

WIFI-IOT-BLUETOOTH FINAL YEAR PROJECT CODE

```
#define BLYNK_PRINT Serial
#include <BlynkSimpleEsp8266.h>
#include <SPI.h>
#include <Ethernet.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

ADC_MODE(ADC_VCC);

#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 Right      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)
char auth[] = "c12617c76f92451f8796f68d94c1ceda";
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid2[] = "Redmi";
char pass[] = "rheagupta";

//String to store app command state.
int speedCar = 800;      // 400 - 1023.
int speed_Coeff = 3;
int val=0;
char command;
String command2;
const char* ssid = "NodeMCU Car";
ESP8266WebServer server(80);
void HTTP_handleRoot(void);
void bluenet();
void setup() {

  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(IN_1, OUTPUT);
  pinMode(IN_2, OUTPUT);
  pinMode(IN_3, OUTPUT);
  pinMode(IN_4, OUTPUT);
  Blynk.begin(auth, ssid2, pass);

  Serial.begin(9600);

  // Connecting WiFi

  // Starting WEB-server
  val=ESP.getVcc();
  if(val<3400)
  {
```

```
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();  
bluenet();  
}  
}
```

```
void goAhead(){
```

```
    digitalWrite(IN_1, LOW);  
    digitalWrite(IN_2, HIGH);  
    analogWrite(ENA, speedCar);
```

```
    digitalWrite(IN_3, LOW);  
    digitalWrite(IN_4, HIGH);  
    analogWrite(ENB, speedCar);  
    Serial.println("forawrd");  
}
```

```
void goBack(){
```

```
    digitalWrite(IN_1, HIGH);  
    digitalWrite(IN_2, LOW);  
    analogWrite(ENA, speedCar);
```

```
    digitalWrite(IN_3, HIGH);  
    digitalWrite(IN_4, LOW);  
    analogWrite(ENB, speedCar);  
    Serial.println("back");  
}
```

```
void goRight(){
```

```
    digitalWrite(IN_1, HIGH);  
    digitalWrite(IN_2, LOW);  
    analogWrite(ENA, speedCar);
```

```
    digitalWrite(IN_3, LOW);  
    digitalWrite(IN_4, HIGH);  
    analogWrite(ENB, speedCar);  
    Serial.println("right");  
}
```

```
void goLeft(){
```

```
digitalWrite(IN_1, LOW);  
digitalWrite(IN_2, HIGH);  
analogWrite(ENA, speedCar);
```

```
digitalWrite(IN_3, HIGH);  
digitalWrite(IN_4, LOW);  
analogWrite(ENB, speedCar);  
Serial.println("left");  
}
```

```
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);  
    Serial.println("forward right");  
}
```

```
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);  
    Serial.println("forward left");  
}
```

```
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);  
    Serial.println("back right");  
}
```

```
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);
    Serial.println("back left");
}

void stopRobot(){

    digitalWrite(IN_1, LOW);
    digitalWrite(IN_2, LOW);
    analogWrite(ENA, speedCar);

    digitalWrite(IN_3, LOW);
    digitalWrite(IN_4, LOW);
    analogWrite(ENB, speedCar);
    Serial.println("stop");
}

void loop()
{
    Blynk.run();
}

void bluenet(){

while(1){
    Serial.print("System voltage(mV): ");Serial.println(ESP.getVcc());
    val=ESP.getVcc();
    if(val<3000){

        while(Serial.available()>0){
            command = Serial.read();
            // Serial.println(command);
        }
        if (command == 'F') goAhead();
        else if (command == 'B') goBack();
        else if (command == 'R') goLeft();
        else if (command == 'L') goRight();
        else if (command == 'I') goAheadRight();
        else if (command == 'G') goAheadLeft();
        else if (command == 'J') goBackRight();
        else if (command == 'H') goBackLeft();
        else if (command == 'S') stopRobot();

    }
    else if(val<3400 && val >3000) {
        server.handleClient();

        command2 = server.arg("State");
        if (command2 == "F") goAhead();
        else if (command2 == "B") goBack();
        else if (command2 == "L") goLeft();
        else if (command2 == "R") goRight();
    }
}
}

```

```

    else if (command2 == "I") goAheadRight();
    else if (command2 == "G") goAheadLeft();
    else if (command2 == "J") goBackRight();
    else if (command2 == "H") goBackLeft();
    else if (command2 == "0") speedCar = 400;
    else if (command2 == "1") speedCar = 470;
    else if (command2 == "2") speedCar = 540;
    else if (command2 == "3") speedCar = 610;
    else if (command2 == "4") speedCar = 680;
    else if (command2 == "5") speedCar = 750;
    else if (command2 == "6") speedCar = 820;
    else if (command2 == "7") speedCar = 890;
    else if (command2 == "8") speedCar = 960;
    else if (command2 == "9") speedCar = 1023;
    else if (command2 == "S") stopRobot();
  }
}
}

```

```

void HTTP_handleRoot(void) {

```

```

  if( server.hasArg("State") ){
    Serial.println(server.arg("State"));
  }
  server.send ( 200, "text/html", "" );
  delay(1);
}

```

```

BLYNK_WRITE(V1){

```

```

  int x = param[0].asInt();
  int y = param[1].asInt();
  if(x==-1 && y==-1){          //Backward and Left
    digitalWrite(IN_1, HIGH);
    digitalWrite(IN_2, LOW);
    digitalWrite(ENA, HIGH);

    digitalWrite(IN_3, HIGH);
    digitalWrite(IN_4, LOW);
    digitalWrite(ENB, HIGH);
    Serial.println("back left");
  }else if(x==-1 && y==0){      //Left Turn
    digitalWrite(IN_1, LOW);
    digitalWrite(IN_2, HIGH);
    digitalWrite(ENA, HIGH);

    digitalWrite(IN_3, HIGH);
    digitalWrite(IN_4, LOW);
    digitalWrite(ENB, HIGH);
    Serial.println("left");
  }else if(x==-1 && y==1){      //Forward and Left

```

```

digitalWrite(IN_1, LOW);
digitalWrite(IN_2, HIGH);
digitalWrite(ENA, HIGH);

digitalWrite(IN_3, LOW);
digitalWrite(IN_4, HIGH);
Serial.println("forward left");
digitalWrite(ENB,HIGH);

}else if(x==0 && y==1){    //Backward
digitalWrite(IN_1, HIGH);
digitalWrite(IN_2, LOW);
digitalWrite(ENA, HIGH);

digitalWrite(IN_3, HIGH);
digitalWrite(IN_4, LOW);
digitalWrite(ENB, HIGH);
Serial.println("back");
}else if(x==0 && y==0){    //Stay
digitalWrite(IN_1, LOW);
digitalWrite(IN_2, LOW);
analogWrite(ENA, HIGH);

digitalWrite(IN_3, LOW);
digitalWrite(IN_4, LOW);
analogWrite(ENB, HIGH);
Serial.println("stop");
}else if(x==0 && y==1){    //Forward
digitalWrite(IN_1, LOW);
digitalWrite(IN_2, HIGH);
digitalWrite(ENA, HIGH);

digitalWrite(IN_3, LOW);
digitalWrite(IN_4, HIGH);
digitalWrite(ENB, HIGH);
Serial.println("forawrd");
}else if(x==1 && y==1){    //Backward and Right
digitalWrite(IN_1, HIGH);
digitalWrite(IN_2, LOW);
digitalWrite(ENA, HIGH);

digitalWrite(IN_3, HIGH);
digitalWrite(IN_4, LOW);
digitalWrite(ENB, HIGH);
Serial.println("back right");
}else if(x==1 && y==0){    //Right turn
digitalWrite(IN_1, HIGH);
digitalWrite(IN_2, LOW);
digitalWrite(ENA, HIGH);

digitalWrite(IN_3, LOW);
digitalWrite(IN_4, HIGH);
digitalWrite(ENB, HIGH);

```

```
Serial.println("right");
}else if(x==1 && y==1){ //Forward and Right
    digitalWrite(IN_1, LOW);
    digitalWrite(IN_2, HIGH);
    digitalWrite(ENA, HIGH);

    digitalWrite(IN_3, LOW);
    digitalWrite(IN_4, HIGH);
    digitalWrite(ENB, HIGH);
    Serial.println("forward right");
}
}
```