

The first usage of NodeMCU ESP12E DEVKIT V2

by **Bay Yola** on April 1, 2016

Table of Contents

The first usage of NodeMCU ESP12E DEVKIT V2	1
Intro: The first usage of NodeMCU ESP12E DEVKIT V2	2
Step 1: Let's take a look as overview	3
Step 2: The first time to connect to pc	4
Step 3: How to use Arduino IDE to program NodeMCU?	4
Step 4: Blink code for NodeMCU - Hello World LED	6
Step 5: Let's make some LAN joy :) - Hello World on Browser	7
Related Instructables	8
Advertisements	9
Comments	9

Intro: The first usage of NodeMCU ESP12E DEVKIT V2

Today, I am going to show you how to use NodeMCU ESP12E DEVKIT V2 for the first time. At very first, let's call this board as NodeMCU development board, although it has got ESP8266MOD Module on it, that makes this board is fit to call development board enough. Or, whatever you call it is okay. NodeMCU board lets us to go to the internet via Wi-Fi. Such shields Arduino Wi-Fi Shield, ethernet shield; NodeMCU is much more powerful than the others, I think.

The need list:

1. NodeMCU ESP12E DEVKIT V2

Note: There are two version types of NodeMCU: **NodeMCU v0.9 with ESP-12 module**, and **NodeMCU v1.0 with ESP-12E module**. On this instructable, I will use NodeMCU v2 ESP-12E which refers to NodeMCU v1.0 with ESP-12E module. Arduino v1.6.8 knows as 1.0 for ESP-12E module, although it is v2. So I can say that v1.0 means v2.0 that's because of Arduino side not about NodeMCU. [The source](#)

Read more: <http://www.cnx-software.com/2015/10/29/getting-st...>

2. Arduino IDE

3. Your computer (Windows is okay on here)

4. Micro USB cable

5. Wi-Fi Network (We need SSID and Password to get used on next steps, wireless modem would be okay)

Note: While you give an energy to NodeMCU via micro usb cable from your computer to that, ESP8266 on-boarded module may hot. This is moderate. I mean, there is no danger, a little bit heat is not seem a dangerous thing for our module. Be aware, surface is hot enough.

Note: On here, I am using Microsoft Windows. For Linux user, [check this link](#).



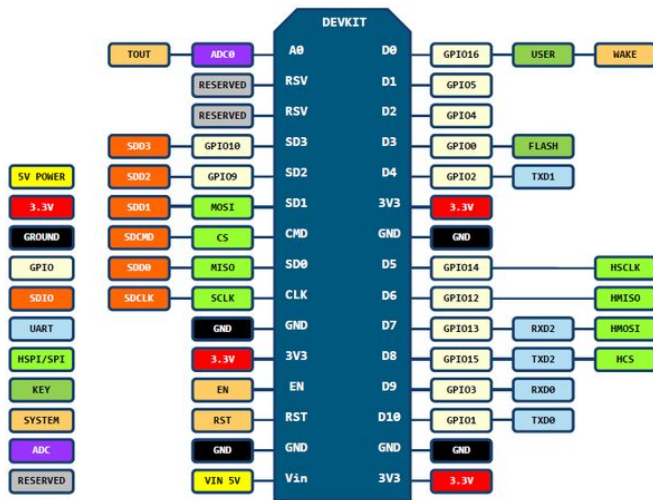
Step 1: Let's take a look as overview

On top side, there is an other board that you can see, this is the ESP part. ESP8266 is a module, but this is on-boarded circuit organism which let us to connect over wi-fi.

On rear side, there is written name given: ESP12E DEVKIT V2 and the link <http://www.doit.am/> for reference.



PIN DEFINITION

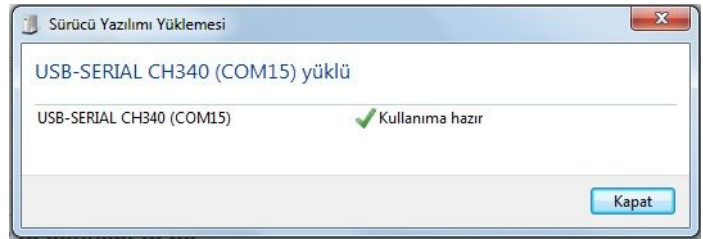
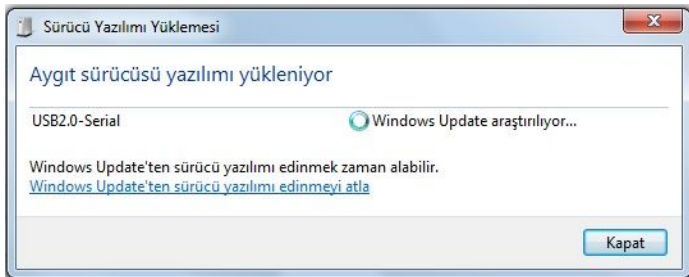


D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/iw supported.

Step 2: The first time to connect to pc

When we make a connection between NodeMCU board and computer, there is automatically driver loading window that you can easily learn which COM Port Number that your computer had gave. Mine is successfully loaded the driver and gave me COM Port Number 15 to access to my NodeMCU board.

There are two black buttons near the micro usb port. One is RESET other is FLASH. When you click on RESET button, you can see ESP8266 side's blue is blinked.



Step 3: How to use Arduino IDE to program NodeMCU?

1. Install the Arduino IDE from download section of arduino.cc website (Also, check my instructable about installing, updating, removing Arduino IDE from here)

2. Run Arduino IDE's arduino.exe

3. Arduino IDE >> File >> Preferences (Shourtcut is CTRL + COMMA)>> Settings tab >> on Additional Board Manager URL side type this >> and click ok.

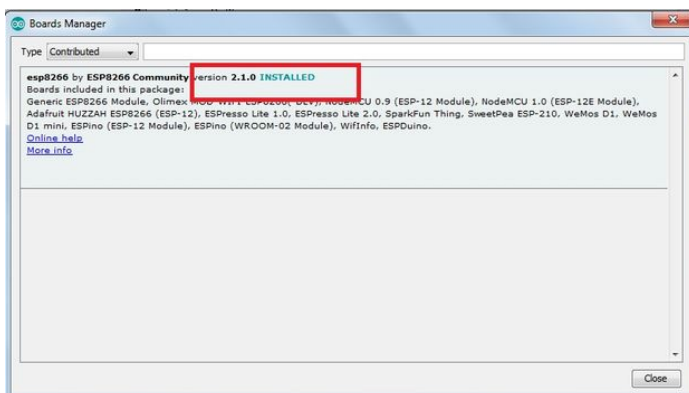
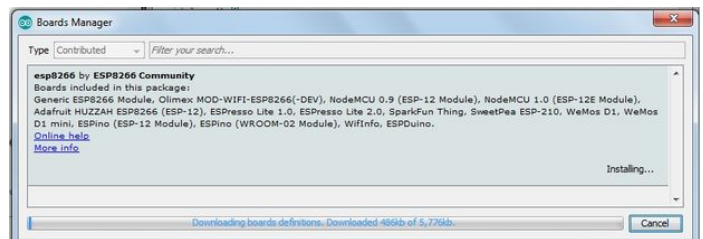
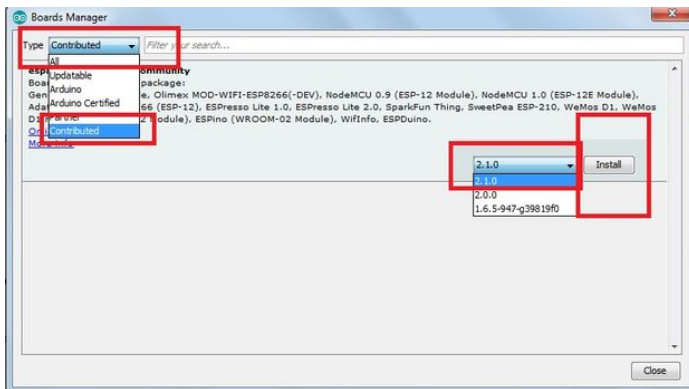
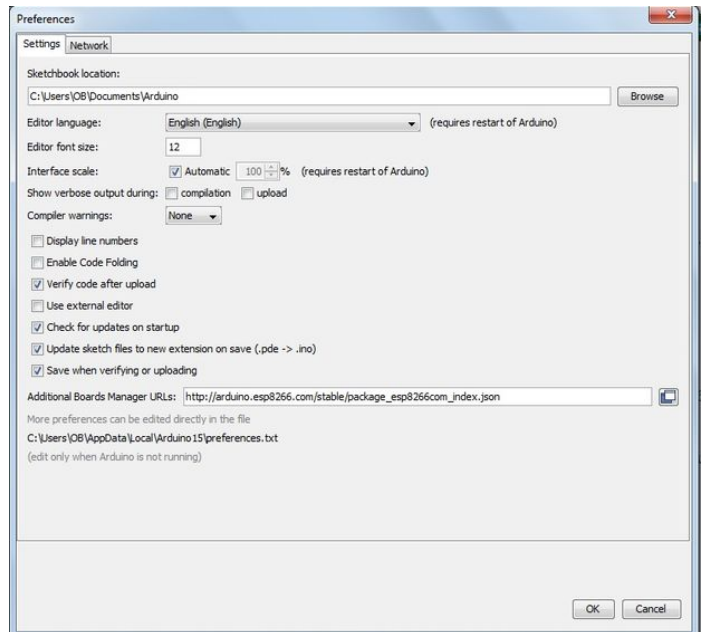
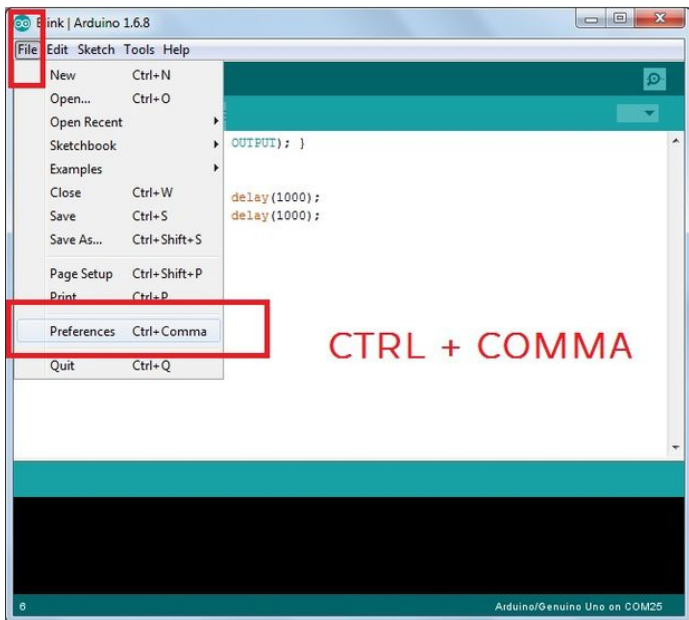
http://arduino.esp8266.com/stable/package_esp8266com_index.json

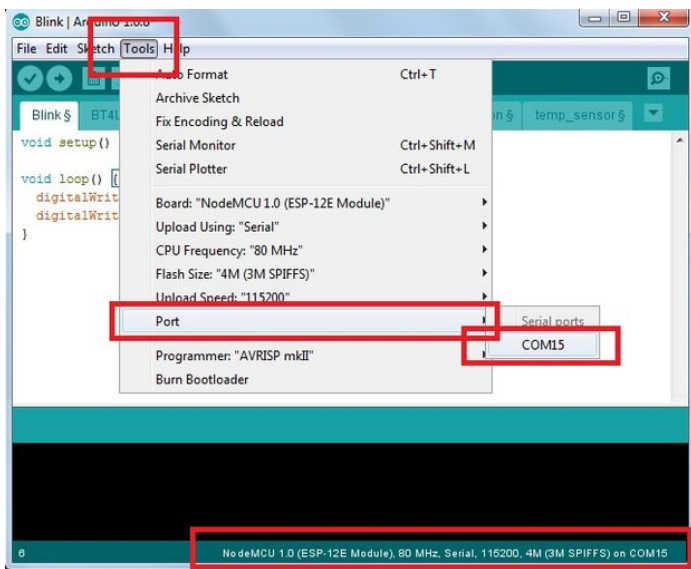
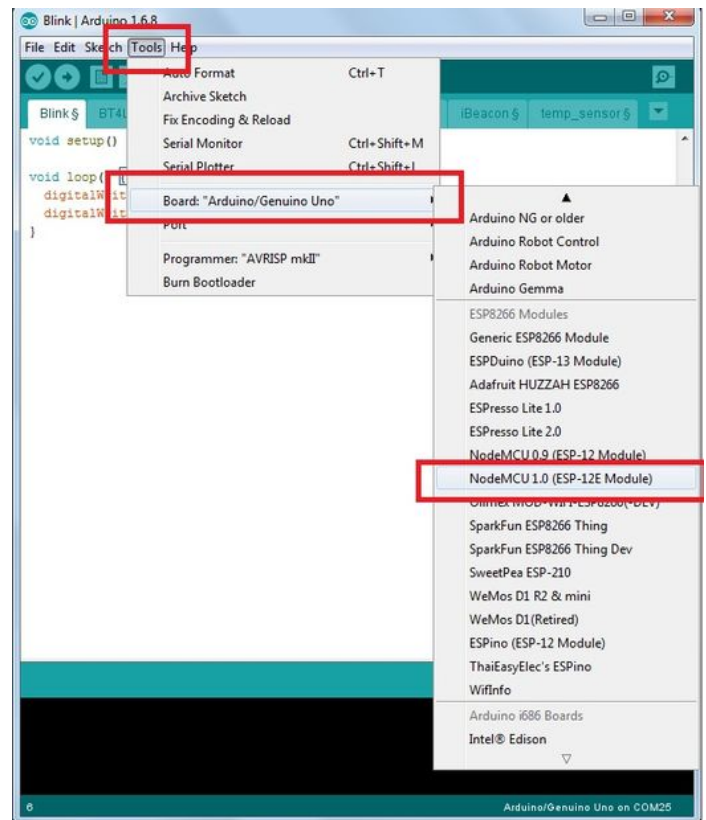
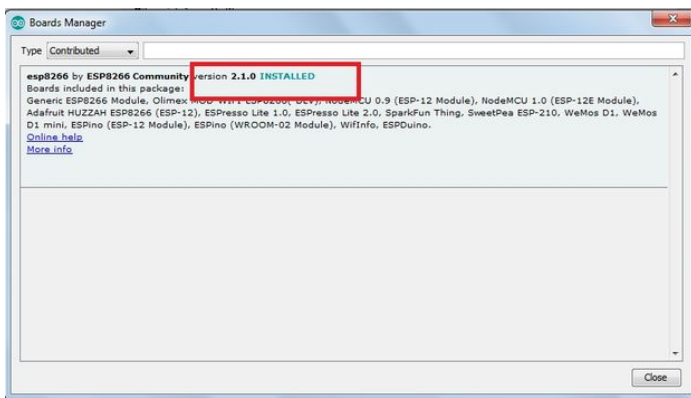
This json object will bring necessary drivers for NodeMCU from the website [esp8266.com](http://arduino.esp8266.com) to our Arduino IDE.

4. Arduino IDE >> Tools >> Board >> Boards Manager... >> Type = Contributed >> click on install (Or, search ESP8266 from the search bar)

5. Arduino IDE >> Tools >> Board >> NodeMCU 1.0 (ESP-12E MODULE)

NOTE: We have NodeMCU v2, but we select as 1.0. Focus on inside of paranthesis ESP-12E is more important than this.





Step 4: Blink code for NodeMCU - Hello World LED

On Arduino IDE >> File >> Examples >> ESP8266 >> Blink

/*

ESP8266 Blink by Simon Peter Blink the blue LED on the ESP-01 module This example code is in the public domain The blue LED on the ESP-01 module is connected to GPIO1 (which is also the TXD pin; so we cannot use Serial.print() at the same time) Note that this sketch uses LED_BUILTIN to find the pin with the internal LED */

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT); // Initialize the LED_BUILTIN pin as an output }
```

```
// the loop function runs over and over again forever void loop() {
  digitalWrite(LED_BUILTIN, LOW); // Turn the LED on (Note that LOW is the voltage level
  // but actually the LED is on; this is because // it is active low on the ESP-01) delay(1000); // Wait for a second
  digitalWrite(LED_BUILTIN, HIGH); // Turn the LED off by making the voltage HIGH delay(2000); // Wait for two seconds (to demonstrate the active low
  LED) }
```

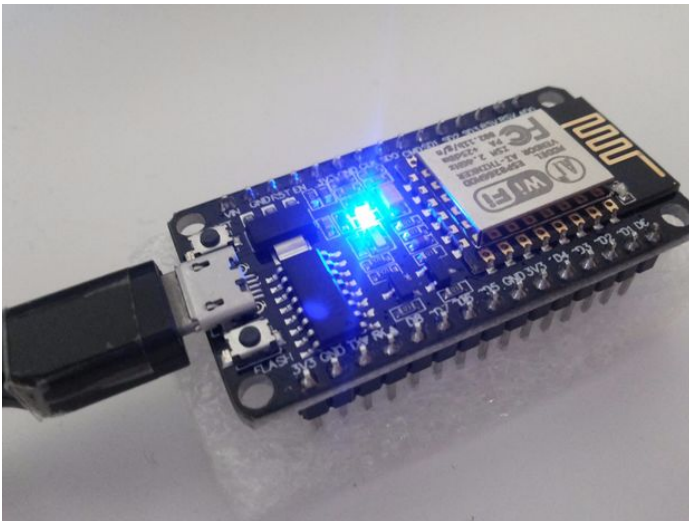
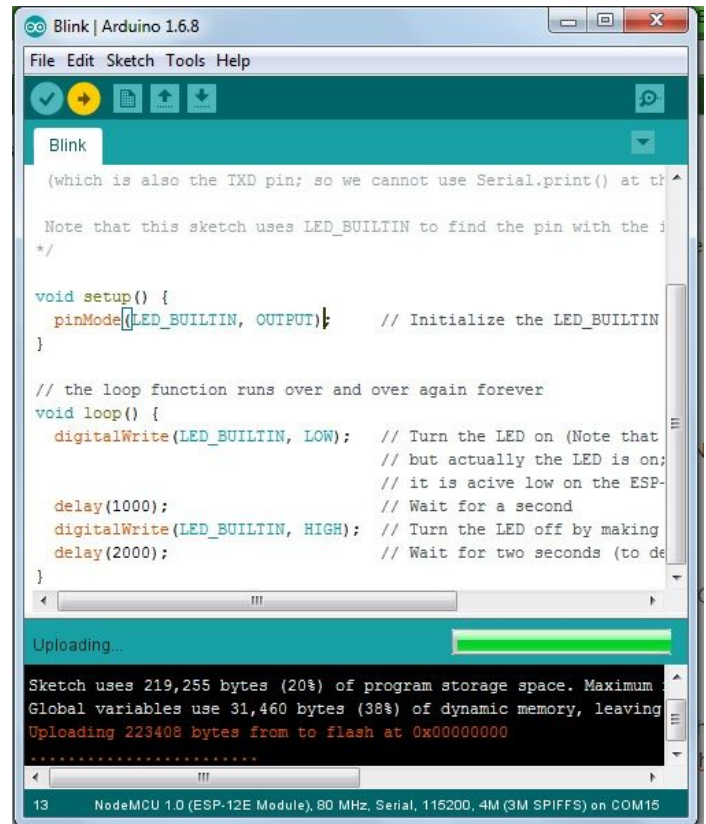
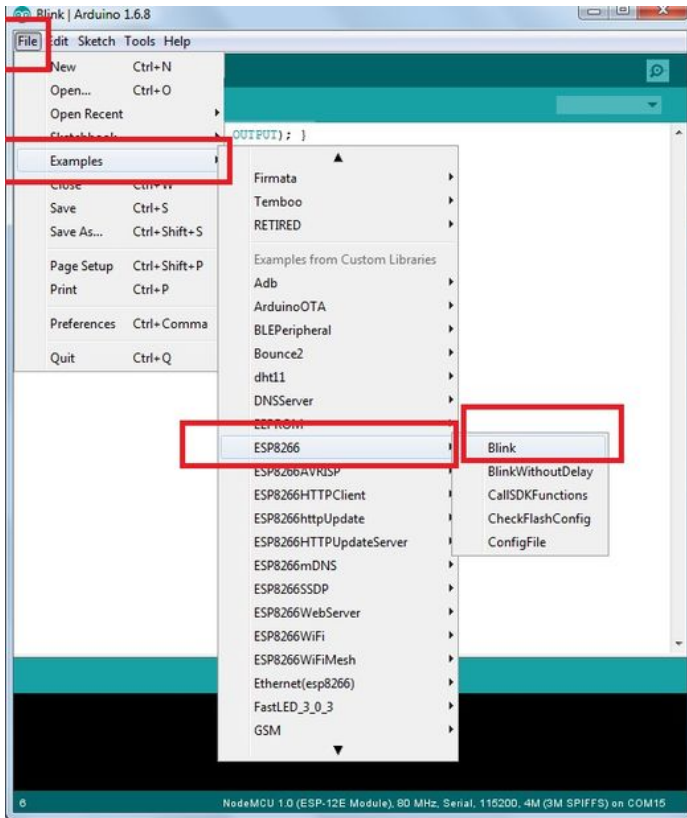
Attach NodeMCU to your computer with micro usb cable then click on upload button to upload your Blink codes to NodeMCU.

This code will run your NodeMCU's blue led for 1 second then close in a forever loop.

<http://www.instructables.com/id/The-First-Usage-of-NodeMCU-ESP12E-DEVKIT-V2/>

Now, we have working NodeMCU board just like an Arduino boards basically.

Note: On this site, it is mentioned that Arduino IDE v1.6.7 may not working well with NodeMCU, if you faced such an Arduino IDE problem, you need to update your Arduino IDE program. I can tell you how to update your program now.



Step 5: Let's make some LAN joy :) - Hello World on Browser

On Arduino IDE >> File >> Example >> ESP8266 >> ESP8266HelloServer >> Modify these lines:

```
const char* ssid = "....."; //Your Wi-Fi Modem's SSID name const char* password = "....."; //The password for the Wi-Fi
```

Then click on upload button. After uploading, open Serial Monitor. Click on reset button then learn which IP adress you need to use from Serial Monitor

NOT: We are using Baud Rate as: 115200

Now you did it well!! :D

That's all!

```
File Edit Sketch Tools Help
HelloServer
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP());

if (MDNS.begin("esp8266")) {
  Serial.println("MDNS responder started");
}

server.on("/", handleRoot);

server.on("/inline", [](){});
server.send(200, "text/plain", "this works as well");

server.onNotFound(handleNotFound);

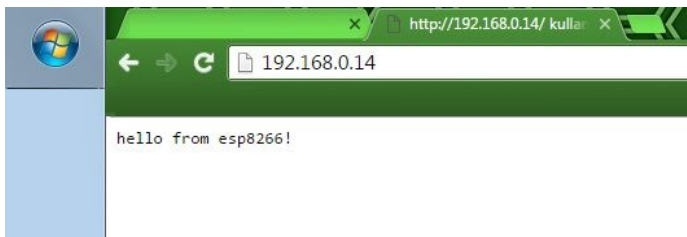
server.begin();
Serial.println("HTTP server started");
}

void loop(void){
  server.handleClient();
}
```

Uploading...

Sketch uses 250,000 bytes (23%) of program storage space. Maximum is 1,044,464 bytes.
Global variables use 36,276 bytes (44%) of dynamic memory, leaving 45,644 bytes for local variables.
Uploading 254144 bytes from to flash at 0x00000000

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Serial, 115200, 4M (3M SPIFFS) on COM15



Related Instructables



Programming ESP8266 ESP-12E NodeMCU Using Arduino IDE - a Tutorial
by TheElectromania



wireless logger ESP8266 NodeMCU v1.0 with Arduino IDE
by shinteo



Get Started with ESP8266 Using AT Commands, NodeMCU, or Arduino (ESP-12E)
by acrobotic



Flashing NodeMCU on esp8266 and Getting Started
by ARK4579



NodeMCU onboard LED access via web server
by AslanA1



A very cheap ESP8266 WiFi smart car controlled by phone based on NodeMCU
by yichone

Comments