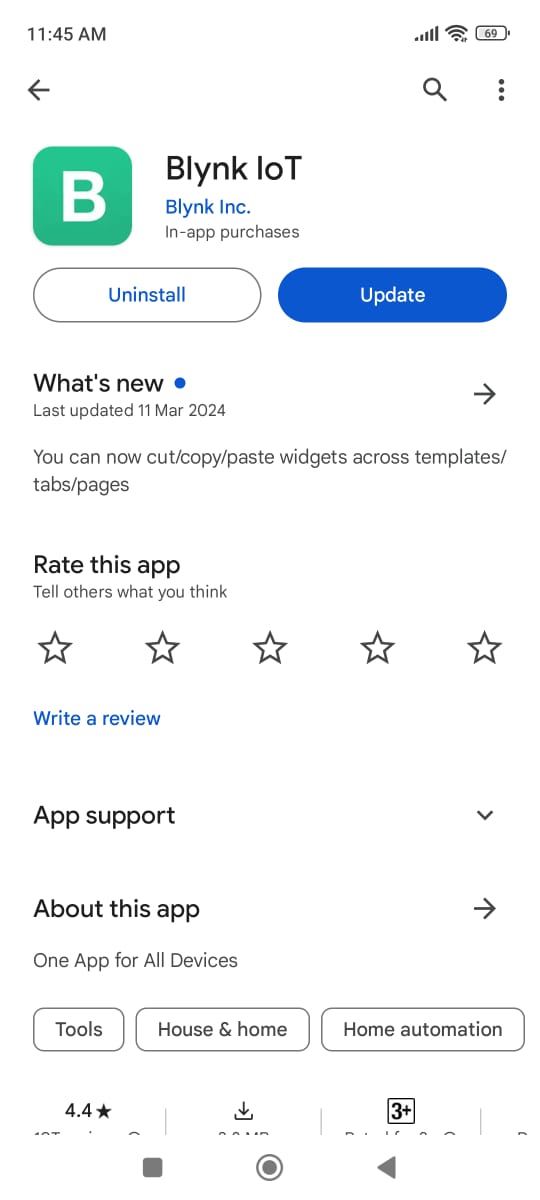
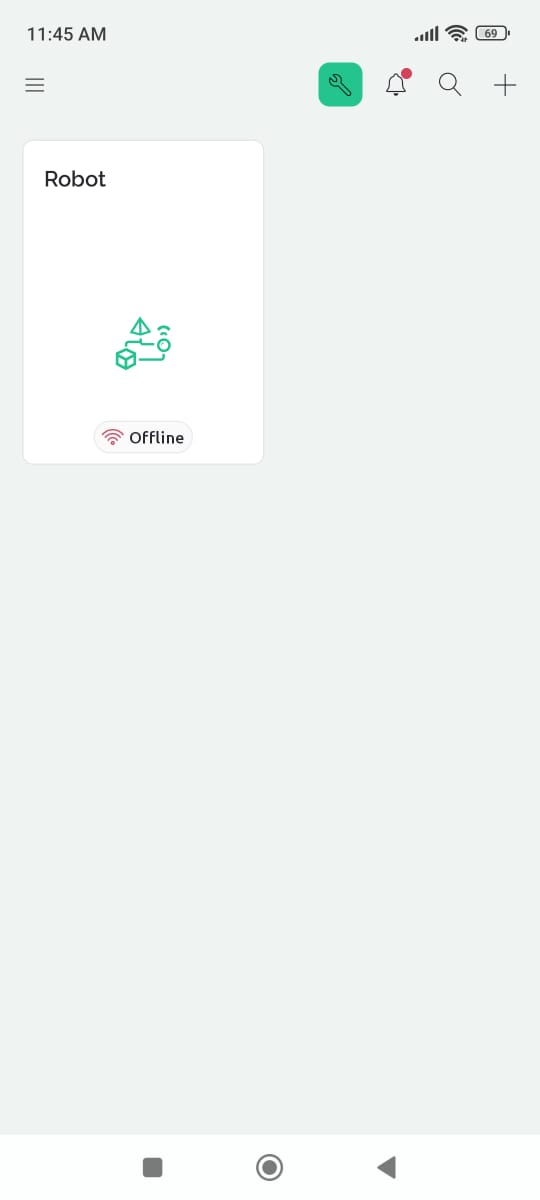
Robot car using Node MCU and Blynk IoT

Blynk IoT app

Components Required:

1. NodeMCU esp8266

2. L293D Motor Driver

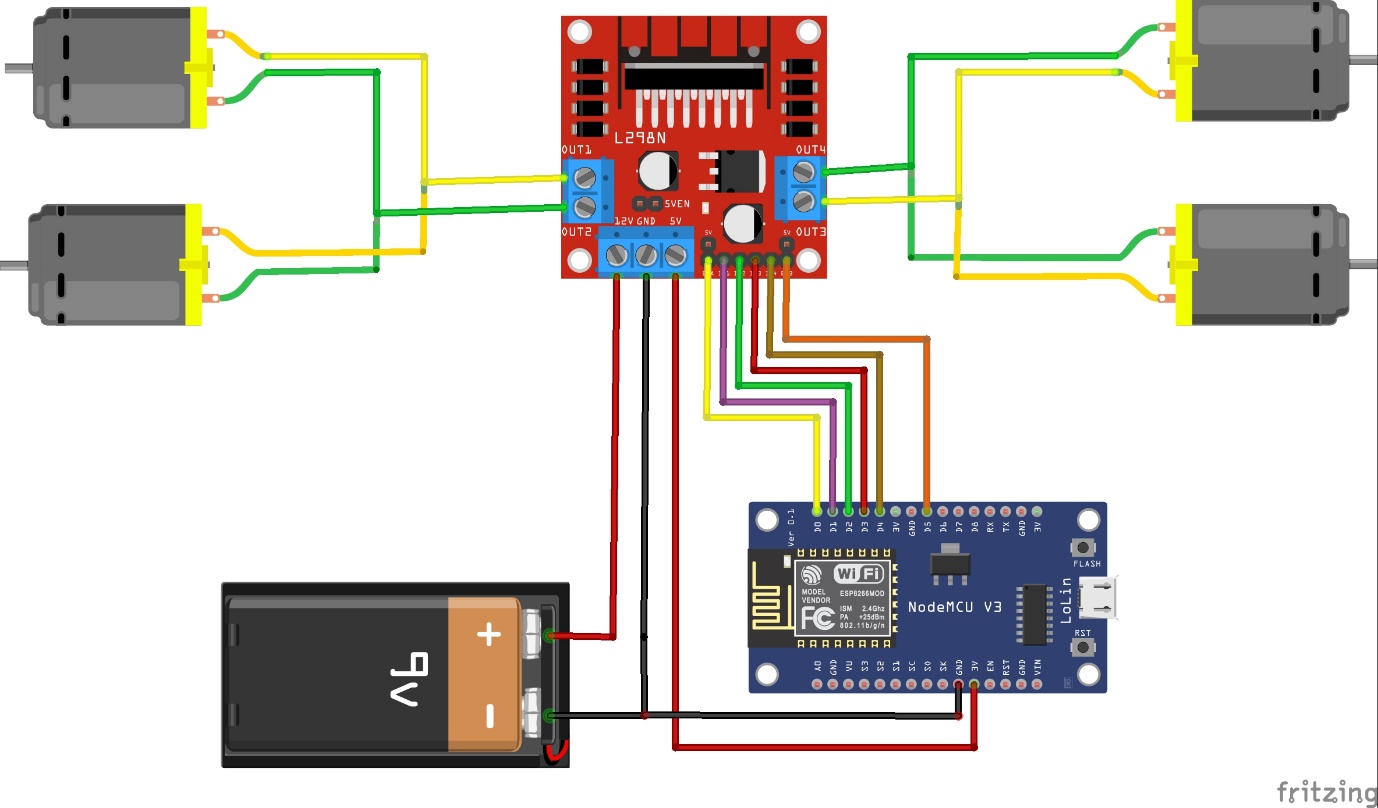
3. Bread Board

4. Jumper Wires

5. 12 Volt Battery   
6. Gear Motors (x4) & Wheels(x4) & [Nuts & bolts]

8. BO Motor Clamps(x4)

Circuit Diagram



Code:

/\*Nodemcu ESP8266 WIFI control car.

 \* This code created by sritu hobby team.

 \* https://srituhobby.com

 \*

/

#define BLYNK\_PRINT Serial

#define BLYNK\_TEMPLATE\_ID "TMPL3Oi1Vji5l"

#define BLYNK\_TEMPLATE\_NAME "Robot"

#define BLYNK\_AUTH\_TOKEN "6cQA1r8pZmX\_sMflUXmuhMMhqI-6wbbF"

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

//Motor PINs

#define ENA D0

#define IN1 D1

#define IN2 D2

#define IN3 D3

#define IN4 D4

#define ENB D5

bool forward = 0;

bool backward = 0;

bool left = 0;

bool right = 0;

int Speed;

char auth[] = "6cQA1r8pZmX\_sMflUXmuhMMhqI-6wbbF"; //Enter your Blynk application auth token

char ssid[] = "imein"; //Enter your WIFI name

char pass[] = "123456789"; //Enter your WIFI passowrd

void setup() {

  Serial.begin(9600);

  pinMode(ENA, OUTPUT);

  pinMode(IN1, OUTPUT);

  pinMode(IN2, OUTPUT);

  pinMode(IN3, OUTPUT);

  pinMode(IN4, OUTPUT);

  pinMode(ENB, OUTPUT);

  Blynk.begin(auth, ssid, pass);

}

BLYNK\_WRITE(V0) {

  forward = param.asInt();

}

BLYNK\_WRITE(V1) {

  backward = param.asInt();

}

BLYNK\_WRITE(V2) {

  left = param.asInt();

}

BLYNK\_WRITE(V3) {

  right = param.asInt();

}

BLYNK\_WRITE(V4) {

  Speed = param.asInt();

}

void smartcar() {

  if (forward == 1) {

    carforward();

    Serial.println("carforward");

  } else if (backward == 1) {

    carbackward();

    Serial.println("carbackward");

  } else if (left == 1) {

    carturnleft();

    Serial.println("carfleft");

  } else if (right == 1) {

    carturnright();

    Serial.println("carright");

  } else if (forward == 0 && backward == 0 && left == 0 && right == 0) {

    carStop();

    Serial.println("carstop");

  }

}

void loop() {

  Blynk.run();

  smartcar();

}

void carforward() {

  analogWrite(ENA, Speed);

  analogWrite(ENB, Speed);

  digitalWrite(IN1, LOW);

  digitalWrite(IN2, HIGH);

  digitalWrite(IN3, HIGH);

  digitalWrite(IN4, LOW);

}

void carbackward() {

  analogWrite(ENA, Speed);

  analogWrite(ENB, Speed);

  digitalWrite(IN1, HIGH);

  digitalWrite(IN2, LOW);

  digitalWrite(IN3, LOW);

  digitalWrite(IN4, HIGH);

}

void carturnleft() {

  analogWrite(ENA, Speed);

  analogWrite(ENB, Speed);

  digitalWrite(IN1, HIGH);

  digitalWrite(IN2, LOW);

  digitalWrite(IN3, HIGH);

  digitalWrite(IN4, LOW);

}

void carturnright() {

  analogWrite(ENA, Speed);

  analogWrite(ENB, Speed);

  digitalWrite(IN1, LOW);

  digitalWrite(IN2, HIGH);

  digitalWrite(IN3, LOW);

  digitalWrite(IN4, HIGH);

}

void carStop() {

  digitalWrite(IN1, LOW);

  digitalWrite(IN2, LOW);

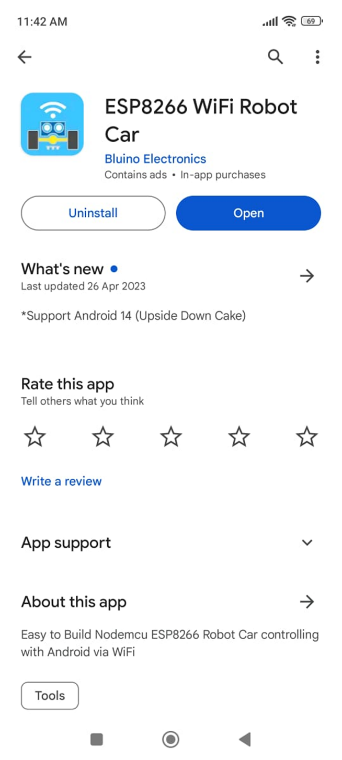
  digitalWrite(IN3, LOW);

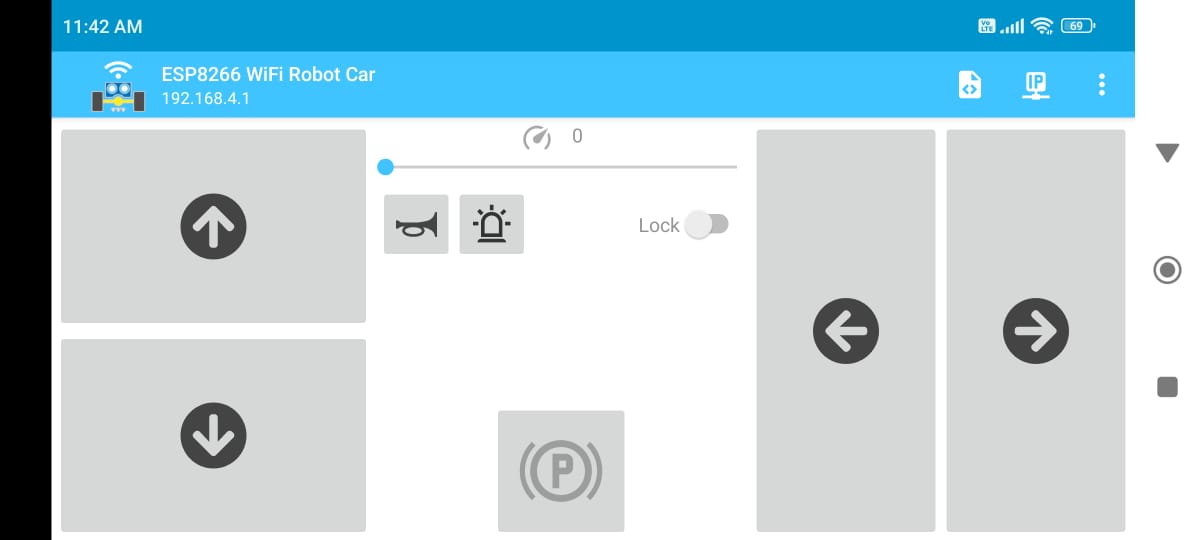
  digitalWrite(IN4, LOW);

}

Robot car using Node MCU and ESP8266 wifi robot app

ESP8266 Wifi robot app





remote control car schematic 


#define ENA   14          // Enable/speed motors Right        GPIO14(D5)

#define ENB   12          // Enable/speed motors Left         GPIO12(D6)

#define IN\_1  15          // L298N in1 motors Rightx          GPIO15(D8)

#define IN\_2  13          // L298N in2 motors Right           GPIO13(D7)

#define IN\_3  2           // L298N in3 motors Left            GPIO2(D4)

#define IN\_4  0           // L298N in4 motors Left            GPIO0(D3)

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <ESP8266WebServer.h>

String command;             //String to store app command state.

int speedCar = 800;         // 400 - 1023.

int speed\_Coeff = 3;

const char\* ssid = "Make DIY";

ESP8266WebServer server(80);

void setup() {

 pinMode(ENA, OUTPUT);

 pinMode(ENB, OUTPUT);

 pinMode(IN\_1, OUTPUT);

 pinMode(IN\_2, OUTPUT);

 pinMode(IN\_3, OUTPUT);

 pinMode(IN\_4, OUTPUT);

  Serial.begin(115200);

// Connecting WiFi

  WiFi.mode(WIFI\_AP);

  WiFi.softAP(ssid);

  IPAddress myIP = WiFi.softAPIP();

  Serial.print("AP IP address: ");

  Serial.println(myIP);

 // Starting WEB-server

     server.on ( "/", HTTP\_handleRoot );

     server.onNotFound ( HTTP\_handleRoot );

     server.begin();

}

void goAhead(){

      digitalWrite(IN\_1, LOW);

      digitalWrite(IN\_2, HIGH);

      analogWrite(ENA, speedCar);

      digitalWrite(IN\_3, LOW);

      digitalWrite(IN\_4, HIGH);

      analogWrite(ENB, speedCar);

  }

void goBack(){

      digitalWrite(IN\_1, HIGH);

      digitalWrite(IN\_2, LOW);

      analogWrite(ENA, speedCar);

      digitalWrite(IN\_3, HIGH);

      digitalWrite(IN\_4, LOW);

      analogWrite(ENB, speedCar);

  }

void goRight(){

      digitalWrite(IN\_1, HIGH);

      digitalWrite(IN\_2, LOW);

      analogWrite(ENA, speedCar);

      digitalWrite(IN\_3, LOW);

      digitalWrite(IN\_4, HIGH);

      analogWrite(ENB, speedCar);

  }

void goLeft(){

      digitalWrite(IN\_1, LOW);

      digitalWrite(IN\_2, HIGH);

      analogWrite(ENA, speedCar);

      digitalWrite(IN\_3, HIGH);

      digitalWrite(IN\_4, LOW);

      analogWrite(ENB, speedCar);

  }

void goAheadRight(){

      digitalWrite(IN\_1, LOW);

      digitalWrite(IN\_2, HIGH);

      analogWrite(ENA, speedCar/speed\_Coeff);

      digitalWrite(IN\_3, LOW);

      digitalWrite(IN\_4, HIGH);

      analogWrite(ENB, speedCar);

   }

void goAheadLeft(){

      digitalWrite(IN\_1, LOW);

      digitalWrite(IN\_2, HIGH);

      analogWrite(ENA, speedCar);

      digitalWrite(IN\_3, LOW);

      digitalWrite(IN\_4, HIGH);

      analogWrite(ENB, speedCar/speed\_Coeff);

  }

void goBackRight(){

      digitalWrite(IN\_1, HIGH);

      digitalWrite(IN\_2, LOW);

      analogWrite(ENA, speedCar/speed\_Coeff);

      digitalWrite(IN\_3, HIGH);

      digitalWrite(IN\_4, LOW);

      analogWrite(ENB, speedCar);

  }

void goBackLeft(){

      digitalWrite(IN\_1, HIGH);

      digitalWrite(IN\_2, LOW);

      analogWrite(ENA, speedCar);

      digitalWrite(IN\_3, HIGH);

      digitalWrite(IN\_4, LOW);

      analogWrite(ENB, speedCar/speed\_Coeff);

  }

void stopRobot(){

      digitalWrite(IN\_1, LOW);

      digitalWrite(IN\_2, LOW);

      analogWrite(ENA, speedCar);

      digitalWrite(IN\_3, LOW);

      digitalWrite(IN\_4, LOW);

      analogWrite(ENB, speedCar);

 }

void loop() {

    server.handleClient();

      command = server.arg("State");

      if (command == "F") goAhead();

      else if (command == "B") goBack();

      else if (command == "L") goLeft();

      else if (command == "R") goRight();

      else if (command == "I") goAheadRight();

      else if (command == "G") goAheadLeft();

      else if (command == "J") goBackRight();

      else if (command == "H") goBackLeft();

      else if (command == "0") speedCar = 400;

      else if (command == "1") speedCar = 470;

      else if (command == "2") speedCar = 540;

      else if (command == "3") speedCar = 610;

      else if (command == "4") speedCar = 680;

      else if (command == "5") speedCar = 750;

      else if (command == "6") speedCar = 820;

      else if (command == "7") speedCar = 890;

      else if (command == "8") speedCar = 960;

      else if (command == "9") speedCar = 1023;

      else if (command == "S") stopRobot();

}

void HTTP\_handleRoot(void) {

if( server.hasArg("State") ){

       Serial.println(server.arg("State"));

  }

  server.send ( 200, "text/html", "" );

  delay(1);

}