How to make a WIFI controlled car using Nodemcu and Blynk app – Step by step instructions

**WIFI controlled car using Nodemcu**

        Hello guys, welcome back. In this tutorial, we are going to talk about how to make a WIFI-controlled car using Nodemcu. [Nodemcu ESP8266](https://srituhobby.com/nodemcu-arduino-ide-setup-and-led-blink" \t "_blank) board and [Blynk application](https://srituhobby.com/blynk-app-setup-up-tutorial-blynk-app-with-nodemcu-esp8266) are mainly used for this. Also, this can be easily controlled by your smartphone. We have described to you step-by-step how to do this through this tutorial.

**Process of this WIFI controlled car**

When this WIFI-controlled car is powered on, the Nodemcu board connects to the Blynk cloud via a WiFi connection. Then, when you press the Commands (Forward, Backward, Left, Right) buttons on the interface created in the Blynk app, those values will be sent to the Nodemcu board via the Blynk cloud. Then, the gear motors rotate according to those values. The [L298N motor driver](https://srituhobby.com/motor-controlling-with-arduino) board is used for this. Also, the speed of these motors can be controlled by the slider created in the Blynk app.

So, let’s make this smart car step by step. The required components are as follows.

* Nodemcu ESP8266 x 1 — [Amazon](https://amzn.to/3hwqG7v) / [Banggood](https://bit.ly/325G7yL" \t "_blank)
* L298N motor driver x 1 — [Amazon](https://amzn.to/3c6vkYM) / [Banggood](https://bit.ly/3fnvhcx" \t "_blank)
* Gear motor x 4 — [Amazon](https://amzn.to/2ZrL0AD) /[Banggood](https://bit.ly/2KlGXkt)
* Robot wheel x 4 — [Amazon](https://amzn.to/3bRmZb6) / [Banggood](https://bit.ly/3feEbsQ" \t "_blank)
* Battery holder x 1 — [Amazon](https://amzn.to/3kd8O3a)
* Li-ion battery  x 2 — [Amazon](https://amzn.to/35wst9W) / [Banggood](https://bit.ly/3tcjcN5" \t "_blank)
* Jumper wires — [Amazon](https://amzn.to/2RjTt4e) / [Banggood](https://bit.ly/3kMWzKr" \t "_blank)
* Breadboard  x 1 — [Amazon](https://amzn.to/35rdMVE) / [Banggood](https://bit.ly/3f4fI9w" \t "_blank)
* Foamboard piece  — [Amazon](https://amzn.to/2Zz2pr2) / [Banggood](https://bit.ly/333UFiC" \t "_blank)

**Step 1**

Firstly, identify these components.

**Nodemcu ESP8266**

**L298N motor driver**

**Gear motor**

**Robot wheel**

**Battery holder**

**Li-ion battery**

**Jumper wires**

**Breadboard**

**Foamboard piece**

**Step 2**

Secondly, glue the four gear motors to the foam board, To do this, use the Glue gun.

**Step 3**

Thirdly, attach the motor driver board to the top of the foam board. Afterward, connect the gear motors to the Motor driver board. For that, use the below circuit diagram.

**Step 4**

OK, now glue the breadboard as follows. Then, attach the Nodemcu board to the breadboard.

**Step 5**

Next, connect the motor driver board to the Nodemcu board. For that, use the above circuit diagram.

**Step 6**

Now, connect the battery holder with the GND and 12v terminals on the motor drive board. After, glue it as follows.

**Step 7**

OK. let’s set up the Blynk application. To do this. download and install this app on your smartphone using the app store or play store. After, follow the steps below.

* First, run the Blynk app on your smartphone. Then, click the “New project” button and enter the project name as you like. After, select the device and connection type. Finally, click the “Confirm” button.
* OK, now we add to the widget interface. To do this, click the “+” button and add four “Styled buttons”. After, arrange them as follows.

* Now, let’s set up these buttons. First, click the up button and enter the button name as you like. After, change the PIN value to “virtual V0”. Change the edge type to “PILL” mode.
* Second, click the down button and enter the button name as you like. After, change the PIN value to “virtual V1”. Change the edge type to “PILL” mode.
* Third, click the left button and enter the button name as you like. After, change the PIN value to “virtual V2”. Change the edge type to “PILL” mode.
* Last, click the right button and enter the button name as you like. After, change the PIN value to “virtual V3”. Change the edge type to “PILL” mode.
* Then, we can see the buttons as follows.
* Then, click again the “+” button and add a slider widget. After, click this widget and named it as you like. Change the PIN to “virtual V4”.

* Lastly, arrange these buttons as follows.

**Step 8**

Next, connect this smart car to the computer. OK. let’s now create the program for this smart car. It is as follows.

First, download and install the libraries below.

WIFI library — [Download](https://drive.google.com/file/d/1Tho8TuUx2Uif2HhkG12OOkEoC-ezN4Ff/view?usp=sharing)

Blynk library — [Download](https://drive.google.com/file/d/1cQK8im_e6PcQ1r130TzrDZEIxxMh82Rg/view?usp=sharing)

**The complete program of this project –**[**Download**](https://drive.google.com/drive/folders/1fxRvP5aALbHxSLGp1NPi_euAP8LJhRXv?usp=sharing)

/\*Nodemcu ESP8266 WIFI control car.

\* This code created by sritu hobby team.

\* http://srituhobby.com

\*/

#define BLYNK\_PRINT Serial

#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[] = ""; //Enter your Blynk application auth token

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

char pass[] = ""; //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);

}

**Code explanation**

First, WIFI and Blynk libraries are included.

**#define BLYNK\_PRINT Serial**

**#include <ESP8266WiFi.h>**

**#include <BlynkSimpleEsp8266.h>**

This code defines the PINs that connect the motor driver board to the Nodemcu board.

**//Motor PINs**

**#define ENA D0**

**#define IN1 D1**

**#define IN2 D2**

**#define IN3 D3**

**#define IN4 D4**

**#define ENB D5**

After, these variables were created to put the blynk application command button values.

**bool forward = 0;**

**bool backward = 0;**

**bool left = 0;**

**bool right = 0;**

**int Speed;**

OK, now we enter the Blynk application “Auth token” and WIFI contact information. For that, use your details.

**char auth[] = “”; //Enter your Blynk application auth token**

**char ssid[] = “”; //Enter your WIFI name**

**char pass[] = “”; //Enter your WIFI passowrd**

In the setup function, the serial monitor and Blynk library begin. Also, motor driver board PINs set as output PINs.

**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);**

**}**

This code gets the values through the Blink application.

**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();**

**}**

This “smartcar” function is the main code of this program. Through this code, the blink application button values are checked using the IF condition. That is,

* If the forward variable value is 1, the smart car moves forward.
* If the backward variable value is 1, the smart car moves backward.
* If the left variable value is 1, the smart car moves left.
* If the right variable value is 1, the smart car moves right.

**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”);**

**}**

**}**

In the loop function, called the “smartcar” function and Blynk library is run.

**void loop() {**

**Blynk.run();**

**smartcar();**

**}**

These functions rotate the gear motor forward, backward, left, and right.

**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);**

**}**

**Step 9**

OK, now select board and port. Afterward, upload this code to the Nodemcu board.

**Step 10**

OK, the Blynk app and program have done. Now, unplug the USB cable and attach the smart car wheels.

**Step 11**

Lastly, attach the batteries to the battery holder and turns ON your smart car. Now, run the Blynk app project we created earlier. OK, enjoy this project. The full video guide is given below. So, we will meet in the next tutorial.

How to[make a WIFI controlled car using Nodemcu and Blynk app – Step by step instructions](https://youtu.be/huWn4Sx99hQ)