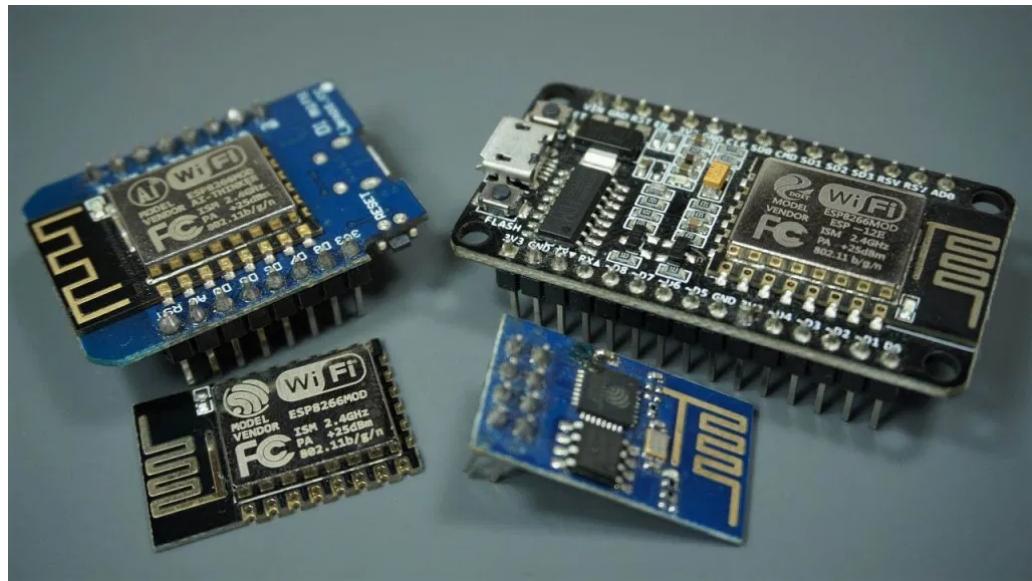


# ESP8266 Pinout Reference: Which GPIO pins should you use?

This article is a guide for the ESP8266 GPIOs: pinout diagrams, their functions and how to use them.



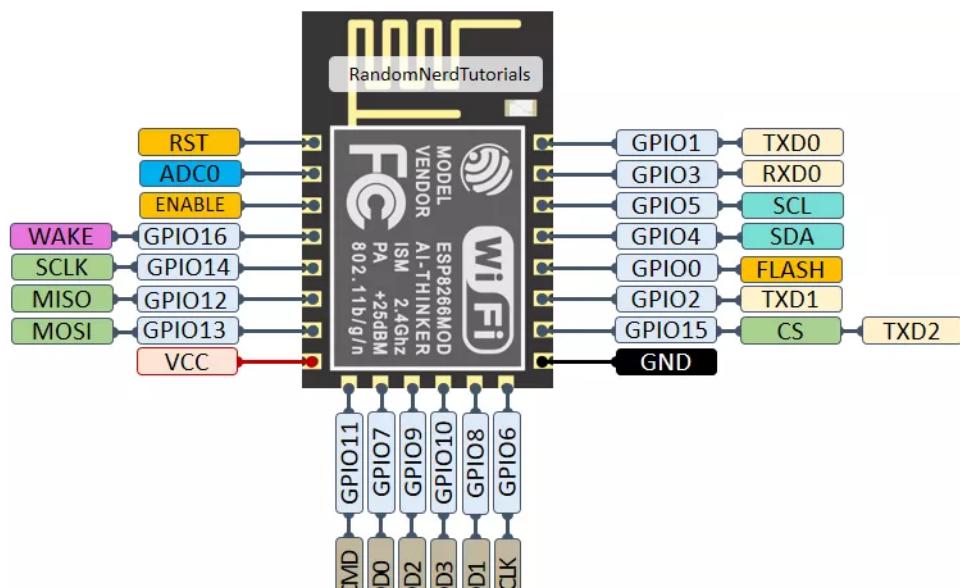
The ESP8266 12-E chip comes with 17 GPIO pins. Not all GPIOs are exposed in all ESP8266 development boards, some GPIOs are not recommended to use, and others have very specific functions.

With this guide, you'll learn how to properly use the ESP8266 GPIOs and avoid hours of frustration by using the most suitable pins for your projects.

We also have a guide for the ESP32 GPIOs: [ESP32 Pinout Reference: Which GPIO pins should you use?](#)

## ESP8266 12-E Chip Pinout

The following figure illustrates the ESP8266 12-E chip pinout. Use this diagram if you're using an ESP8266 bare chip in your projects.



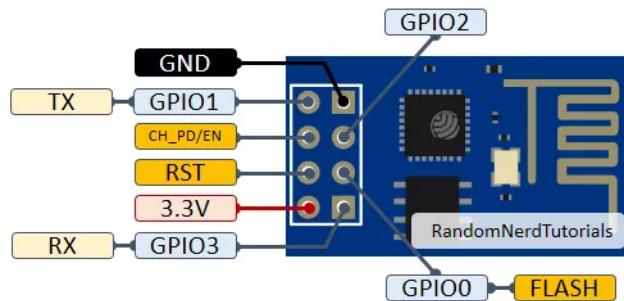
**Note:** not all GPIOs are accessible in all development boards, but each specific GPIO works in the same way regardless of the development board you're using. If you're just getting started with the ESP8266, we recommend reading our guide: Getting Started with the ESP8266.

At the moment, there are a wide variety of development boards with the ESP8266 chip that differ in the number of accessible GPIOs, size, form factor, etc...

The most widely used ESP8266 boards are the [ESP-01](#), [ESP8266-12E NodeMCU Kit](#), and the [Wemos D1 Mini](#). For a comparison of these board, you can read this guide: [ESP8266 Wi-Fi Development Boards comparison](#).

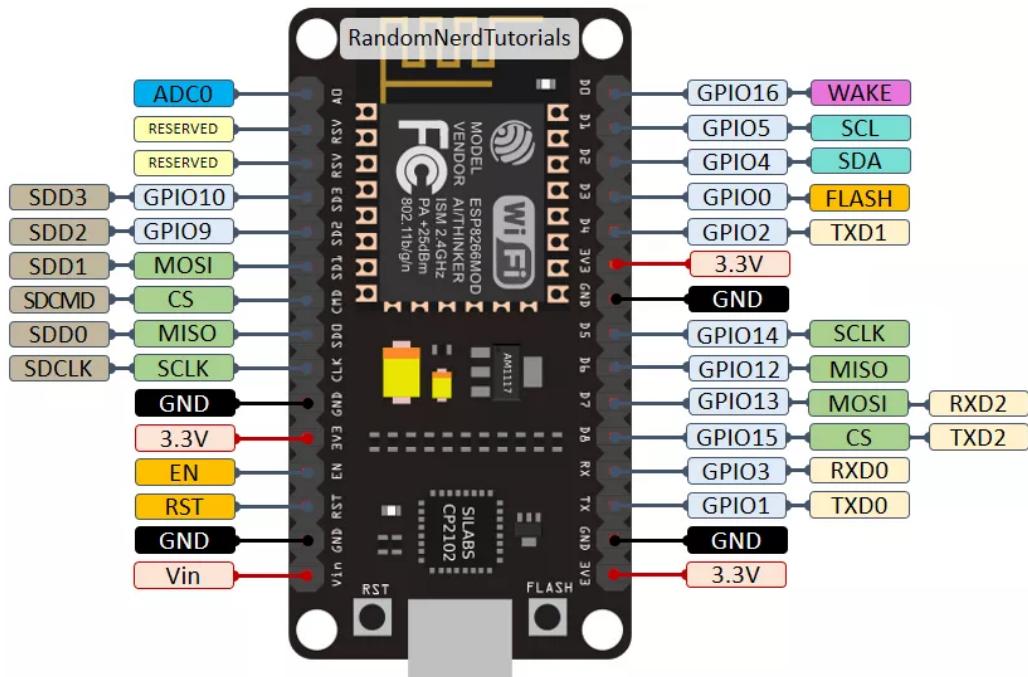
## ESP8266-01 Pinout

If you're using an ESP8266-01 board, you can use the following GPIO diagram as a reference.



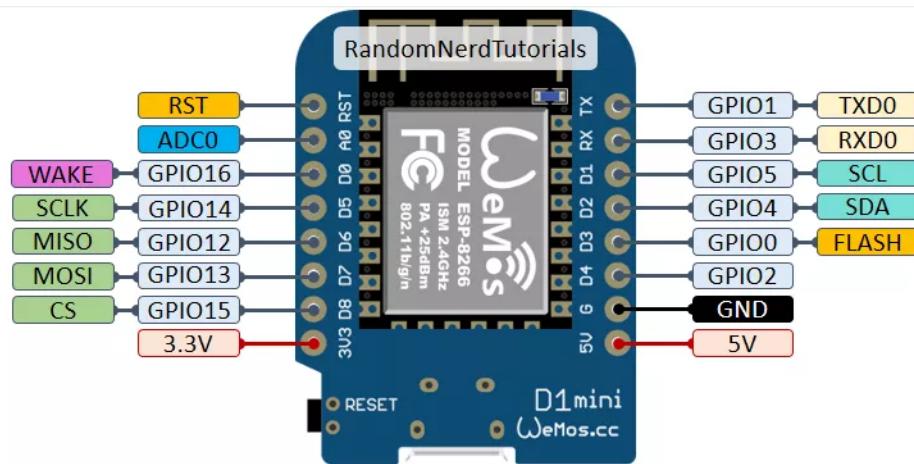
## ESP8266 12-E NodeMCU Kit

The ESP8266 12-E NodeMCU kit pinout diagram is shown below.



## Wemos D1 Mini Pinout

The following figure shows the WeMos D1 Mini pinout.



## Download PDF with ESP8266 Pinout Diagrams

We've put together a handy PDF that you can download and print, so you always have the ESP8266 diagrams next to you:

[Download PDF Pinout Diagrams »](#)

## ESP8266 Peripherals

The ESP8266 peripherals include:

- 17 GPIOs
- SPI
- I2C (implemented on software)
- I2S interfaces with DMA
- UART
- 10-bit ADC

## Best Pins to Use - ESP8266

One important thing to notice about ESP8266 is that the GPIO number doesn't match the label on the board silkscreen. For example, D0 corresponds to GPIO16 and D1 corresponds to GPIO5.

The following table shows the correspondence between the labels on the silkscreen and the GPIO number as well as what pins are the best to use in your projects, and which ones you need to be cautious.

The pins highlighted in green are OK to use. The ones highlighted in yellow are OK to use, but you need to pay attention because they may have unexpected behavior mainly at boot. The pins highlighted in red are not recommended to use as inputs or outputs.

Label	GPIO	Input	Output	Notes
D0	GPIO16	no interrupt	no PWM or I2C support	HIGH at boot used to wake up from deep sleep
D1	GPIO5	OK	OK	often used as SCL (I2C)
D2	GPIO4	OK	OK	often used as SDA (I2C)
D3	GPIO0	pulled up	OK	connected to FLASH button, boot fails if pulled LOW
D4	GPIO2	pulled up	OK	HIGH at boot connected to on-board LED, boot fails if pulled LOW

<b>D6</b>	<b>GPIO12</b>	OK	OK	SPI (MISO)
<b>D7</b>	<b>GPIO13</b>	OK	OK	SPI (MOSI)
<b>D8</b>	<b>GPIO15</b>	pulled to GND	OK	SPI (CS) Boot fails if pulled HIGH
<b>RX</b>	<b>GPIO3</b>	OK	RX pin	HIGH at boot
<b>TX</b>	<b>GPIO1</b>	TX pin	OK	HIGH at boot debug output at boot, boot fails if pulled LOW
<b>A0</b>	<b>ADC0</b>	Analog Input	X	

◀ ▶

Continue reading for a more detailed and in-depth analysis of the ESP8266 GPIOs and its functions.

## GPIOs connected to the Flash Chip

GPIO6 to GPIO11 are usually connected to the flash chip in ESP8266 boards. So, these pins are not recommended to use.

## Pins used during Boot

The ESP8266 can be prevented from booting if some pins are pulled LOW or HIGH. The following list shows the state of the following pins on BOOT:

- **GPIO16:** pin is high at BOOT
- **GPIO0:** boot failure if pulled LOW
- **GPIO2:** pin is high on BOOT, boot failure if pulled LOW
- **GPIO15:** boot failure if pulled HIGH
- **GPIO3:** pin is high at BOOT
- **GPIO1:** pin is high at BOOT, boot failure if pulled LOW
- **GPIO10:** pin is high at BOOT
- **GPIO9:** pin is high at BOOT

## Pins HIGH at Boot

There are certain pins that output a 3.3V signal when the ESP8266 boots. This may be problematic if you have relays or other peripherals connected to those GPIOs. The following GPIOs output a HIGH signal on boot:

- GPIO16
- GPIO3
- GPIO1
- GPIO10
- GPIO9

Additionally, the other GPIOs, except GPIO5 and GPIO4, can output a low-voltage signal at boot, which can be problematic if these are connected to transistors or relays. You can [read this article](#) that investigates the state and behavior of each GPIO on boot.

GPIO4 and GPIO5 are the most safe to use GPIOs if you want to operate relays.

## Analog Input

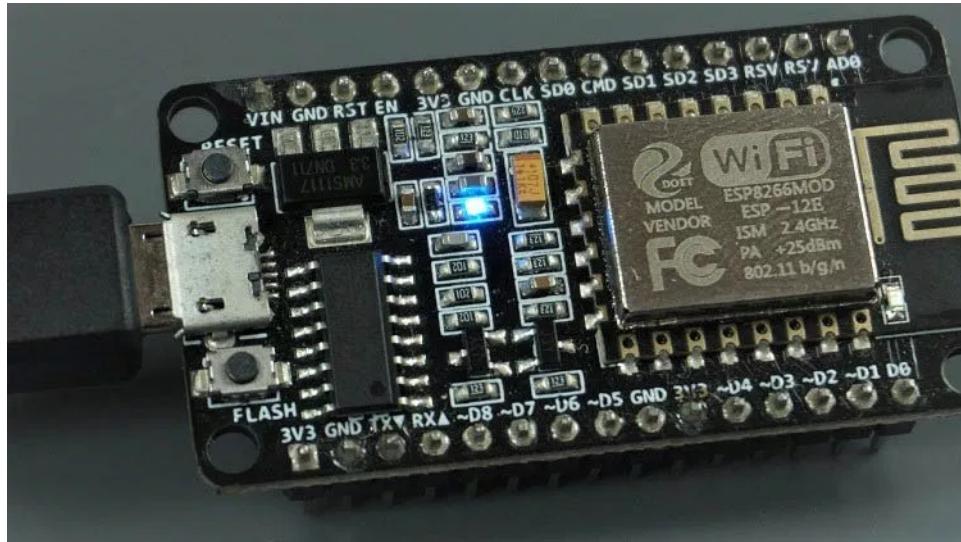
The ESP8266 only supports analog reading in one GPIO. That GPIO is called **ADC0** and it is usually marked on the silkscreen as **A0**.

The maximum input voltage of the ADC0 pin is 0 to 1V if you're using the ESP8266 bare chip. If you're using a development board like the ESP8266 12-E NodeMCU kit, the voltage input range is 0 to 3.3V because these boards contain an internal voltage divider.

- [ESP8266 ADC – Read Analog Values with Arduino IDE, MicroPython and Lua](#)

## On-board LED

Most of the ESP8266 development boards have a built-in LED. This LED is usually connected to GPIO2.



The LED is connected to a pull-down resistor, so when you send a HIGH signal the LED turns off.

## RST Pin

When the RST pin is pulled LOW, the ESP8266 resets. This is the same as pressing the on-board RESET button.



## GPIO0

When GPIO0 is pulled LOW, it sets the ESP8266 into bootloader mode. This is the same as pressing the on-board FLASH/BOOT button.



## GPIO16

GPIO16 can be used to wake up the ESP8266 from deep sleep. To wake up the ESP8266 from deep sleep, GPIO16 should be connected to the RST pin. Learn how to put the ESP8266 into deep sleep mode:

- [ESP8266 Deep Sleep with Arduino IDE](#)
- [ESP8266 Deep Sleep with MicroPython](#)

## I2C

The ESP8266 doesn't have hardware I2C pins, but it can be implemented in software. So you can use any GPIOs as I2C. Usually, the following GPIOs are used as I2C pins:

- **GPIO5:** SCL
- **GPIO4:** SDA

## SPI

The pins used as SPI in the ESP8266 are:

- **GPIO12:** MISO
- **GPIO13:** MOSI
- **GPIO14:** SCLK
- **GPIO15:** CS

## PWM Pins

ESP8266 allows software PWM in all I/O pins: GPIO0 to GPIO16. PWM signals on ESP8266 have 10-bit resolution. Learn how to use ESP8266 PWM pins:

- [ESP8266 PWM with Arduino IDE](#)
- [ESP8266 PWM with MicroPython](#)

## Interrupt Pins

The ESP8266 supports interrupts in any GPIO, except GPIO16.

- [ESP8266 Interrupts and Timers with Arduino IDE](#)

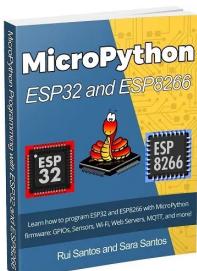
We hope you've found this guide for the ESP8266 GPIOs useful. If you have some tips on how to use the ESP8266 GPIOs properly, you can write a comment below.

We also have a [similar guide for the ESP32 GPIOs that you can read](#).

If you're getting started with the ESP8266, we have some great content you might be interested in:

- [Home Automation using ESP8266 \(course\)](#)
- [Getting Started with the ESP8266](#)
- [30+ ESP8266 Projects and Tutorials](#)
- [ESP8266 Web Server Tutorial](#)
- [ESP32 vs ESP8266 – Pros and Cons](#)

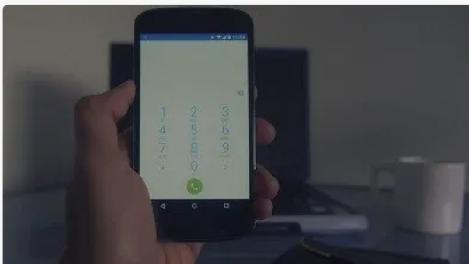
Thanks for reading.



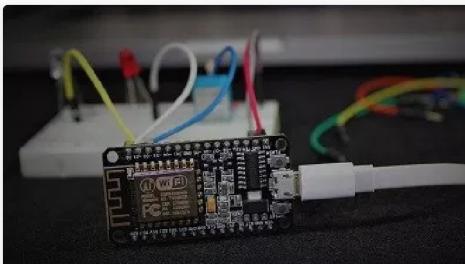
### [eBook] MicroPython Programming with ESP32 and ESP8266

Learn how to program and build projects with the ESP32 and ESP8266 using MicroPython firmware [DOWNLOAD »](#)

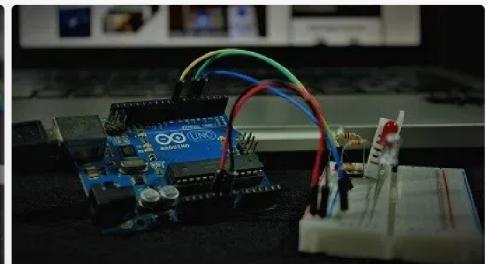
## Recommended Resources



[Build a Home Automation System from Scratch »](#) With Raspberry Pi, ESP8266, Arduino, and Node-RED.



[Home Automation using ESP8266 eBook and video course »](#) Build IoT and home automation projects.



[Arduino Step-by-Step Projects »](#) Build 25 Arduino projects with our course, even with no prior experience!

## What to Read Next...

[ESP-NOW with ESP8266: Receive Data from Multiple Boards \(many-to-one\)](#)[Flashing NodeMCU Firmware on the ESP8266 using Windows](#)[Get ESP32/ESP8266 MAC Address and Change It \(Arduino IDE\)](#)[Getting Started with ESP-NOW \(ESP32 with Arduino IDE\)](#)

Enjoyed this project? Stay updated by subscribing our weekly newsletter!

## 51 thoughts on “ESP8266 Pinout Reference: Which GPIO pins should you use?”

 **Brian**  
May 6, 2019 at 3:53 pm

Excellent all in one place info. Thanks

[Reply](#)

 **Sara Santos**  
May 6, 2019 at 5:48 pm

Thanks 😊

[Free eBooks](#)

[About](#)

[Contact](#)

[Courses Login](#)

[Get Courses](#) ▾



**Andrew Wilson**

August 25, 2019 at 2:12 pm

A really comprehensive guide. I've done a lot of trial and error with these devices. I now have all of the insight to design my projects and possibly a commercial product correctly from scratch. I am much indebted.

[Reply](#)**GurraE**

May 6, 2019 at 3:55 pm

Rui, your site is one of the best there is for learning ESP8266 (& ESP32).  
I've struggled a lot with projects that wont boot etc. and now I understand why !!!  
Your articles are always very professional, very interesting and easy to follow.  
I have found many solutions to problems by studying your articles & projects !!  
Thank you !!

[Reply](#)**Sara Santos**

May 7, 2019 at 9:28 am

Hi.  
Thank you so much for your nice works.  
I'm very happy to see that our readers really enjoy our work. That keeps me motivated to write even better content.  
Regards,  
Sara

[Reply](#)**Michele**

May 6, 2019 at 4:00 pm

Great article, thanks !!  
It's always convenient to have tables showing the pin outs of these fantastic cards.  
I've been trying to flash an ESP-07S with nodemcu firmware lately, but I didn't succeed. Do you have a chart for this card?  
Greetings. Mike.

[Reply](#)**Mike.**

May 6, 2019 at 4:12 pm

Thank you for producing comprehensive guides to ESP8266 devices /boards (also ESP12E), they provide very useful and accessible information sources, without the need for internet searches.

[Reply](#)

**Sara Santos**

May 7, 2019 at 9:21 am

Thank you 😊

[Reply](#)

**Lance Benson**

May 6, 2019 at 4:41 pm

Thanks for this useful guide to the pinouts.

Does the Wemos D1 mini have an onboard voltage divider for the analog input? I ask because I just tried an ACS712 Current Sensor Module, 5V powered, and divided down to the 1V range with a 4K7 & 1K2 resistor divider.

The ACS712 gives a “neutral” reading of half the range, so I was expecting around 512 with nothing connected. Instead I get 170, which would be expected if the A0 input was divided onboard by 3.

(Now what do I need to do to turn a divide by 3 into a divide by 5?)

[Reply](#)

**Dave**

May 6, 2019 at 4:50 pm

This clears up some questions, thanks!

Is there a key or guide for the color tabs used in the pinout diagrams? I understand black and red, but light blue, gray, etc?

Thanks

[Reply](#)

**Sara Santos**

May 7, 2019 at 9:20 am

Hi Dave.

Those are just random colors we've chosen to differentiate the pins.

Pins with similar functions are highlighted with the same color.

For example, SPI is green, I2C is light blue, etc.

[Reply](#)**Transmic**

May 6, 2019 at 6:43 pm

Wonderful, clear and concise article.  
I bough one of your guide and it's written this way.  
Pleasant to read and to use!

[Reply](#)**Sara Santos**

May 7, 2019 at 9:13 am

Thank you so much! 😊

[Reply](#)**Geraldo Cartolano**

May 6, 2019 at 6:51 pm

O trabalho de vcs é perfeito, sempre fico impressionado com a qualidade e precisão de suas postagens.

[Reply](#)**Sara Santos**

May 7, 2019 at 9:12 am

Obrigada 😊

[Reply](#)**Nils Skansar**

May 6, 2019 at 7:04 pm

You are correct, the A0 input is divided. See drawing.  
[escapequotes.net/esp8266-wemos-d1-mini-pins-and-diagram/](http://escapequotes.net/esp8266-wemos-d1-mini-pins-and-diagram/)  
In the middle of the schematic far right you can see the divider.

[Reply](#)

**Nils Skansar**

May 6, 2019 at 7:23 pm

Forgot to mention that you can modify the voltage divider on the ACS712 to a 5 to 3 V divider. Here some math comes in. since you will load 1 voltage divider by another. Some testing needed. Final calibration can be done in software.

[Reply](#)**Roberto Pangallo**

May 6, 2019 at 8:50 pm

Great tutorial! Thank you, Sara and Rui.

Roberto  
Austria

[Reply](#)**Sara Santos**

May 7, 2019 at 9:17 am

Thank you 😊

[Reply](#)**Lance Benson**

May 6, 2019 at 9:13 pm

Re voltage divider on the D1 Mini, the link below says onboard there are 220K and 100K resistors, so to divide down 5 volts, you add 180K in series to A0.

[letscontrolit.com/wiki/index.php/DC\\_Voltage\\_divider](https://letscontrolit.com/wiki/index.php/DC_Voltage_divider)

[Reply](#)**Sara Santos**

May 7, 2019 at 9:17 am

Thanks for answering.

[Reply](#)[Free eBooks](#)[About](#)[Contact](#)[Courses Login](#)[Get Courses](#)

Great resource, thank you.

Your site is one that I recommend to folks often as your courses are so well put together. Please keep it up 😊

[Reply](#)



**Sara Santos**

May 7, 2019 at 9:16 am

Thank you so much for recommending our our to your friends 😊

[Reply](#)



**Logan Byrne**

May 7, 2019 at 12:09 am

I have an ESP-WROOM-32 board, what pin-out should I follow? It looks kinda like the ESP8266 12-E NodeMCU Kit, pins are differently labelled. Any assistance.

[Reply](#)



**Sara Santos**

May 7, 2019 at 9:16 am

Hi Logan.

We have a similar guide for the ESP32. Take a look at it: <https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>

Regards,  
Sara

[Reply](#)



**Sarawut.c**

May 7, 2019 at 2:28 am

You've just answer my problem, very good for reference.

Thank you!

[Reply](#)



**Sara Santos**

May 7, 2019 at 9:15 am

[Reply](#)**Sonny**

May 7, 2019 at 6:18 am

Excellent guide. Thanks.

[Reply](#)**Nuno**

May 8, 2019 at 9:32 am

Greetings

Excellent guide.

I was struggling with a D1 mini, three switch buttons and four relays.

It seems that I must use an esp32.

Thanks

[Reply](#)**Sara Santos**

May 8, 2019 at 9:50 am

Yes, it should probably be more appropriate.

We also have a guide about the ESP32 GPIOs that might help you a lot: <https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>

Regards,

Sara

[Reply](#)**Lance Benson**

May 8, 2019 at 12:32 pm

A D1 mini should be able to handle 3 switch buttons and 4 relays. Note that RX and TX can be made to be GPIOs if need to be. I've been bitten by boot condition needs—specifically, I had a pullup for a DS18B20 on D8, which disabled normal booting.

It's also possible to add an MCP23017 16-bit I2C GPIO expander module (<\$3US on ebay) to gain a lot more I/Os on a D1 Mini (I like the D1 Mini a lot).

[Reply](#)

**Gerard**

August 17, 2019 at 6:17 pm

Hi Sara and Rui,  
What is the function of the "EN" pin on NodeMCU?

Cheers,  
Gerard

[Reply](#)**Sara Santos**

August 24, 2019 at 10:35 am

Hi Gerard.  
That's the Enable pin for the ESP8266. When it is pulled down, it resets the ESP8266.  
Regards,  
Sara

[Reply](#)**Jose Vazquez**

August 20, 2019 at 5:32 pm

Hey guys! This page is very useful!

I'm currently prototyping using a Wemos D1 mini, but the final processor will be an ESP12EX. Since it has more GPIOs available at the bottom (GPIO: 6, 8, 10, 9, 7, 11) do you know if these are usable? I'm planning to create my own breakout board since I don't need the extra stuff NodeMCU has. Thank you!

[Reply](#)**Sara Santos**

August 24, 2019 at 10:18 am

Hi Jose.  
Those pins are usually connected to the ESP8266 flash chip. So, it is better to use other pins.  
Regards,  
Sara

[Reply](#)**Steve**

August 26, 2019 at 11:55 pm

Thanks so much! Saved me a lot of grief, as I was going to use a few GPIO inputs that would have not worked well.

[Reply](#)

**Bogdan Coticopol**

October 3, 2019 at 11:13 pm

The information here is pure gold. Thank you for the detailed explanation on ESP8266!

[Reply](#)

**Sara Santos**

October 4, 2019 at 9:33 am

You're welcome 😊

[Reply](#)

**Rudo**

February 14, 2020 at 11:58 pm

Hello,

You have "MOSI" and "MISO" incorrectly swapped in the "SPI" section.  
Everywhere else it's correct: in the pinout pictures, the Best Pins to Use table.

Cheers.

[Reply](#)

**Sara Santos**

February 15, 2020 at 11:33 am

Hi.  
Yes, you're right!  
Thanks for noticing. It's updated now.  
Regards,  
Sara

[Reply](#)

**Scott Baker**[Free eBooks](#)[About](#)[Contact](#)[Courses Login](#)[Get Courses](#)

This is a fantastic reference. I was wondering if you would consider adding HTML id tags so I can to this page, but a specific section. Since the page is so long I'd like to be able to link directly to the pin table if possible.

[Reply](#)

**Sara Santos**

April 6, 2020 at 9:46 am

Hi Scott.

I added an ID to the table. You can use this URL: <https://randomnerdtutorials.com/esp8266-pinout-reference-gpios/#table>

Regards,  
Sara

[Reply](#)

**Antonius Widjaya**

April 22, 2020 at 5:03 pm

Can we use GPIO2 and GPIO14 as I2C line on ESP12F?

[Reply](#)

**Sara Santos**

April 23, 2020 at 10:50 am

Hi.

Yes, I think so.

You just need to declare that on the code:

```
Wire.begin(SDA, SCL);
```

Regards,  
Sara

[Reply](#)

**George s lavra**

May 13, 2020 at 6:04 pm

Nice tutorials. I have several 8266 01 boards that were programmed as OTA and just reporting near by wi fi stations. Lost the documentation, now trying to restore to factory fresh. Just can not accomplish the task. Will keep trying.

[Reply](#)

Hi.

Follow this: <https://rntlab.com/question/how-perform-reset-factory-esp32/>

Regards,

Sara

[Reply](#)



**Robert FOLKES**

May 28, 2020 at 4:54 pm

Really useful information, I wish I'd seen this before I did my first 8266 project the hard way.

One little trick that worked for me was using the 12-E NodeMCU. I found the 3.3V analogue input too high – I needed 1.0V (and expected this from the documentation I had at the time). With some careful soldering I was able to short out the upper voltage divider resistor giving me 1V sensitivity with (if I recall) 100 Kohm input impedance. Normally I wouldn't attempt such an intricate process but you just need to attach a wire to the copper pad on the chip, actually not so hard.

[Reply](#)



**Karanbir**

June 11, 2020 at 2:45 am

Hello,

I have come up with mention of the fact that the Max of voltage at ADC0 on bare ESP8266 chips is 1V.

Is this true of ESP8266 12-E/F.

And is this true of Wemos D1R1/R2 also.

Appreciate your response.

Best Regards.

[Reply](#)



**Robert Folkes**

June 11, 2020 at 6:15 am

Yes, the bare chip has a maximum voltage of 1v, of course it is possible to increase that with a simple potential divider network (2 resistors, one of which is probably already on your chip/board). Some of the development boards add a potential divider network usually to make the sensitivity 3.3v at the A0 pin, I only have direct experience of the NodeMCU board where this is the case, but it might also be true of the products you mention.

The input impedance at A0 is 330Kohm (probably true for all such boards), so if you want to increase the max voltage further, you can add a resistor in series with A0  $R = V_{max} * 100-330 K$ . As I said previously if you want to decrease the  $V_{max}$  with a little soldering\* skill you can attach a resistor directly to the analogue input pin of the chip itself  $R = V_{max}*100 - 100 K$ .

\*Not sure about the wisdom of this (may not be legal in your jurisdiction), but I have some old leaded solder (left over from my youth before health and safety) it melts about 20 degrees lower than unleaded, so if you set your regulated iron to coolish, its great if you want to tweak boards with surface mount components without risk of disturbing

[Reply](#)**Brad G**

July 4, 2020 at 3:38 am

Please, Please put together a pinout page like this one. Every time I google for images on espXXX pinouts, I always get your page. I looked but was not able to find one for the Arduino boards. Yours is clean and concise.

It would be a GREAT addition !!!

As always, thanks for all the work and time you put in!

BG

[Reply](#)**Sara Santos**

July 4, 2020 at 11:30 am

Thanks for the suggestion.

Regards,  
Sara

[Reply](#)

## Leave a Comment

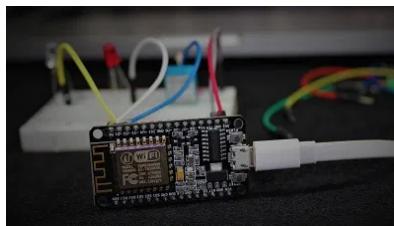
 Name \* Email \* Website Notify me of follow-up comments by email. Notify me of new posts by email.



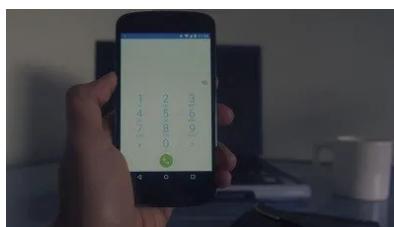
	ESP8266 NodeMCU MQTT - Publish DHT11/DHT22 Temperature and Humidity Readings (Arduino IDE)
	ESP-NOW with ESP32: Send Data to Multiple Boards (one-to-many)
	Install uPyCraft IDE - Windows PC Instructions
	ESP32 Capacitive Touch Sensor Pins with Arduino IDE
	Nextion Display with ESP8266 - Touchscreen User Interface for Node-RED



[Visit Maker Advisor – Tools and Gear for makers, hobbyists and DIYers »](#)



**Home Automation using ESP8266**  
[eBook and video course »](#) Build IoT and home automation projects.



[from Scratch »](#) With Raspberry Pi,  
ESP8266, Arduino, and Node-RED.

[About](#) [Support](#) [Terms](#) [Privacy Policy](#) [Refunds](#) [MakerAdvisor.com](#) [Join the Lab](#)

Copyright © 2013-2020 · RandomNerdTutorials.com · All Rights Reserved