

# TECHTUTORIALSX

A site about programming, IoT and technology. Find tutorials for many different projects.

APRIL 2, 2017

## ESP8266: NodeMCU Pin mappings

The objective of this post is to explain how to use the ESP8266 library defined constants that have the correct mapping between NodeMCU and ESP8266 pins.

### Introduction

The objective of this post is to explain how to use the ESP8266 library defined constants that have the correct mapping between NodeMCU and ESP8266 pins.

NodeMCU is a very easy to use ESP8266 board that can be bought at eBay for less than 4 euros. It's also very practical since it already has a USB header, so we can program the microcontroller without any additional

hardware.

Besides that, the pins are easily accessible, allowing us to take full advantage of the capabilities of the ESP8266, as opposed to other simpler boards, such as the ESP-01, which only expose some of the GPIOs of the microcontroller.

Nevertheless, as indicated in some previous tutorials, the numbers of the pins in the board don't map to the numbers of the pins on the ESP8266. So, for example, pin D1 of the board doesn't map to GPIO1 of the ESP8266 (it actually maps to GPIO5).

Naturally, if this is not taken in consideration, it will lead to a difficult debugging process, since we will be assuming that the board is not working correctly.

So, the correct pin mapping is the following [1][2] (NodeMCU on the left and ESP8266 on the right):

- D0 = GPIO16;
- D1 = GPIO5;
- D2 = GPIO4;
- D3 = GPIO0;
- D4 = GPIO2;
- D5 = GPIO14;
- D6 = GPIO12;
- D7 = GPIO13;
- D8 = GPIO15;
- D9 = GPIO3;
- D10 = GPIO1;
- LED\_BUILTIN = GPIO16 (auxiliary constant for the board LED, not a board pin);

Fortunately, this mapping is defined as constant on ESP8266 the libraries, so we don't need to constantly check it. Thus, we can, for example, call a **digitalWrite** on pin D0, which will be translated to the real GPIO pin 16 [1].

You can check the full mappings that can be used for the NodeMCU [here](#). Note that there are also mappings for other ESP8266 boards, which you can check [here](#).

**Important:** Please note that there are lots of generic ESP8266 boards and there is the possibility that some of them are sold under the name of NodeMCU and have different pin mappings. Besides that, there are different NodeMCU versions. Although this mappings have worked for all the NodeMCU boards I've used, please take this in consideration if you are experiencing problems.

## Example code

Just as a very simple example code, we are going to use one of these mapping in the famous [blink](#) example. To avoid the need for external hardware, we are going to use the NodeMCU built in LED, which is connected to pin D0 of the board [3].

So, we are going to use the defined D0 constant to control the LED without the need to worry about the mapping to the actual ESP8266 GPIO pin.

```
1 | void setup() {  
2 |  
3 |     pinMode(D0, OUTPUT); //Declare Pin mode  
4 | }
```

```
5 | }
6 |
7 | void loop() {
8 |
9 |     digitalWrite(D0, HIGH);    //Turn the LED on
10 |    delay(1000);               //Wait 1 second
11 |    digitalWrite(D0, LOW);     //Turn the LED off
12 |    delay(1000);               //Wait 1 second
13 |
14 | }
```

Note that we could also have used the **LED\_BUILTIN** constant, which would map to the same exact result.

As an additional test, you can also try to print these constants to the serial port and confirm that the mappings are correct.

## References

[1] <https://github.com/esp8266/Arduino/issues/584#issuecomment-123715951>

[2] [https://github.com/esp8266/Arduino/blob/3e7b4b8e0cf4e1f7ad48104abfc42723b5e4f9be/variants/nodemcu/pins\\_arduino.h](https://github.com/esp8266/Arduino/blob/3e7b4b8e0cf4e1f7ad48104abfc42723b5e4f9be/variants/nodemcu/pins_arduino.h)

[3] <https://arduining.com/2015/08/20/nodemcu-breathing-led-with-arduino-ide/>

# Technical details

ESP8266 libraries: v2.3.0

---

Share this:



---

Like this:



4 bloggers like this.

## One Reply to “ESP8266: NodeMCU Pin mappings”

**TechnologyTinker**

APRIL 3, 2017 AT 2:56 AM

Reblogged this on [The Technology Tinker](#) and commented:

I had some problems with this when I first used a NodeMCU board. It's simple enough to figure out, but it still can give you a momentary urge to yank out hair!

Thanks for making this easy to see and understand!

Liked by 1 person

## TechnologyTinker

APRIL 3, 2017 AT 2:56 AM

Reblogged this on The Technology Tinker and commented:

I had some problems with this when I first used a NodeMCU board. It's simple enough to figure out, but it still can give you a momentary urge to yank out hair!

Thanks for making this easy to see and understand!

Loading...