

# ARDUINO IDE SETTINGS FOR ESP8266 12E NODEMCU

## INTRODUCTION:



Wi-Fi is an essential bit of kit for any Internet of Things (IoT) DIY projects, but our favorite Arduino doesn't come with Wi-Fi, and adding in a Wi-Fi shield can increase the costs substantially.

Meet the ESP8266 12E (also known as NodeMCU) was originally marketed as a low cost Wi-Fi add-on for Arduino boards, until the hacker community realized you could cut the Arduino out of the equation entirely.

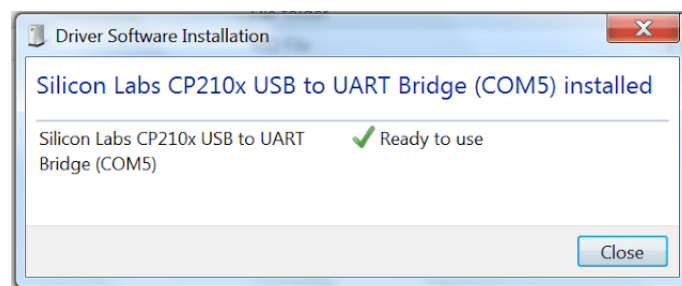
In less than a year, the ESP8266 has rocketed in popularity, and is now so well supported and developed.

There's quite a few models of ESP8266 around now, but we recommend this one: ESP-12E (also known as NodeMCU 2.0). This board includes the serial driver needed to program the chip, and has a built-in power regulator, as well as lots of IO pins. This NodeMCU Devkit is available from our Store.

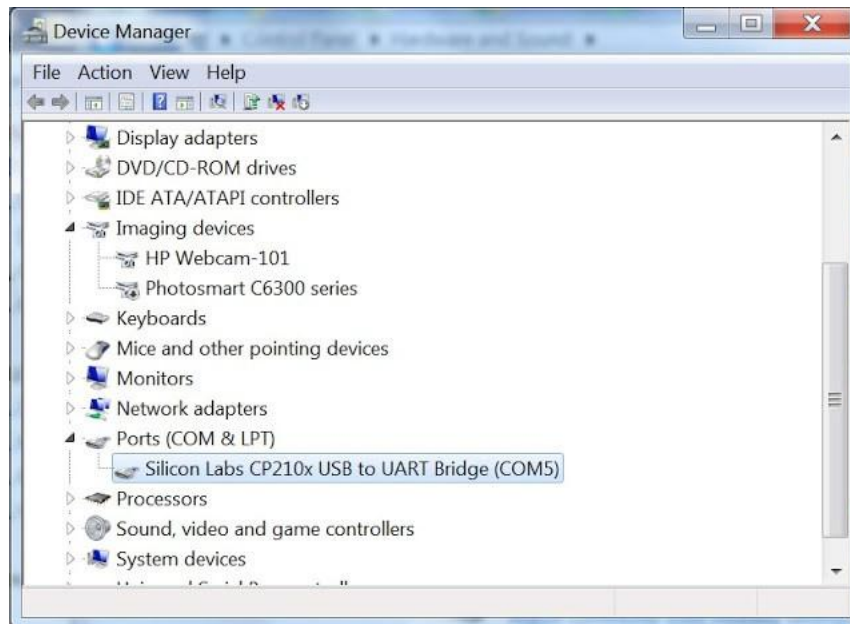
## GETTING STARTED WITH ESP8266-12E:

Using Micro usb to standard usb cable Connect ESP8266-12E to your PC. we are using windows 7.

Wait and allow the windows to install the USB driver as shown below.



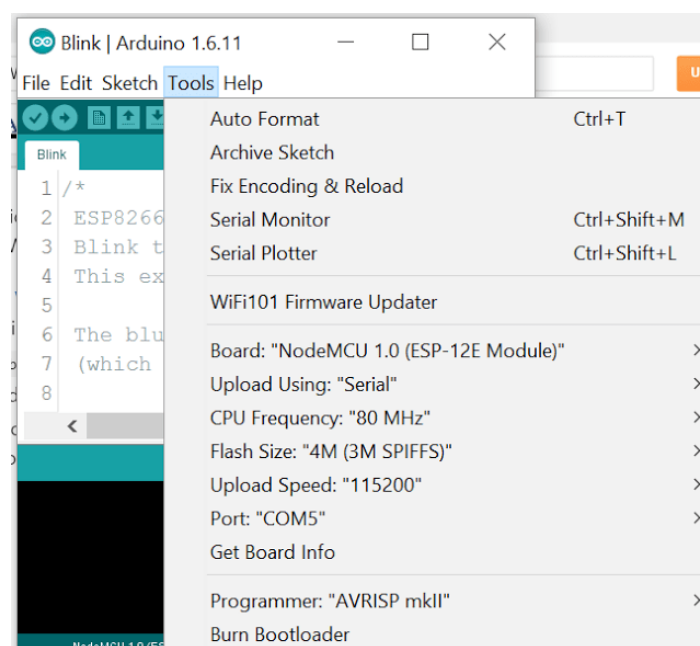
NodeMCU is connected to COM5 . This can also be verified from Device Manager.



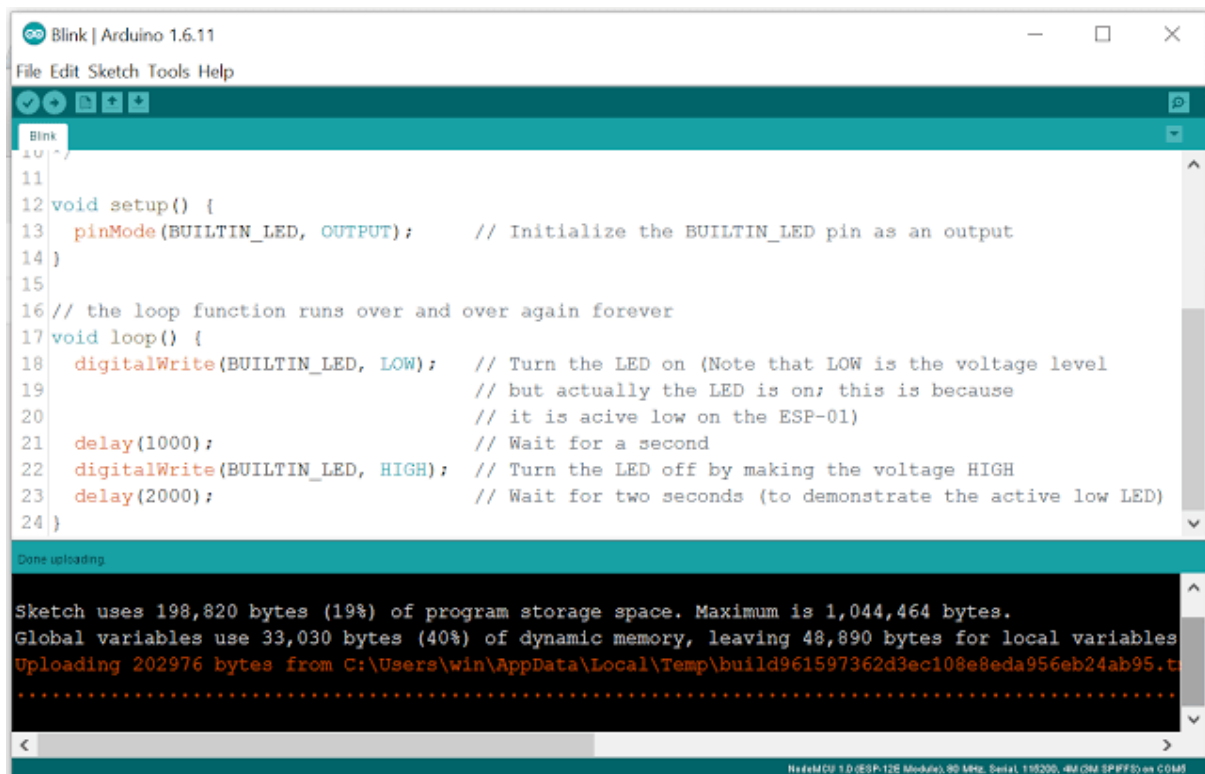
## ARDUINO IDE SETTINGS

- Install the latest Arduino from the Arduino website.
- Start Arduino and open Preferences window.
- Enter [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) into Additional Board Manager URLs field. You can add multiple URLs, separating them with commas.
- Open Boards Manager from Tools > Board menu and install esp8266 platform

Selecting the NodeMCU board from Tools Menu



Restart Arduino IDE and select blink Example for ESP8266 and upload.



```
Blink | Arduino 1.6.11
File Edit Sketch Tools Help

Blink
10 ~
11
12 void setup() {
13   pinMode(BUILTIN_LED, OUTPUT); // Initialize the BUILTIN_LED pin as an output
14 }
15
16 // the loop function runs over and over again forever
17 void loop() {
18   digitalWrite(BUILTIN_LED, LOW); // Turn the LED on (Note that LOW is the voltage level
19                                   // but actually the LED is on; this is because
20                                   // it is active low on the ESP-01)
21   delay(1000); // Wait for a second
22   digitalWrite(BUILTIN_LED, HIGH); // Turn the LED off by making the voltage HIGH
23   delay(2000); // Wait for two seconds (to demonstrate the active low LED)
24 }
```

Done uploading.

Sketch uses 198,820 bytes (19%) of program storage space. Maximum is 1,044,464 bytes.  
Global variables use 33,030 bytes (40%) of dynamic memory, leaving 48,890 bytes for local variables  
Uploading 202976 bytes from C:\Users\win\AppData\Local\Temp\build961597362d3ec108e8eda956eb24ab95.t  
.....

< >

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Serial, 115200, 4M (0M SPIFFS) on COM5