WIFI-BOT CAR USING NODEMCU ESP8266

**Nauvyashree A 2018115060 and Harini S 201811035**

**Description:**

 Here we are using ESP8266 [Installed In NodeMCU ] To Communicate With Smartphone And L298N Motor Driver Module To Control Motor. **Nodemcu car** is controlled by using android mobile phone instead of any other method like buttons, gesture, etc. here we only need to touch the button in android phone to control the car in forward, backward, left and right directions. So here android phone is used as transmitting device (STA- station) and [nodemcu esp8266](https://amzn.to/2HgYZkj" \t "_blank) placed in car is used as receiver (AP- access point). android phone will transmit command using its in-built Wi-Fi to car so that it can move in the required direction like moving forward, reverse, turning left, turning right and stop.

**Components Required:**

**HARDWARE REQUIRMENTS** :

**1.NodeMCU ESP8266**: NodeMCU is an open-source firmware and **development board** specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi and hardware which is based on the ESP-12 module.

**2.L298N Motor Driver**: **L298N Motor Driver Module** is a high power motor driver module for driving DC and Stepper Motors. It can control up to 4 DC motors, or 2 DC motors with directional and speed control.

**3. Gear Motor**: BO (Battery Operated) light weight DC geared motor which gives good torque and rpm at lower voltages. This motor can run at approximately 200 rpm when driven by a single Li-Ion cell. Great for battery operated light weight robots. It can do reverse and forward directions.

**4.BO motor wheel:** Wheels to attach to bo motors.

**5.2s Li-po Battery:** Battery to charge the L298N motor driver and nodemcu ESP8266

**6.Jumper Wire:** Wires with a connector or pin at each end used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

**7.Chasis:** Outer plastic body for assembling the parts of the car

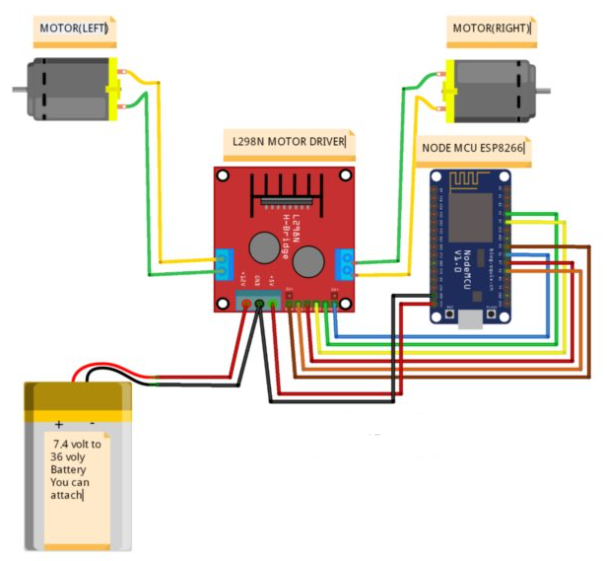
**SOFTWARE REQUIREMENTS :**

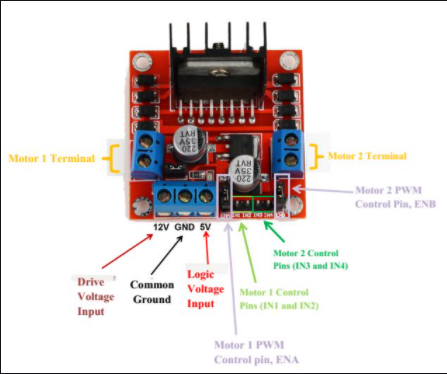
**1.Arduino IDE**: a cross-platform application that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards.

**OUR CAR:**

** **

**ARCHITECTURE AND DIAGRAM :**





**Architecture explanation:**

The left and right motors are connected to the 2 motor terminals of the L298N motor driver.

Volatge pin of the nodemcu is connected to the **Logic voltage input** of L298N motor driver .

Ground pin of nodemcu and the positive terminal of battery are connected to the **common ground** of L298N motor driver.

Negative of battery is connected to the **Driver Volatge Input** of the L298N motor driver.

IN and EN pins are connected appropriately to the Nodemcu ESP8266.

**CODE:**

#define ENA 14 // Enable/speed motors Right

#define ENB 12 // Enable/speed motors Left

#define IN\_1 15 // L298N in1 motors Right

#define IN\_2 13 // L298N in2 motors Right

#define IN\_3 2 // L298N in3 motors Left

#define IN\_4 0 // L298N in4 motors Left

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <ESP8266WebServer.h>

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

int speedCar = 800;

int speed\_Coeff = 3;

const char\* ssid = "NodeMCU Car";

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"));

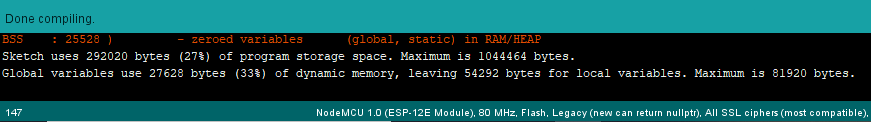
}

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

delay(1);

}

**OUTPUT( IN ARDUINO IDE): After verifying**

****

**THANK YOU!**