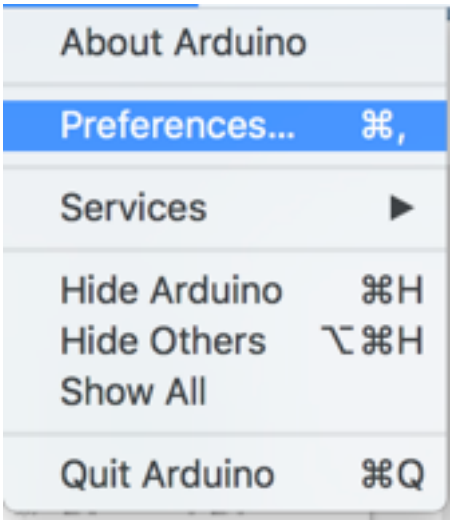
SETUP NODEMCU on Arduino IDE

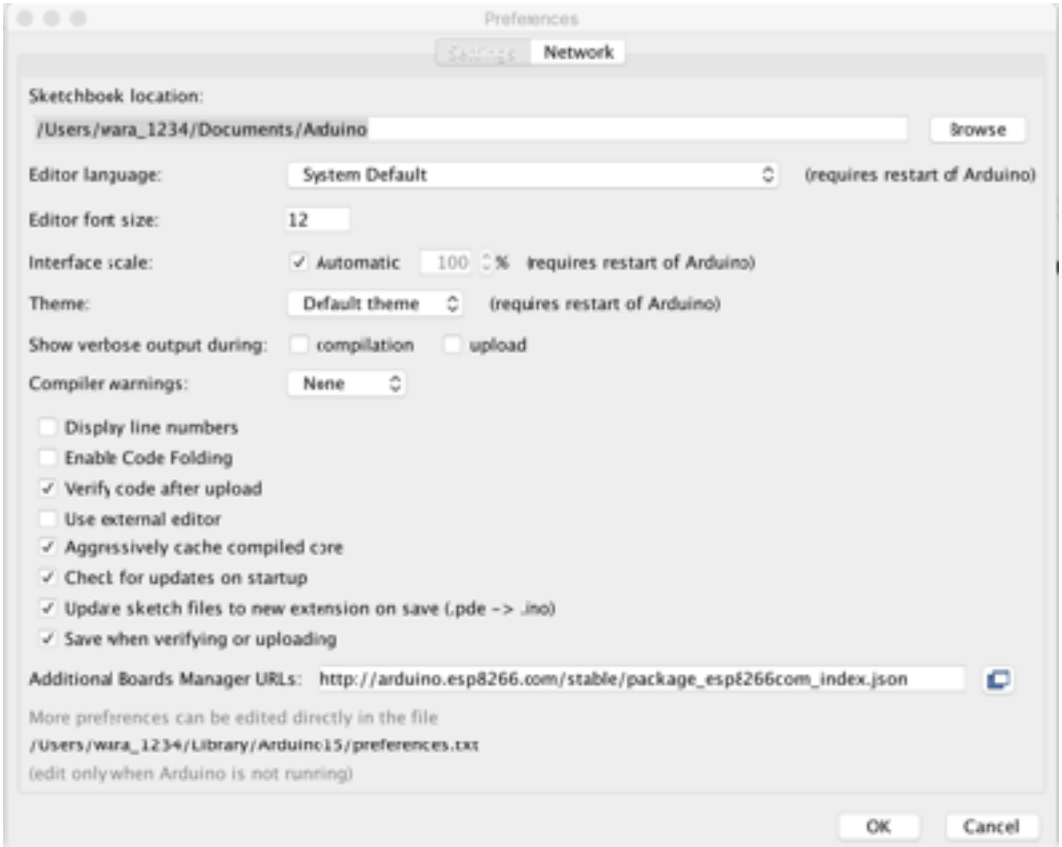
You can try the following steps to set up your NodeMCU ready to use for controlling hardware devices online as shown below.

1) Open your Arduino ID

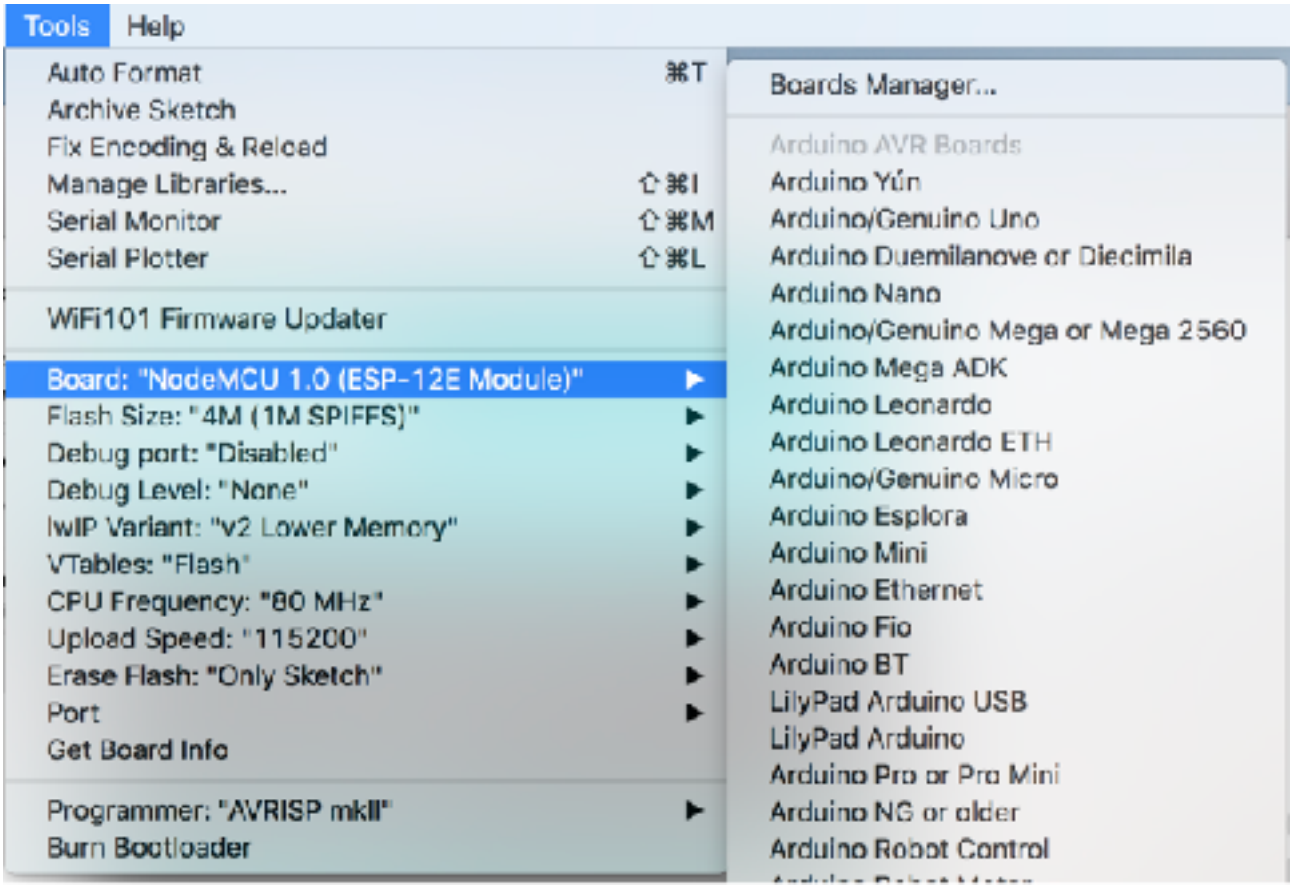
2) For Windows, Go on files and click to open " Preferences… ".
For MacOS, Click on "Arduino IDE", and select " Preferences.. "



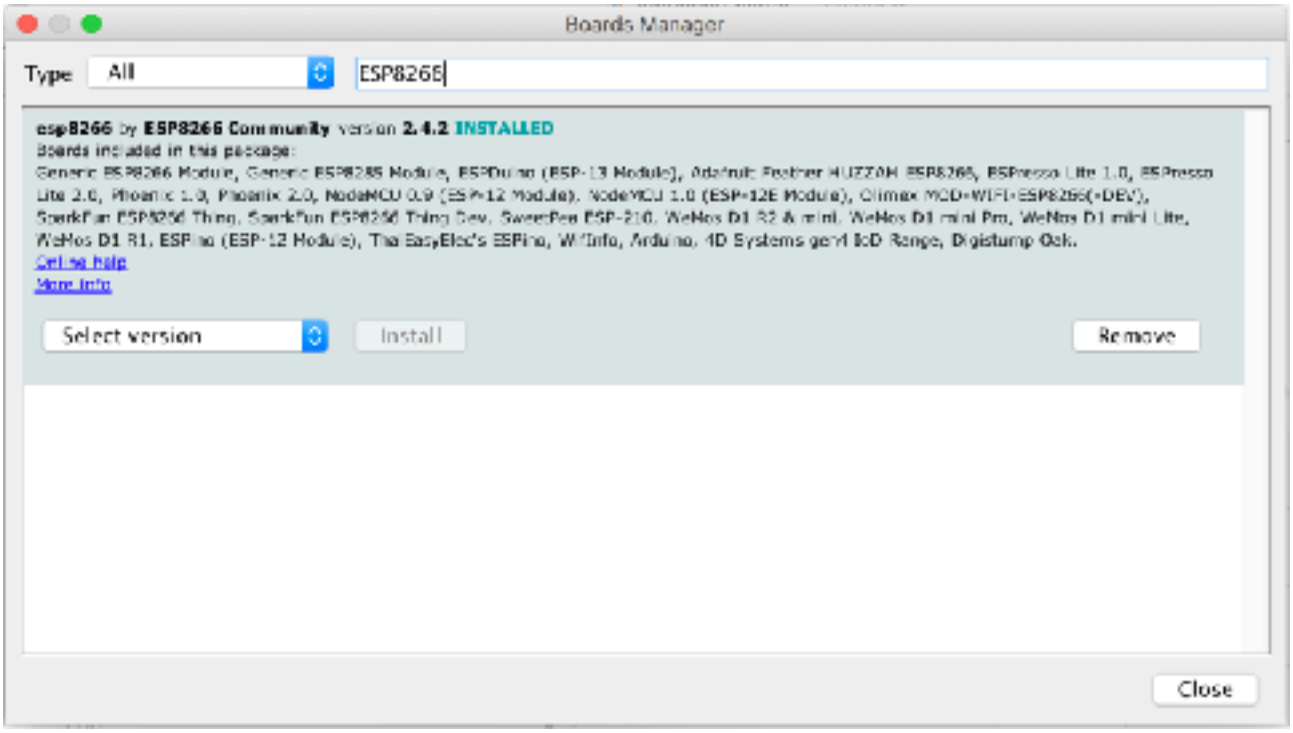
3) Choose your Sketchbook location, then copy the link "http://arduino.esp8266.com/stable/package_esp8266com_index.json" and paste to Additional Boards Manager URLS, and click OK



4) After completing the above steps , go to Tools and board, and then select board Manager



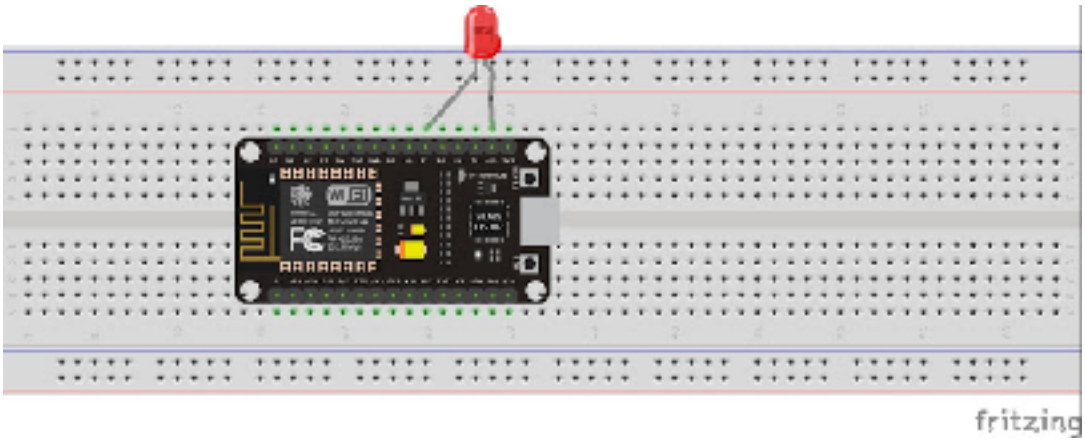
5) Type in ESP8266 in the searching edit-text. Then click " Install " on esp8266 by ESP8266 Community in order to allow Arduino IDE to recognise NodeMCU. After that, go back to step 4 and Select NodeMCU 1.0 (ESP-12E Module)



SETUP NODEMCU on Arduino IDE - Testing

After installing ESP8266 successfully, test that you can upload the program to your Arduino device successfully as follow.

1) Try to connect the circuit as shown below.



For this example I have used NodeMCU Nodemcu and if you are using any other vendor wifi chips or generic wifi module please check with the Nodemcu Pin mapping which is very essential to make things works.

We are using D7 pin for this example. I uploaded the basic blink program that comes with the examples program in the Arduino IDE which is connected with 13 pin of Arduino. The 13th pin is mapped into D7 pin of NodeMCU.

2) GO to board and select the type of Nodemcu you are using. and select the correct COM port to run the program on your Nodemcu device.

3) Upload the following program and see the results.

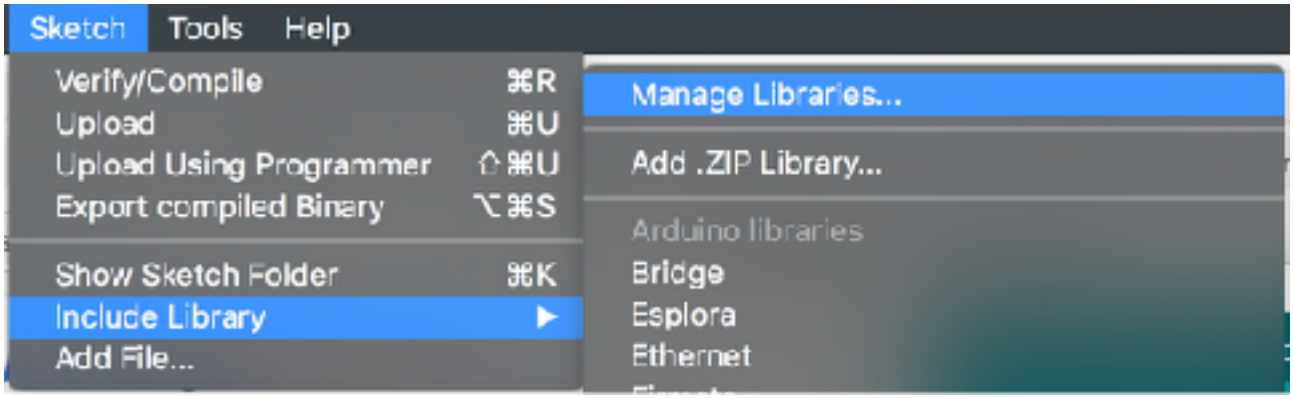
```
void setup() {  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
} // the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

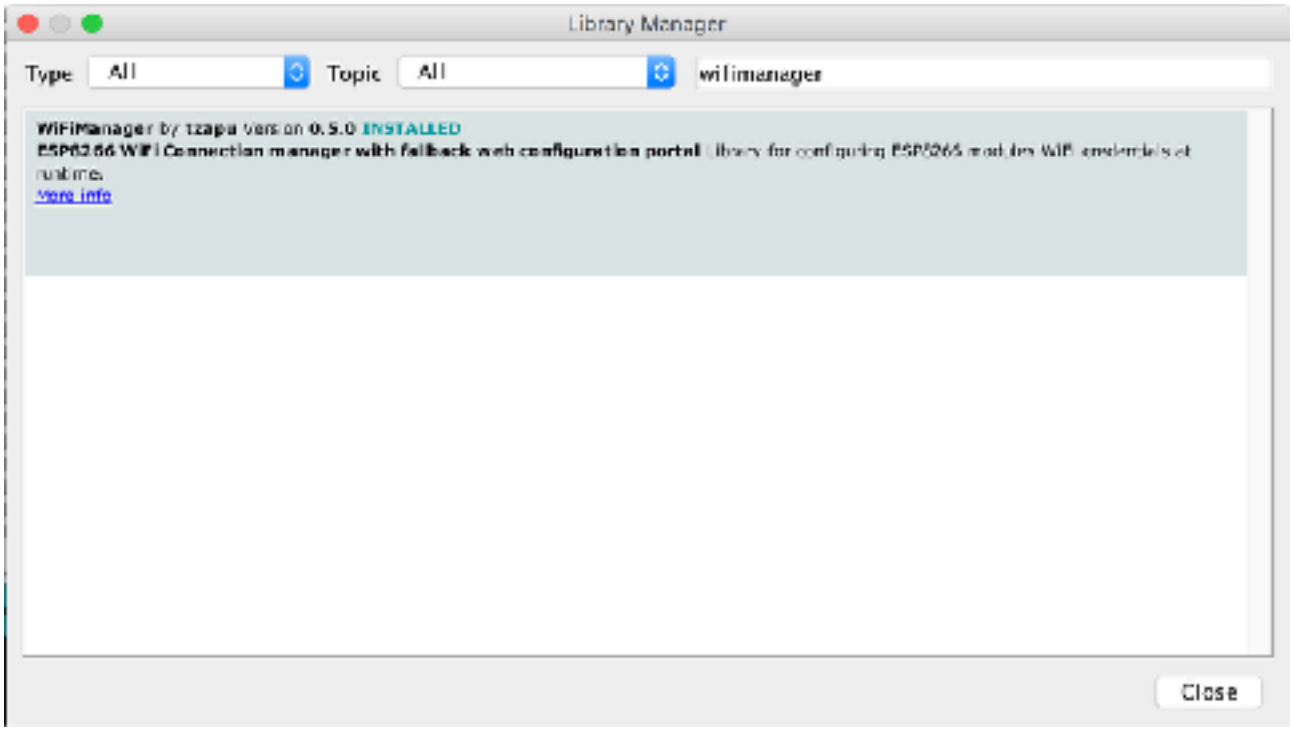
Note: If you only have NodeMCU with yourself, Change the first parameter of pinMode, and digitalWrite as “LED_BUILTIN”

Connect NodeMCU to WiFi

Even though, Arduino UNO is capable of controlling hardware devices, but the lack of connection to internet has become its huge limitations. To eliminate such a problem, ESP8266 or NodeMCU has its unique capability to connect hardware devices online in exchange of having a lower memory space. The internet connection can be created successfully with few steps.

1) Include the WiFi Library - Include the WiFi library by navigating to the Sketch > Include Library > ESP8266WiFi and WiFiManager by tzapu. This library is pre-installed when you set up the ESP8266 add on in your IDE.





2) Please try the following source code below. The “WIFI_SSID” is the WiFi name you would like your NodeMCU to access. The “ WIFI_PASSWORD ” is the password of your WiFi.

You can access to the internet WiFi using the command “ WiFi.begin(WIFI_SSID, WIFI_PASSWORD)”. After this line, you must set the condition to wait for internet connection between NodeMCU and its target WiFi. The “WiFi.status() ” represents for current status of the WiFi and the condition “WL_CONNECTED” stands for wireless connected.

This function returns following codes to describe what is going on with Wi-Fi connection:

- 0 : `WL_IDLE_STATUS` when Wi-Fi is in process of changing between statuses
- 1 : `WL_NO_SSID_AVAIL`in case configured SSID cannot be reached
- 3 : `WL_CONNECTED` after successful connection is established
- 4 : `WL_CONNECT_FAILED` if password is incorrect
- 6 : `WL_DISCONNECTED` if module is not configured in station mode

Note: **If your SSID or password is wrong the serial monitor will only print dot.** And it is inside a infinite loop so it won’t stop until you disconnect or re-upload new correct credentials.

```
#include <ESP8266WiFi.h>

#define WIFI_SSID "name_of_SSID"
#define WIFI_PASSWORD "password_here"

void setup() {
   Serial.begin(9600);
   WiFi.begin( WIFI_SSID,  WIFI_PASSWORD) ;

    Serial.print("connecting");

    while (WiFi.status() != WL_CONNECTED ) {
        Serial.print(".");
        delay(500);
    }

     Serial.println();
     Serial.print("connected");
}

void loop () {
    // Your code that you want to do after connecting

}
```

3) Get your Internet Protocol address(IP Address) in order to allow human to interact with hardware the via the web server. You can obtain the IP address using the command “Serial.println(WiFi.localIP());”

4) Upload the source code and copy your IP address and Paste to your web browser.

Connect NodeMCU to WiFi - SmartConfig

Alternatively, SmartConfig is a more efficient way to connect your NodeMCU to WiFi. Please check out the following source code

```
#include <ESP8266WiFi.h>
#include <DNSServer.h>            //Local DNS Server used for redirecting all requests to the configuration portal
#include <ESP8266WebServer.h>     //Local WebServer used to serve the configuration portal
#include <WiFiManager.h>          //https://github.com/tzapu/WiFiManager WiFi Configuration Magic

const char* ssid = "Bi GUA 5GHz";
//warakorn jetlohasiri's iPhone
const char* password = "naionaia";
int LEDPin13 = LED_BUILTIN;
WiFiManager wifiManager;


WiFiServer server(8888);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  delay(10);
  pinMode(LEDPin13, OUTPUT);
  digitalWrite(LEDPin13, LOW);

  //Connect to WiFi network
  wifiManager.autoConnect("ESP8266 #1");

  //Start the server
  server.begin();
  Serial.println("Server started");
}
```

From the following source code, you have to include the following libraries to make SmartConfig works

```
#include <ESP8266WiFi.h>

#include <DNSServer.h>
#include <ESP8266WebServer.h>
#include <WiFiManager.h>
```

We need to initialise library by adding function

```
WiFiManager wifiManager;
```

After we initialise the library, we need to create a new access point for a computers/device connected in the same WiFi router / personal hotspot to connect to the access point using the following command. To intentionally make the NodeMCU/ ESP8266 access point unsecured, you can leave AP-PASSWORD blank.

```
//first parameter is name of access point, second is the password
wifiManager.autoConnect("AP-NAME", "AP-PASSWORD");
```

Connect NodeMCU to WiFi - Testing

To be able to blink your LED Light on webpage through the web server with specific domain and port. Please try the following source and read the instruction carefully.

In this source code, you have to initialise the web server outside both setup and loop function to make the global WiFiServer variable with a specific port(The pathway for internet communication).

After the internet has been connected successfully, you have to start the server with the command “Server.begin()”

Notice that the variable WiFiClient is initialised inside the loop function in order to accept a client in that specific server domain. You can get the Client, it can be done by the command “ server.available() ”

In this scenario we provide 2 operations that our NodeMCU is capable to do which is to control the light. This means that there should be 2 requests including, / LED=ON and LED=OFF. These requests used to determine the NodeMCU operations to control all connected devices.

The following source code is to create the web layout with 2 buttons for user to control the light. This source code is not well implemented. For more information, please visit the following tutorial “ https://www.youtube.com/watch?v=YjwkVHva-gk "