

**TUTORIAL EXAMPLE:****Que. What is an ESP-12E NodeMcu V3 module?**

**Ans.** NodeMCU is an [LUA](#) based interactive firmware for Express if ESP8622 Wi-Fi SoC, as well as an open source hardware board that, unlike the \$3 ESP8266 Wi-Fi modules, includes a CP2102 TTL to USB chip for programming and debugging, is breadboard-friendly, and can simply be powered via its micro USB port.

This module is one of the cheapest available Wi-Fi-modules in market. V3 or Version3 is the latest version of this module. This tutorial however will facilitate you to connect all the versions of ESP8266 NodeMcu, i.e V1, V2 or V3.

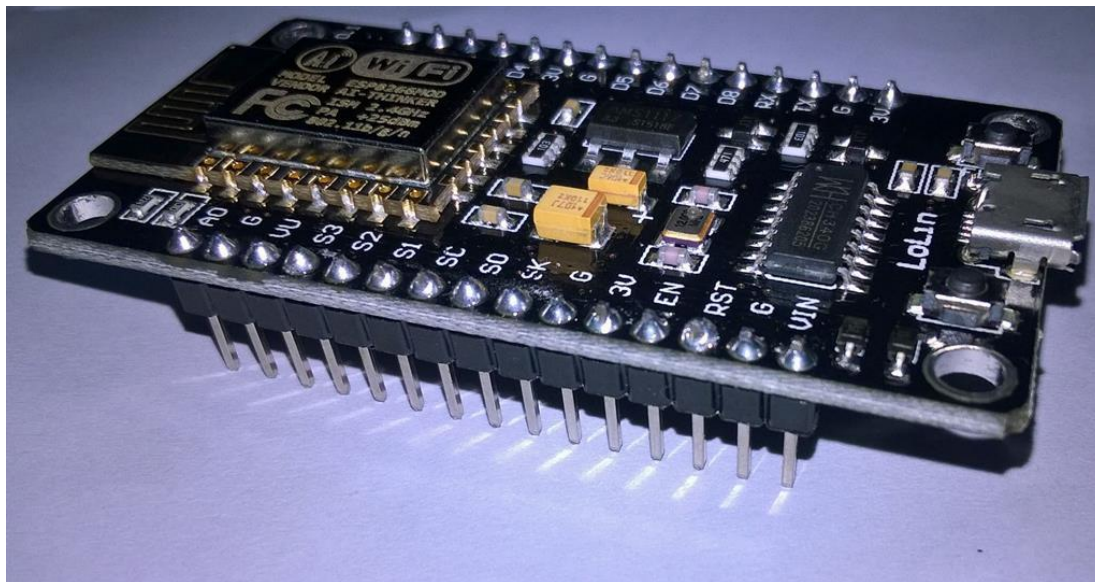
For makers, having ESP-12 module, rest assured, 12E is no different from its precursor. The only difference: 6 extra GPIOs are present in ESP-12E

**Headers:** 15-pin header with access to GPIOs, SPI, UART, ADC, and power pins.

**Power:** 5V via micro USB port

**Hardware Components required:**

**ESP 12E NodeMcu V3,** A breadboard for placing the LED and a couple of Jumper wires to connect them.

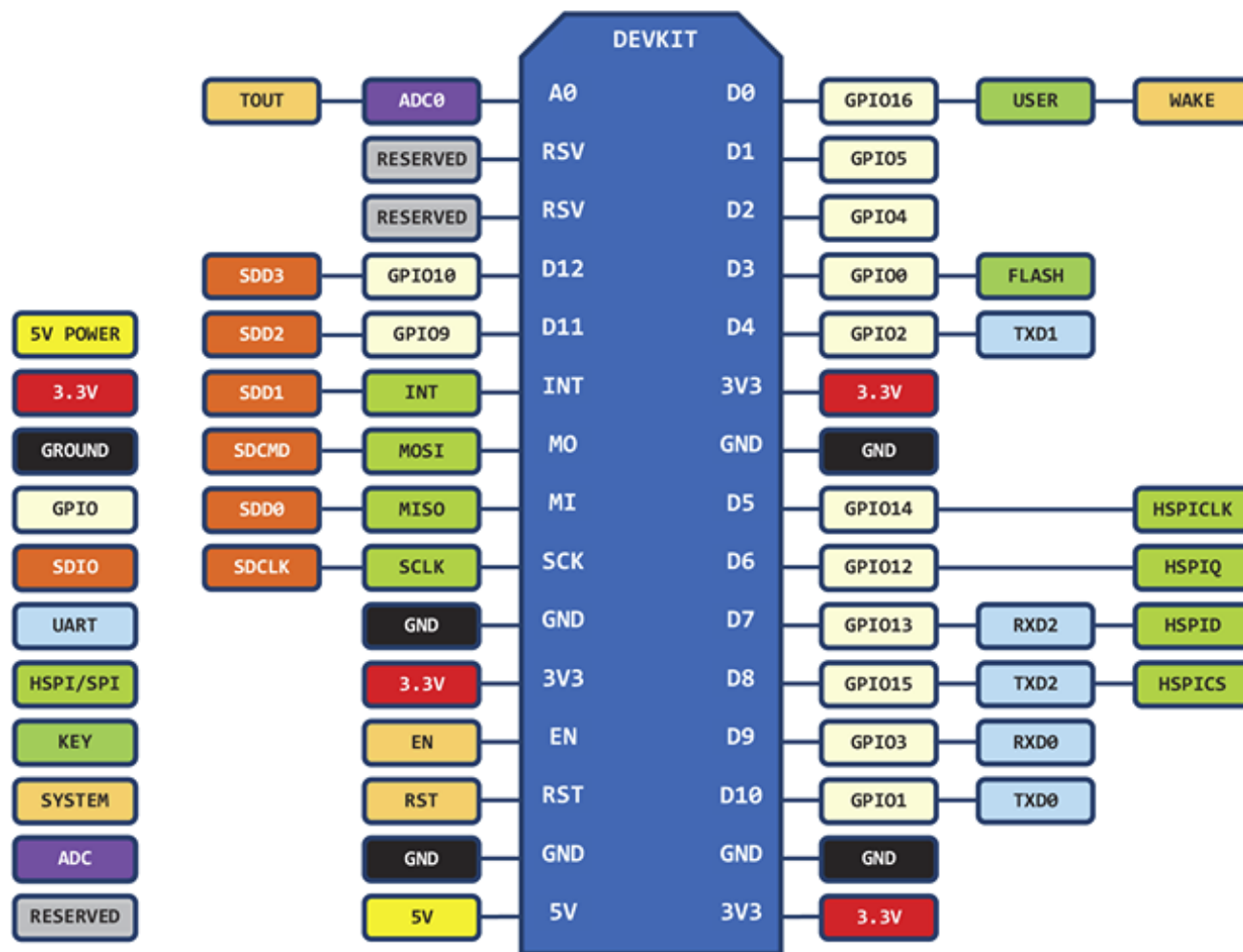


This NodeMcu breakout module is built around ESP-12E board

**Software required:** [Arduino](#) (In future tutorial, we'll see how to work in Lua)

But, since most of us are familiar with ArduinoIDE, lets ponder upon this for awhile.

**Header PINOUT:** (Save this picture for future references)



*D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.*

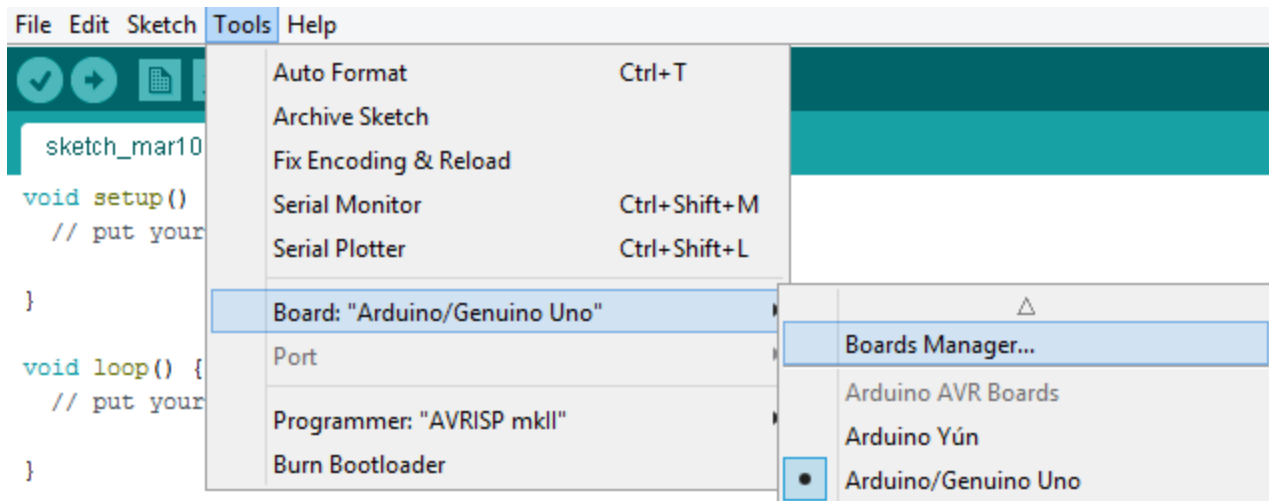
*NodeMcu header pin-out*

After you've assembled all the hardware components, click on this [link](#) and download all the files. The file contains requisite drivers needed to use this module. This will save you a lot of time.

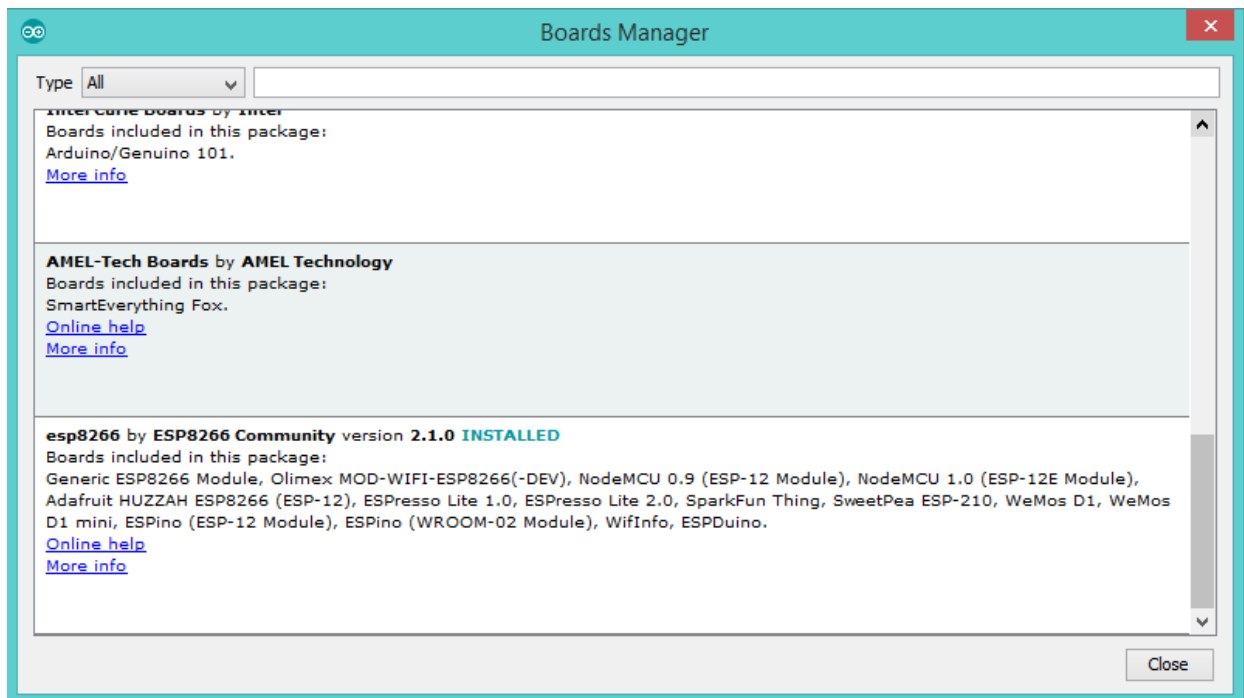
Now, the ZIP file you've just downloaded, will be put to use in this tutorial as well as in the forthcoming tutorials.

Well, it's very simple!

First of all, your ArduinoIDE needs to know that you want to use this NodeMcu module. That is, you need to select the right board. Confused? Follow this flowchart: (**Tools**, **Board**, then **Boards Manager**).

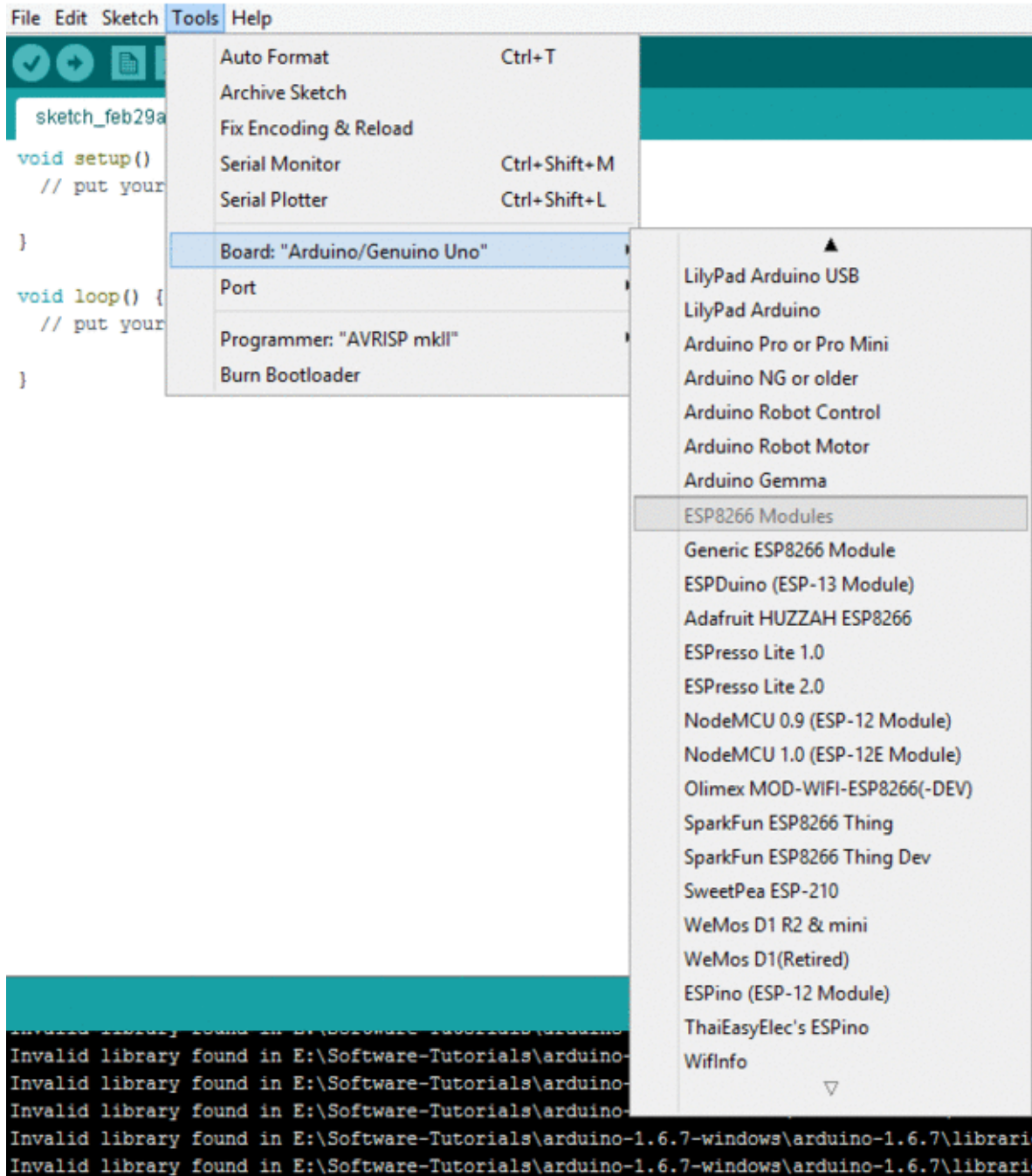


Next, select the requisite board.



Having done so, you can cross-check. (**Tools, Board**, then **ESP8266 Modules**)

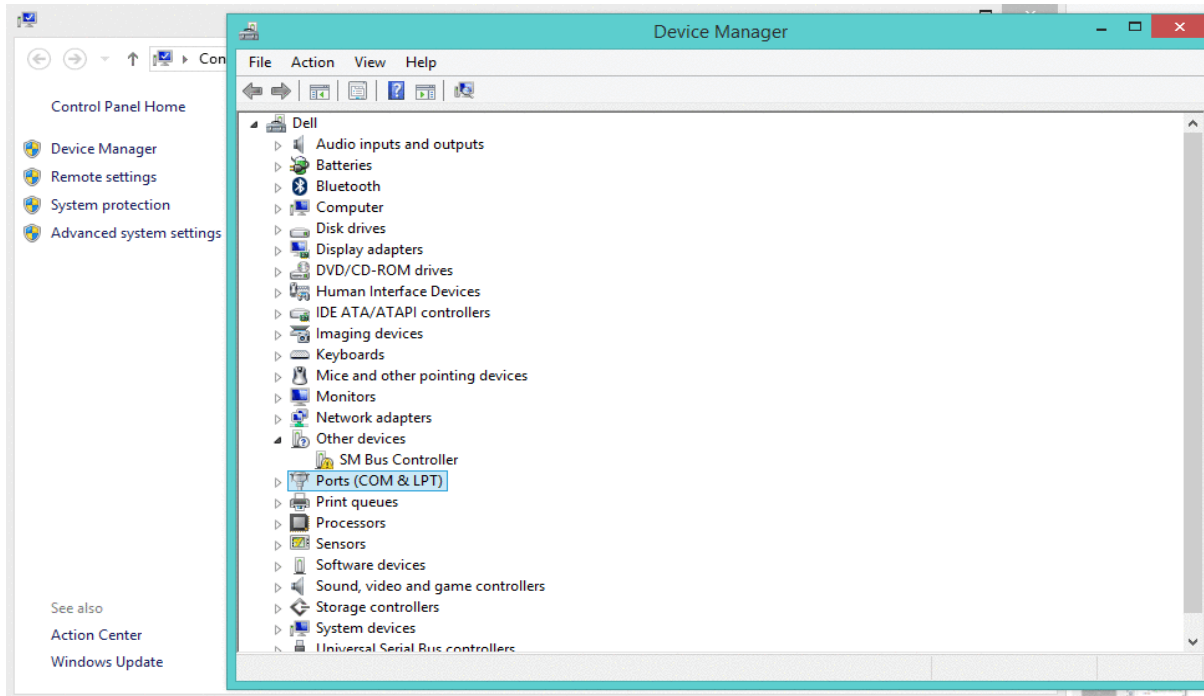
Out of this entire list, you need to select "NodeMCU 1.0 (ESP-12E Module)" as in the figure below:



Now, select the correct port. Can't recognize the port?? Wait!

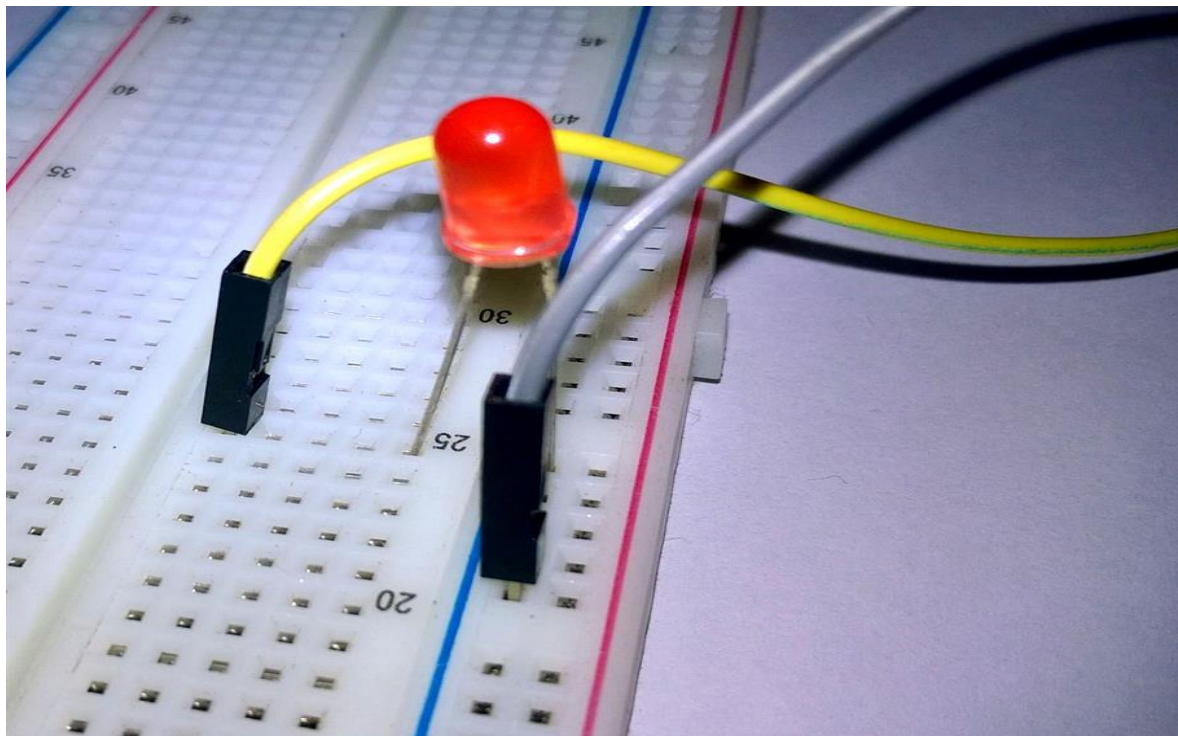
Check this out.....

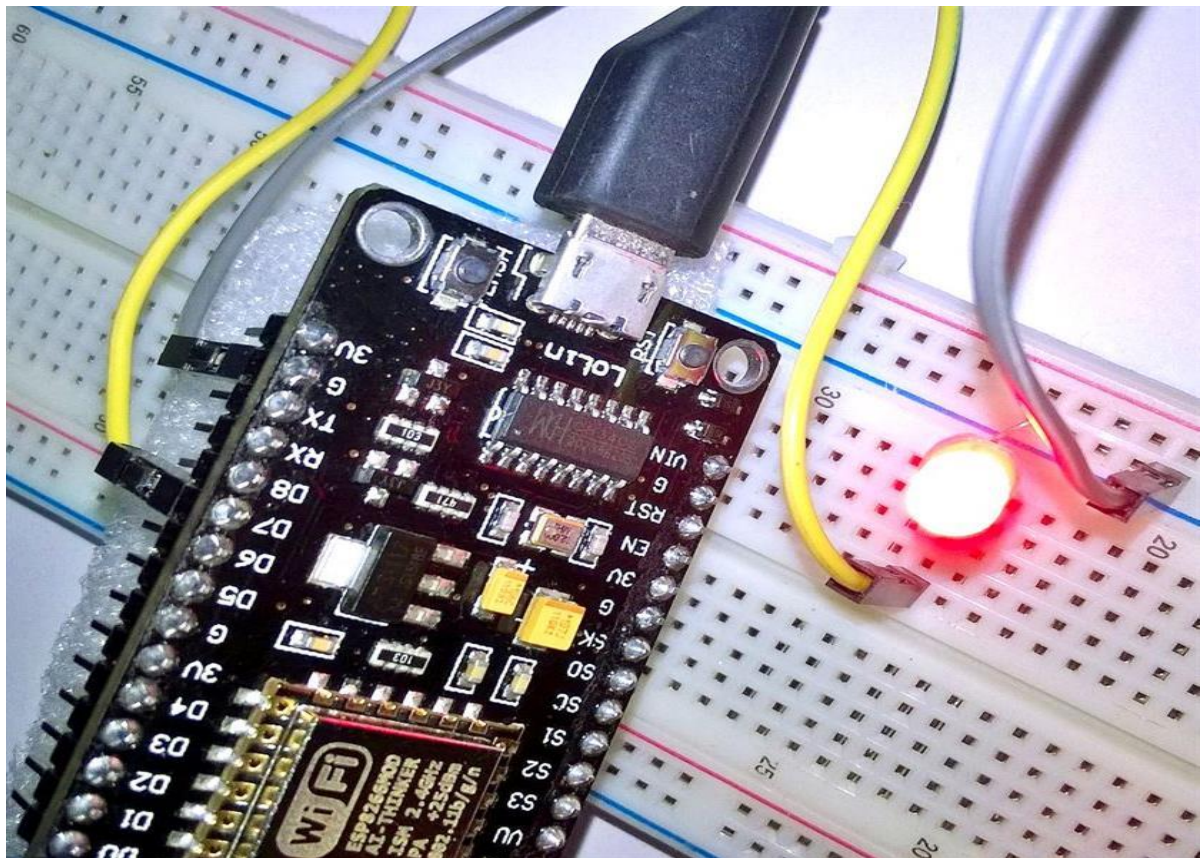
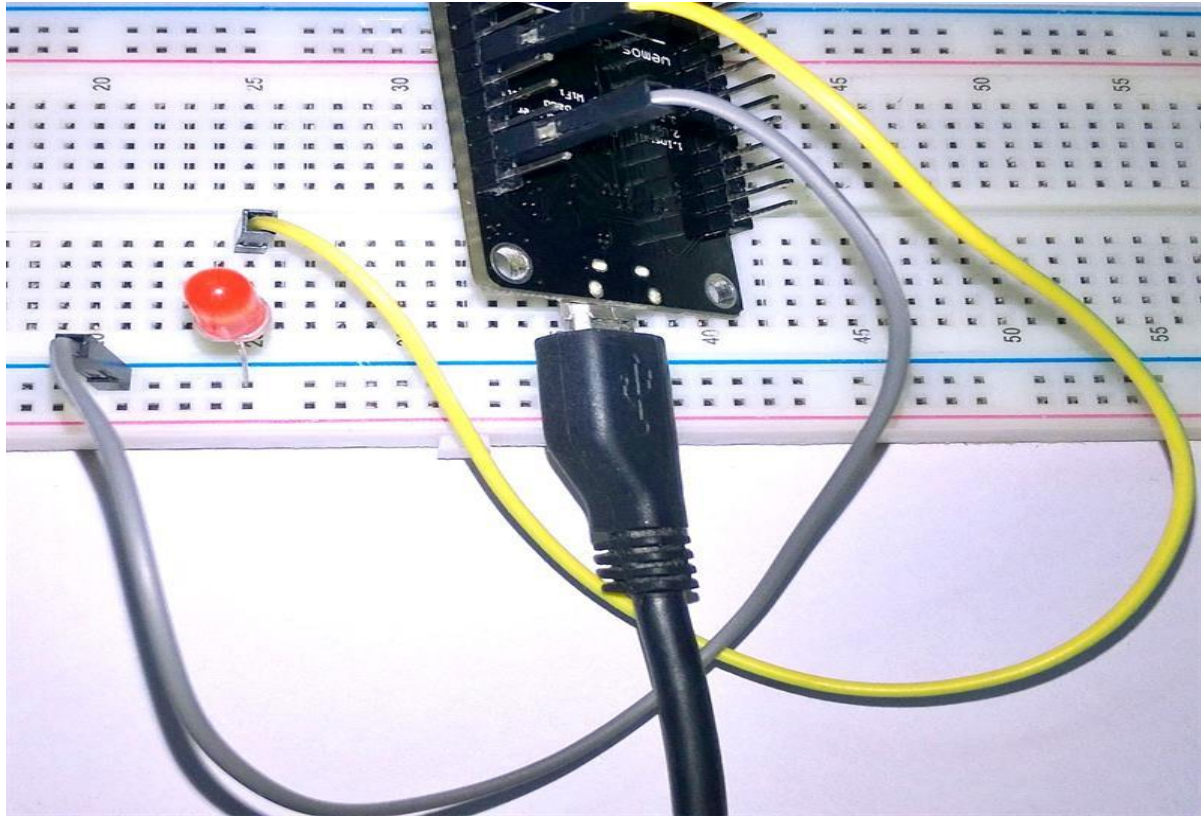
Go to the **Device Manager** and update the driver. (*Follow the GIF below*)



Now, its a matter of few minutes before you can get started.... Let's proceed.

### Connecting the LED to NodeMcu: (Zoomed-snapshot of the Circuit)





### Connection Profile:

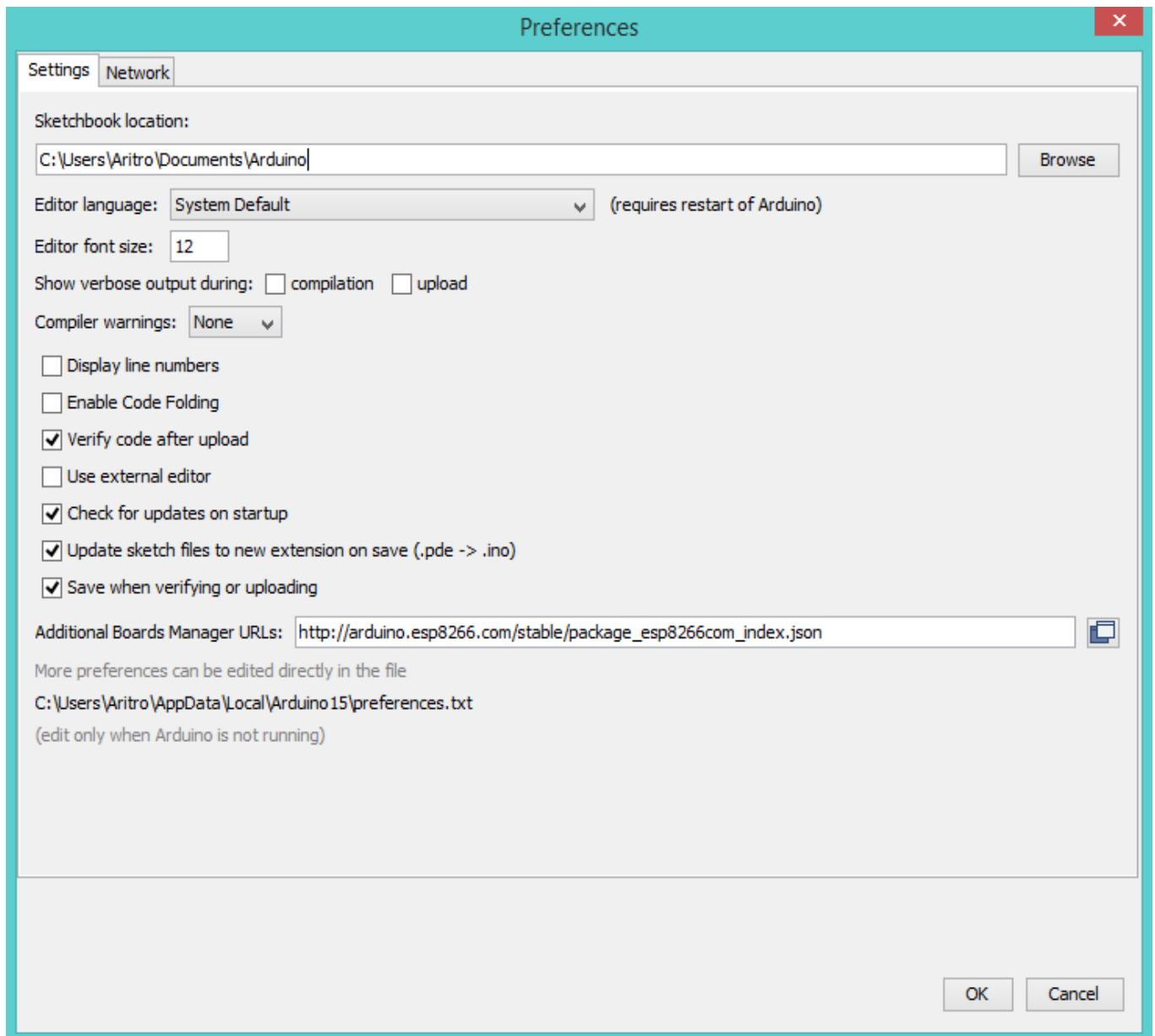
Connect **D7 pin** (GPIO13 of NodeMcu) to the long leg of the LED (the positive leg, called the anode).

Connect **G(Ground)pin of NodeMcu** to the short leg of the LED (the negative leg, called the cathode)

Now, come back to the ArduinoIDE. You need to set up the preferences.

(File > Preferences)

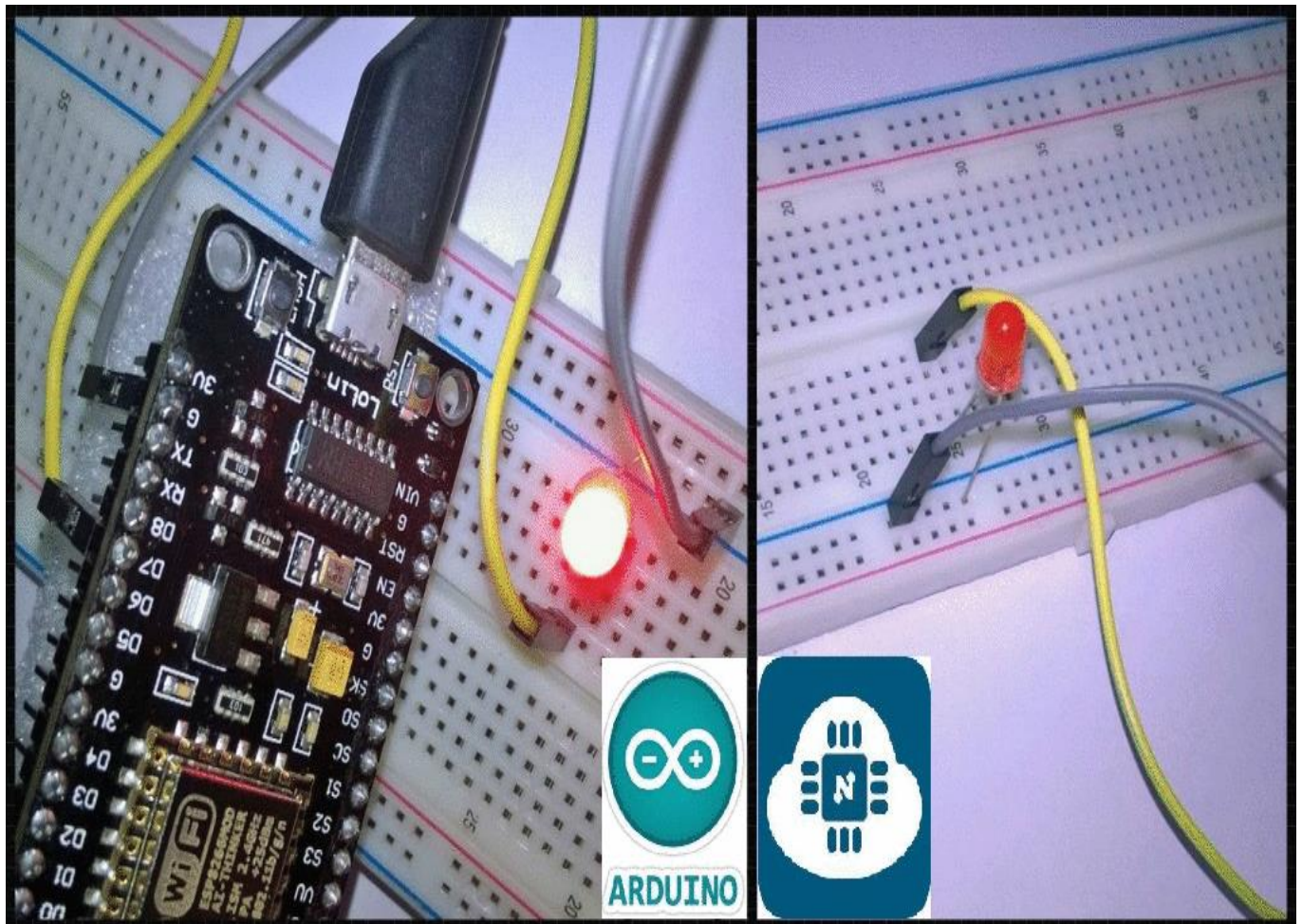
Copy this [URL](#) and paste it in the **Additional Boards Manager URLs** option.



### Final Task: Upload the code for Blinking an LED

```
1. void setup()
2. {
3.   // initialize digital pin 13 as an output.
4.   pinMode(13, OUTPUT);
5. }
6.
7. // the loop function runs over and over again forever
8. void loop() {
9.   digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
10.  delay(1000);           // wait for a second
11.  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
12.  delay(1000);           // wait for a second
13. }
```

Next, run the code. And your job's done!



Congratulations! You've successfully interfaced an ESP NodeMcu...and even Blinked an LED!