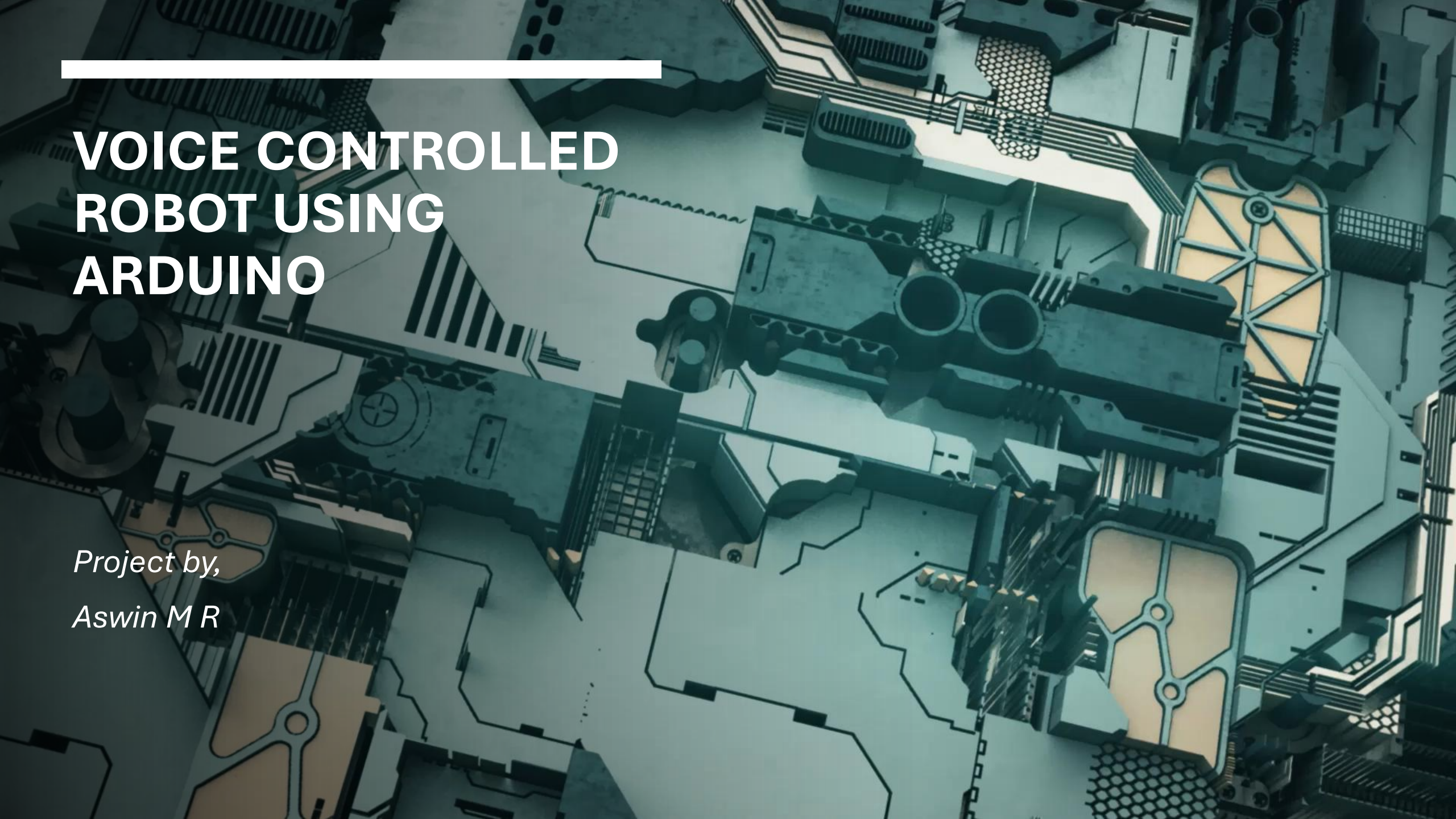


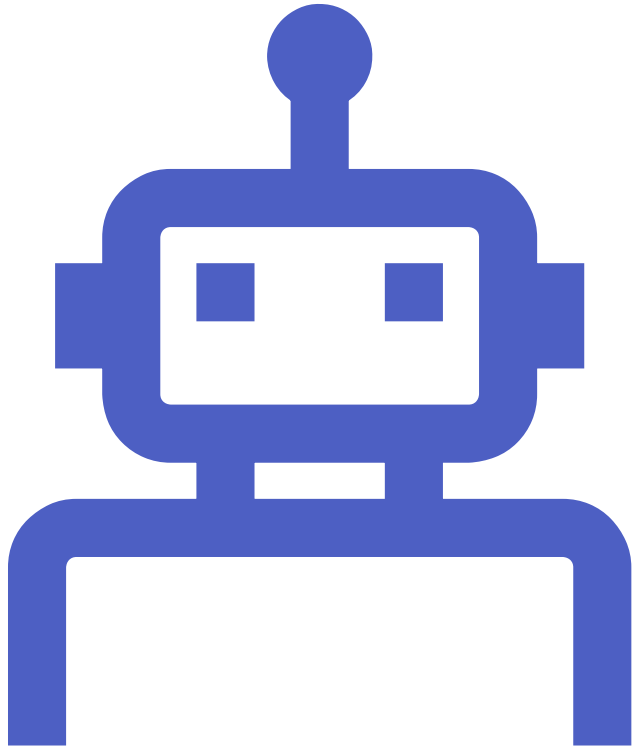
VOICE CONTROLLED ROBOT USING ARDUINO

Project by,

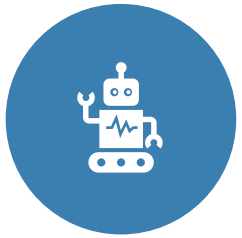
Aswin M R

Introduction

The project involves the development of a voice-controlled robot using Internet of Things (IoT) principles and hardware components such as Arduino Uno, microphone, DC motors, L298N motor driver board, and a battery. The goal is to create a robot that interprets voice commands and translates them into movement commands executed by motors, all controlled through an Arduino platform, with potential IoT connectivity for remote command and monitoring.



Objectives



DEVELOP A ROBOT CAPABLE OF RECEIVING AND PROCESSING VOICE COMMANDS.

USE ARDUINO UNO AS THE CENTRAL CONTROLLER FOR READING MICROPHONE INPUT, PROCESSING COMMANDS, AND CONTROLLING MOTORS.

INTERFACE WITH A MICROPHONE TO CAPTURE VOICE DATA.

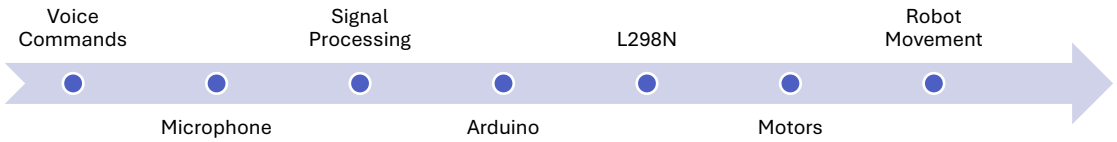
USE AN L298N MOTOR DRIVER BOARD TO CONTROL DC MOTORS BASED ON VOICE COMMANDS.

IMPLEMENT A WIRELESS OR IOT FEATURE FOR REMOTE CONTROL AND MONITORING (OPTIONAL/ADVANCED).

Hardware Components

- **Arduino Uno**: Microcontroller board to process inputs, run control algorithms, and send signals to the motor driver.
- **Microphone Module**: To capture audio commands.
- **L298N Motor Driver Board**: To manage speed and direction of DC motors based on Arduino signals.
- **DC Motors (4)**: For robot locomotion.
- **Battery**: Power source for the robot (suitable voltage and current rating).
- **Additional Components**: Wires, breadboard (or PCB), resistors, voltage regulator and switches.

System Architecture



The system architecture includes:

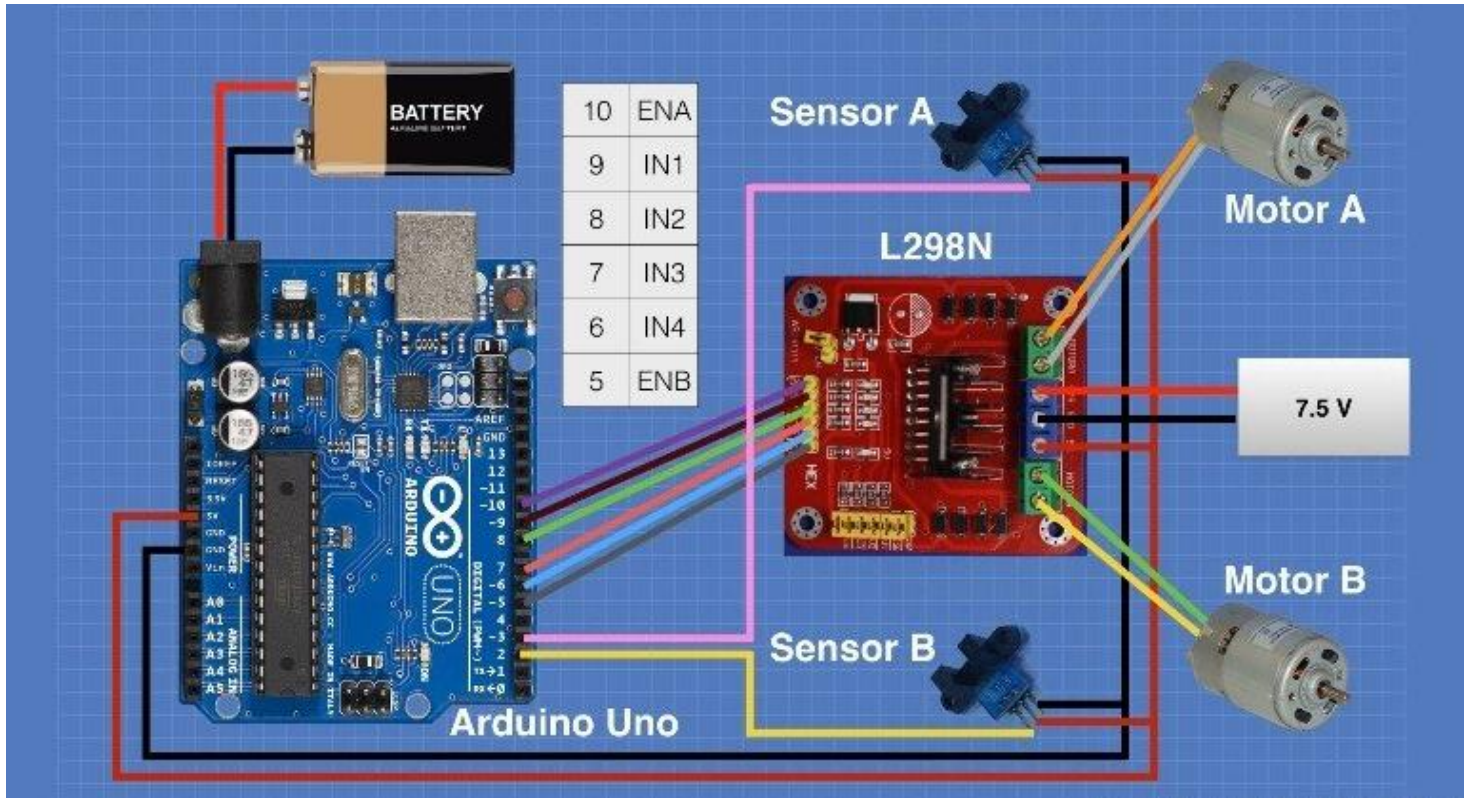
Input Layer: Microphone captures sound which is processed by a voice recognition module or by the Arduino (if speech recognition is implemented on the microcontroller).

Processing Unit: Arduino Uno interprets commands and sends control signals.

Output Layer: L298N motor driver receives signals from Arduino to control motor actions.

Power Supply: Battery powers the entire robot system.

Implementation



Circuit Design and Connections

- **Microphone Connection**: Connected the microphone module output to an analog/digital input on the Arduino (depending on module type).
- **Arduino to L298N**:
  - Connected Arduino digital output pins to the L298N input pins (IN1, IN2, IN3, IN4) to control two motors.
  - Connected ENA and ENB on the L298N to PWM capable pins on Arduino for speed control.
- **Motor and Power Connections**:
  - Connected the motors to the L298N output terminals.
  - Connected the battery positive to the L298N power input and the Arduino Vin or barrel jack, ensuring proper voltage regulation.
  - Common ground connections among Arduino, L298N, and battery.

Code

```
#include <SoftwareSerial.h>
SoftwareSerial BT(0, 1);
String readvoice;
const byte MOTOR_A = 3;
const byte MOTOR_B = 2;
const int mspeed = 200;
const float stepcount = 20.00;
const float wheeldiameter = 66.10;
volatile int counter_A = 0;
volatile int counter_B = 0;
// Motor A
int enA = 10;
int in1 = 9;
int in2 = 8;
// Motor B
int enB = 5;
int in3 = 7;
int in4 = 6;
void ISR_countA()
{
 counter_A++;
}
void ISR_countB()
{
 counter_B++;
}
int CMtoSteps(float cm)
{
 int result;
 float circumference =
(wheeldiameter * 3.14) / 10;
 float cm_step = circumference /
stepcount;
 float f_result = cm / cm_step;
 result = (int) f_result;
 return result;
}
void MoveForward(int steps)
{
  counter_A = 0;
  counter_B = 0;
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  while (steps > counter_A && steps >
counter_B)
  {
   if (steps > counter_A)
   {
   analogWrite(enA,mspeed);
   }
   else
   {
   analogWrite(enA,0);
   }
   if (steps > counter_B)
   {
   analogWrite(enB,mspeed);
   }
   else
   {
   analogWrite(enB,0);
   }
  }
  analogWrite(enA,0);
  analogWrite(enB,0);
  counter_A = 0;
  counter_B = 0;
}
void MoveReverse(int steps)
{
  counter_A = 0;
  counter_B = 0;
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  while (steps > counter_A && steps >
counter_B)
  {
   if (steps > counter_A)
   {
   analogWrite(enA,mspeed);
   }
   else
   {
   analogWrite(enA,0);
   }
   if (steps > counter_B)
   {
   analogWrite(enB,mspeed);
   }
   else
   {
   analogWrite(enB,0);
   }
  }
  analogWrite(enA,0);
  analogWrite(enB,0);
  counter_A = 0;
  counter_B = 0;
}
void SpinRight(int steps)
{
  counter_A = 0;
  counter_B = 0;
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  while (steps > counter_A && steps >
counter_B)
  {
   if (steps > counter_A)
   {
   analogWrite(enA,mspeed);
   }
   else
   {
   analogWrite(enA,0);
   }
   if (steps > counter_B)
   {
   analogWrite(enB,mspeed);
   }
   else
   {
   analogWrite(enB,0);
   }
  }
  analogWrite(enA,0);
  analogWrite(enB,0);
  counter_A = 0;
  counter_B = 0;
}
void SpinLeft(int steps)
{
  counter_A = 0;
  counter_B = 0;
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  while (steps > counter_A && steps >
counter_B)
  {
   if (steps > counter_A)
   {
   analogWrite(enA,mspeed);
   }
   else
   {
   analogWrite(enA,0);
   }
   if (steps > counter_B)
   {
   analogWrite(enB,mspeed);
   }
   else
   {
   analogWrite(enB,0);
   }
  }
  analogWrite(enA,0);
  analogWrite(enB,0);
  counter_A = 0;
  counter_B = 0;
}
void setup()
{
  BT.begin(9600);
  Serial.begin(9600);
  attachInterrupt(digitalPinToInterrupt
(MOTOR_A), ISR_countA, RISING);
  attachInterrupt(digitalPinToInterrupt
(MOTOR_B), ISR_countB, RISING);
}
void loop()
{
  while (BT.available())
  {
  delay(10);
  String c = BT.read();
  char x = ' ';
  int b = c.indexOf(x);
  String cm1 = c.substring(0,b);
  String side = c.substring(b+1);
  float cm2 = cm1.atof;
  Serial.println(cm2);
  int y = CMtoSteps(cm2);
  Serial.println(side);
  readvoice += side;
  if (readvoice.length() > 0)
  {
   Serial.println(readvoice);
  if(readvoice == "forward")
  {
   MoveForward(y);
   delay(100);
  }
  else if(readvoice == "back")
  {
   MoveReverse(y);
   delay(100);
  }
  else if (readvoice == "left")
  {
   SpinLeft(y);
   delay(100);
  }
 else if ( readvoice == "right")
 {
  SpinRight(y);
  delay(100);
 }
 }
}
```

Conclusion

The project successfully demonstrates a voice-controlled robot using Arduino Uno, microphone, L298N motor driver, DC motors, and a battery. Voice commands interpreted by the system effectively control the robot’s motion, showcasing the integration of voice recognition and IoT hardware. While basic functionality has been achieved, refining voice recognition accuracy and enhancing IoT capabilities could further improve the system.

Thank you