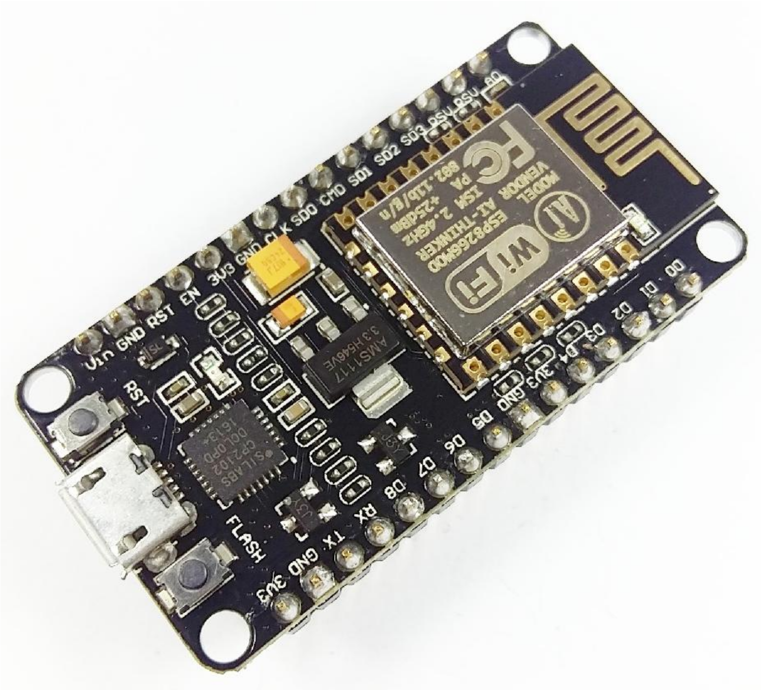
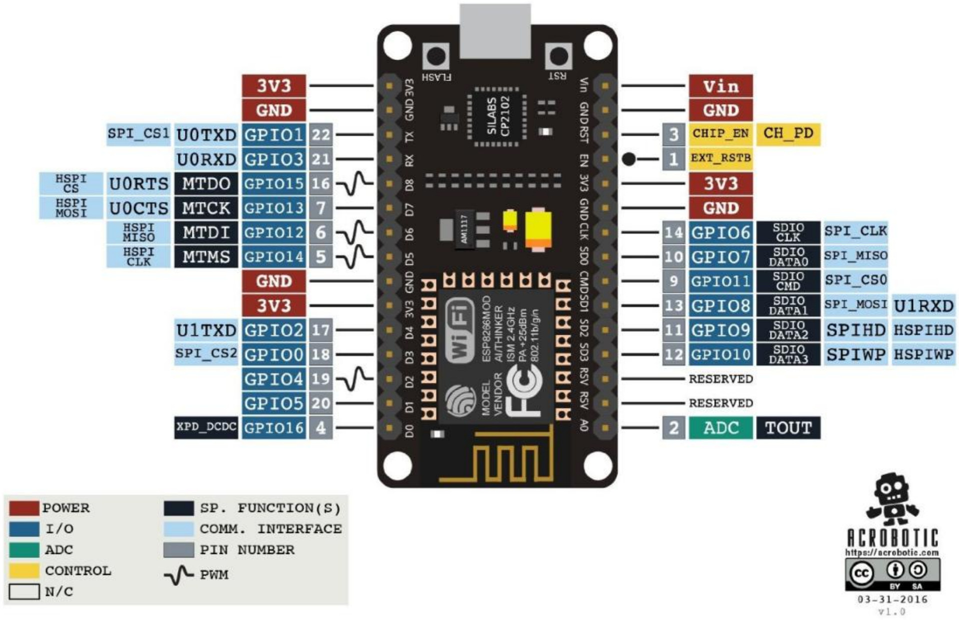
Getting Started with the ESP8266 WiFi Development Board


Vin GND RST EN 3V3 GND CLK SD0 CMD SD1 SD2 SD3 RSV RSV A0
D0 D1 D2 D3 D4 3V3 GND D5 D6 D7 D8 RX TX GND 3V3
RST
FLASH
WiFi
ESP8266MOD
AMS1117

The **ESP8266 WiFi Development Board** is an all-in-one microcontroller + WiFi platform that is very easy to use to create projects with WiFi and IoT (Internet of Things) applications. The board is based on the highly popular ESP8266 WiFi Module chip in an ESP-12 SMD footprint. This WiFi development board contains all the needed parts for the ESP12 to function such as a USB to serial chip to bootload and upload codes, 3.3V regulator and logic level converter circuit so you can immediately upload codes and connect your circuits!

This microcontroller board can easily be programmed using the Arduino IDE software for your Internet of Things projects. Let's dive in and test some example projects!

Hardware Overview



The ESP8266 WiFi Development board pins out a 3.3V and GND port and the Serial Communication pins Tx and RX although let's leave those pins for now. The board pins out the SPI/i2C pins and many general pupose I/O's.

There are 9 Digital Input & Output pins for general use and labeled as (GPIO16) **D0,** (GPIO5) **D1,** (GPIO4) **D2,** (GPIO0) **D3,** (GPIO2) **D4,** (GPIO14) **D5,** (GPIO12) **D6,** (GPIO13) **D7,** (GPIO15)**D8**.

The board has 1 Analog Input pin labeled as (ADC) **A0.**

Hardware Requirements

Parts needed for this guide. You can get these items and other information from the link provided

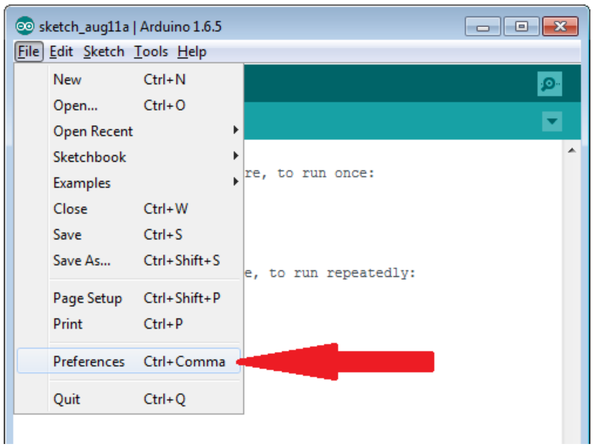
- ESP8266 WiFi Development Board – http://www.bitstoc.com/index.php?route=product/product&product_id=172
- USB A to micro B cable – http://www.bitstoc.com/index.php?route=product/product&product_id=165
- Breadboard – http://www.bitstoc.com/index.php?route=product/product&path=73&product_id=144
- Connecting Wires – http://www.bitstoc.com/index.php?route=product/product&path=73&product_id=63
- LED – http://www.bitstoc.com/index.php?route=product/product&path=77&product_id=147
- Resistor– http://www.bitstoc.com/index.php?route=product/product&path=73&product_id=145

Prepare the Software

The easiest way to program the WiFi ESP8266 Development board is through the Arduino IDE. This board is compatible to Arduino IDE and in this guide we will be using the Arduino IDE with version **1.6.5**. (**We use here version 1.6.5 of Arduino IDE** as tested, since other versions of the Arduino IDE have difficulties in running the programs). If you don't have yet this version of the Arduino IDE, go to this link to download the software. Find the download link that has the version **1.6.5**.

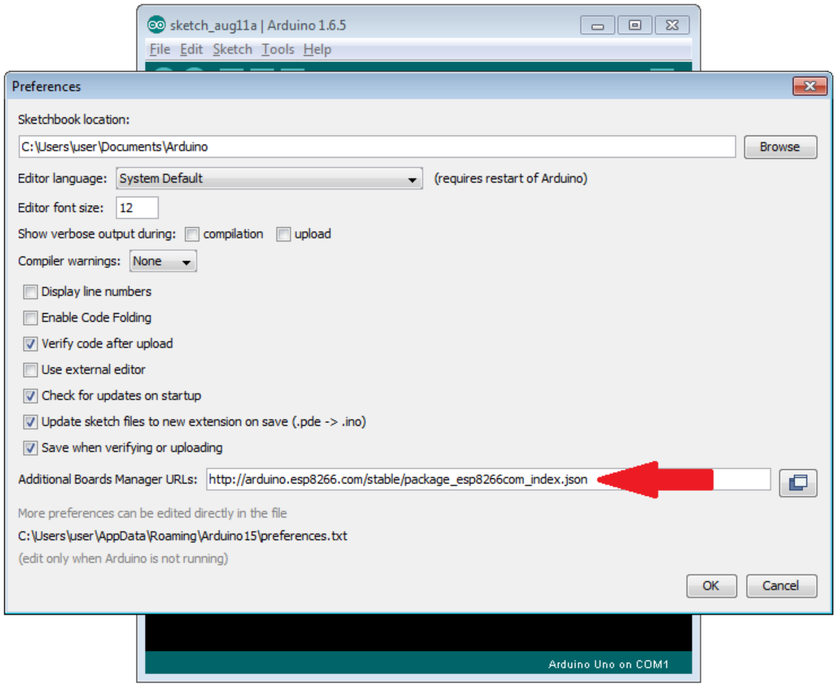
https://www.arduino.cc/en/Main/OldSoftwareReleases#previous

Open your **Arduino IDE**. Go to **File>Preferences**.



At the bottom part find the **Additional Boards Manager URLS:** and copy the link below and paste it.

---> **http://arduino.esp8266.com/stable/package_esp8266com_index.json**



Sketchbook location:
C:\Users\user\Documents\Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Show verbose output during: □ compilation □ upload

Compiler warnings: None

□ Display line numbers
□ Enable Code Folding
☑ Verify code after upload
□ Use external editor
☑ Check for updates on startup
☑ Update sketch files to new extension on save (.pde -> .ino)
☑ Save when verifying or uploading

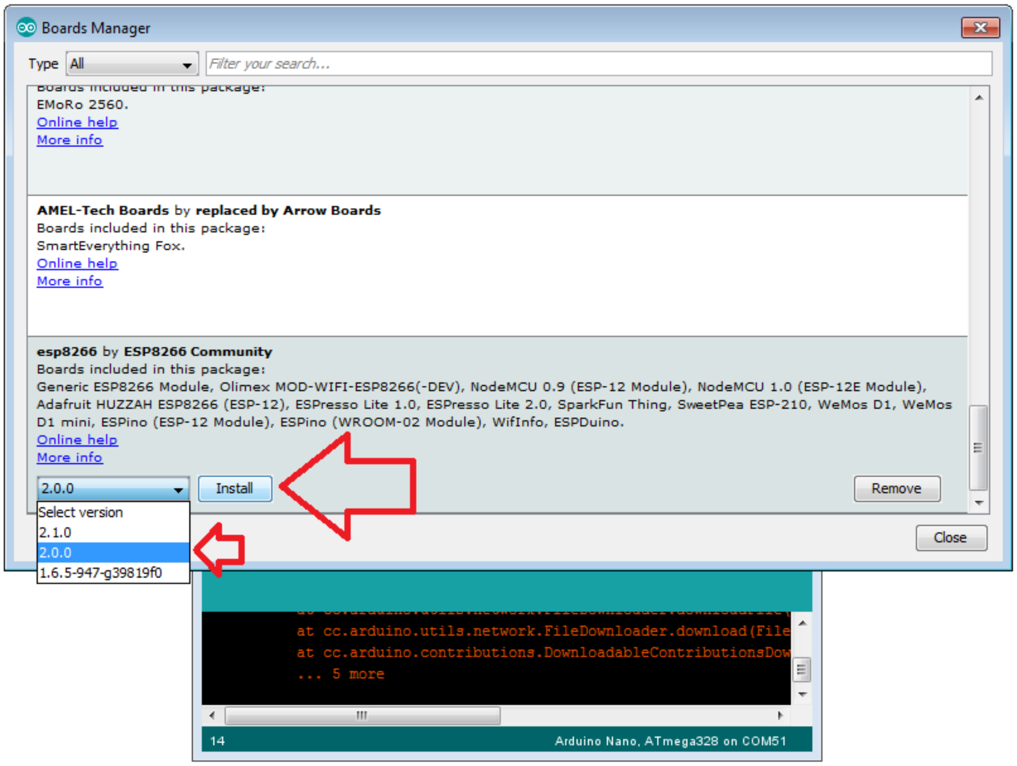
Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

More preferences can be edited directly in the file
C:\Users\user\AppData\Roaming\Arduino15\preferences.txt
(edit only when Arduino is not running)

Click **OK** to close the Preferences Tab

Now, go to **Tools>Board>Boards Manager…**

Scroll down below and you should see the **esp8266 by ESP8266 Community** column board installation Click on the dropdown arrow and choose **2.0.0** and then **Install**. Wait for the installation to finish and your esp8266 board will be listed in the compatible boards.



Boards included in this package:
EMoRo 2560.
Online help
More info

AMEL-Tech Boards by **replaced by Arrow Boards**
Boards included in this package:
SmartEverything Fox.
Online help
More info

esp8266 by **ESP8266 Community**
Boards included in this package:
Generic ESP8266 Module, Olimex MOD-WIFI-ESP8266(-DEV), NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Adafruit HUZZAH ESP8266 (ESP-12), ESPresso Lite 1.0, ESPresso Lite 2.0, SparkFun Thing, SweetPea ESP-210, WeMos D1, WeMos D1 mini, ESPino (ESP-12 Module), ESPino (WROOM-02 Module), WifInfo, ESPDuino.
Online help
More info

```
        at cc.arduino.utils.network.FileDownloader.download(File
        at cc.arduino.contributions.DownloadableContributionsDow
        ... 5 more
```

14 Arduino Nano, ATmega328 on COM51

Install the USB driver for the CP2102 USB-to-Serial Chip

Next, we need to install the software driver for the built in **USB to Serial chip based on CP2102**. To do this, go to the link below and follow the instructions. Don't worry it just takes a minute to install!

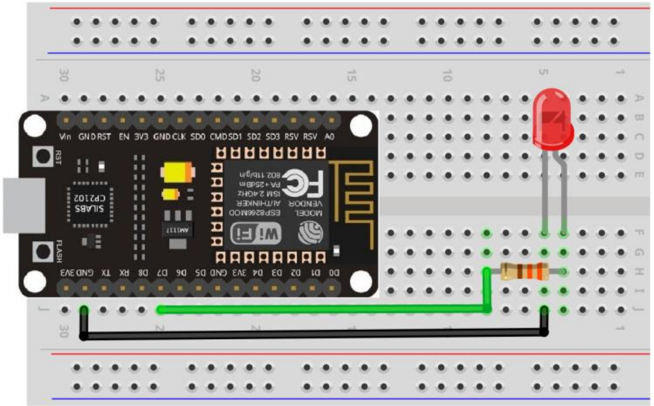
https://drive.google.com/drive/folders/0B5RP5NsysFygOFljUmxfbjE1Q2M

After installing the CP2102 USB to Serial driver, the board will be recognized by the Arduino IDE and we are now ready to program the ESP8266 Development board!

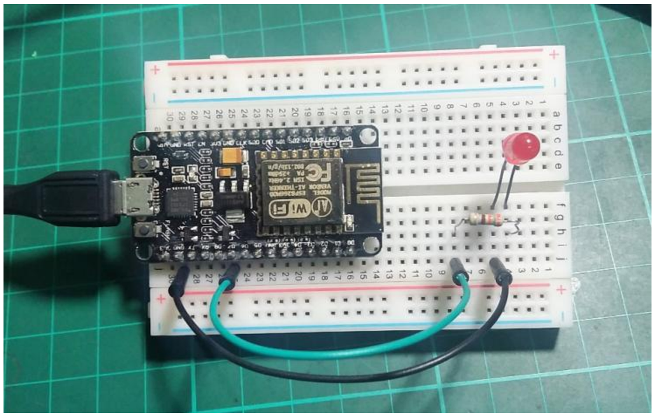
Test Project: Blink an LED

Wiring Connection

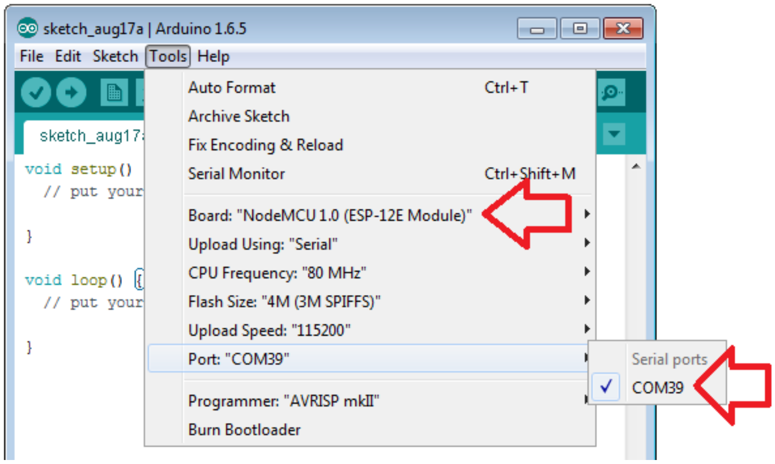
Construct the circuit below. Note that the long leg of the LED is on the right side and connected to the 330ohm resistor. The short left leg of the LED is connected to GND of the board.


Vin GND RST EN 3V3 GND CLK SDO CMD SD1 SD2 SD3 RSV RSV A0
3V3 GND TX RX D8 D7 D6 D5 GND 3V3 D4 D3 D2 D1 D0
RST
FLASH
SILABS CP2102
WiFi
30
25
20
15
10
5
1

Your circuit should look like this below. Then connect your ESP8266 WiFi Development board to the computer using the USB Cable - **USB A to micro B.**



Now in your Arduino IDE go to **Tools>Board** and click on the **NodeMCU 1.0 (ESP-12E Module).** Then go to **Tools>Port** and select the **COMxx** number where your CP210x USB to Serial chip is installed. In this case our chip is installed in COM39.



```
void setup() {
  // put your

}

void loop() {
  // put your

}
```

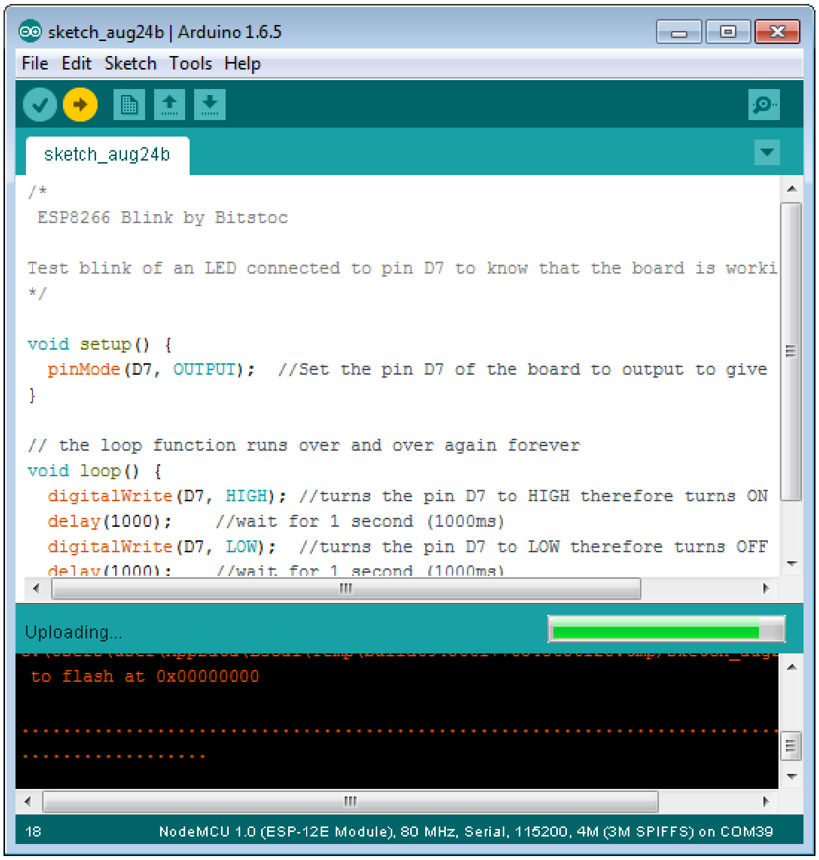
Board: "NodeMCU 1.0 (ESP-12E Module)"
Upload Using: "Serial"
CPU Frequency: "80 MHz"
Flash Size: "4M (3M SPIFFS)"
Upload Speed: "115200"
Port: "COM39"
Programmer: "AVRISP mkII"
✓ COM39

Upload the Code

Create a new Arduino sketch by going to **File>New** . Copy the code below and paste it in the Arduino IDE. Compile the code and save it to your desired location.

```
/*
 ESP8266 Blink by Bitstoc

Test blink of an LED connected to pin D7 to know that the board is working
*/

void setup() {
  pinMode(13, OUTPUT);    //Set the pin D7 of the board to output to give out a signal.
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);    //turns the pin D7 to HIGH therefore turns ON the LED.
  delay(1000);               //wait for 1 second (1000ms)
  digitalWrite(13, LOW);     //turns the pin D7 to LOW therefore turns OFF the LED.
  delay(1000);               //wait for 1 second (1000ms)
}
```

Upload the code to the ESP8266 WiFi Development board by clicking on the arrow Upload button.



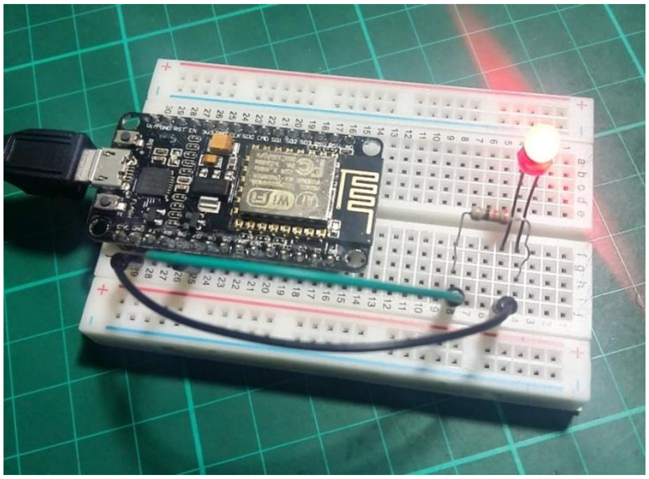
```
/*
 ESP8266 Blink by Bitstoc

Test blink of an LED connected to pin D7 to know that the board is worki
*/

void setup() {
  pinMode(D7, OUTPUT);  //Set the pin D7 of the board to output to give
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(D7, HIGH); //turns the pin D7 to HIGH therefore turns ON
  delay(1000);     //wait for 1 second (1000ms)
  digitalWrite(D7, LOW);  //turns the pin D7 to LOW therefore turns OFF
  delay(1000);     //wait for 1 second (1000ms)
```

Uploading...

```
 to flash at 0x00000000

...........................................................................
..................
```

18 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Serial, 115200, 4M (3M SPIFFS) on COM39

After uploading the code, you should see that the red LED will turn ON and OFF every second. Yey! Your board is working and we are good to go for the next tutorials!



Now that we have finished testing the board with a test Blink…

Head on to the next page and we will perform a new project, blinking an LED over WiFi!

Test Project: Blink an LED over WiFi (WiFi Webserver)

Using the same circuit from the previous example, we will control the blink of the LED wirelessly through WiFi connection. To do this, simply copy the code below and paste it in your Arduino IDE. You can also create a new project file by opening up a new sketch from **File>New.**

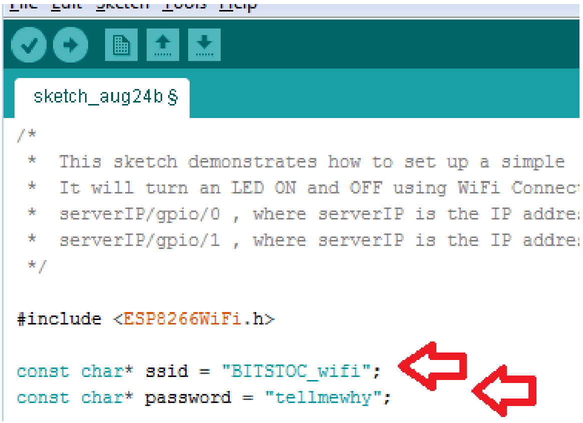
```
/*
 *  This sketch demonstrates how to set up a simple (local) Web server.
 *  It will turn an LED ON and OFF using WiFi Connection with commands:
 *  serverIP/gpio/0 , where serverIP is the IP address of the ESP8266 board and 0 is OFF
 *  serverIP/gpio/1 , where serverIP is the IP address of the ESP8266 board and 1 is ON
 */

#include <ESP8266WiFi.h>

const char* ssid = "XXXXX";
const char* password = "XXXXX";

// Create an instance of the server
// specify the port to listen on as an argument
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);

  // prepare GPIO2
  pinMode(13, OUTPUT);
  digitalWrite(13, 0);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

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

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

  // Print the IP address
  Serial.println(WiFi.localIP());
}
```

code continuation.......

```
void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // Wait until the client sends some data
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }

  // Read the first line of the request
  String req = client.readStringUntil('\r');
  Serial.println(req);
  client.flush();

  // Match the request
  int val;
  if (req.indexOf("/gpio/0") != -1)
    val = 0;
  else if (req.indexOf("/gpio/1") != -1)
    val = 1;
  else {
    Serial.println("invalid request");
    client.stop();
    return;
  }

  // Set GPIO2 according to the request
  digitalWrite(13, val);
  client.flush();

  // Prepare the response
  String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE HTML>\r\n<html>\r\nGPIO is now ";
  s += (val)?"high":"low";
  s += "</html>\n";

  // Send the response to the client
  client.print(s);

  // buttons to turn the LED ON (/gpio/1) or OFF (/gpio/0)
  client.println("<br><br>");
  client.println("Click <a href=\"/gpio/1\">here</a> turn the LED on pin 13 ON<br>");
  client.println("Click <a href=\"/gpio/0\">here</a> turn the LED on pin 13 OFF<br>");
  client.println("</html>");

  delay(1);
  Serial.println("Client disonnected");
}
```

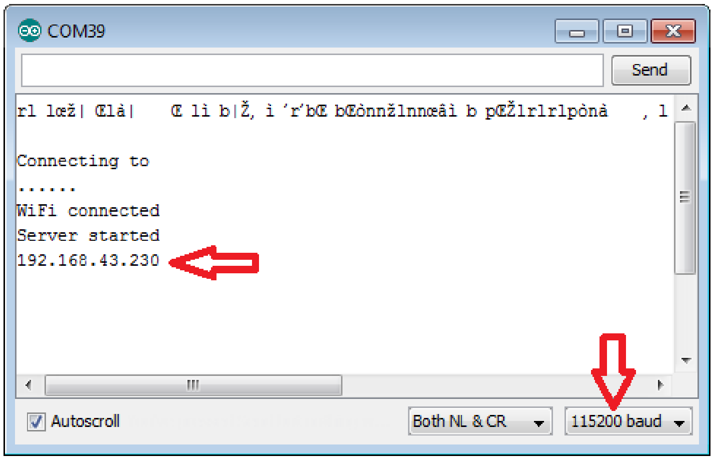
Before you Compile and Upload the code above change the credentials of these line below to your own **WiFi Name and Password**.

const char* ssid = “(your_wifi_name)”;
const char* password = “(your_wifi_password)”;



```
/*
 *  This sketch demonstrates how to set up a simple
 *  It will turn an LED ON and OFF using WiFi Connec
 *  serverIP/gpio/0 , where serverIP is the IP addre
 *  serverIP/gpio/1 , where serverIP is the IP addre
 */

#include <ESP8266WiFi.h>

const char* ssid = "BITSTOC_wifi";
const char* password = "tellmewhy";
```

After setting up your WiFi name and password in the code, **Compile and Upload the code** to the board. Now open the Serial Monitor from the Arduino IDE by clicking the magnifying glass button at the upper right corner. When you open the Serial Monitor set the **baud to 115200**. Copy the IP Address shown in your Serial monitor in this case its **192.168.43.230**. You should see an output like the one below.



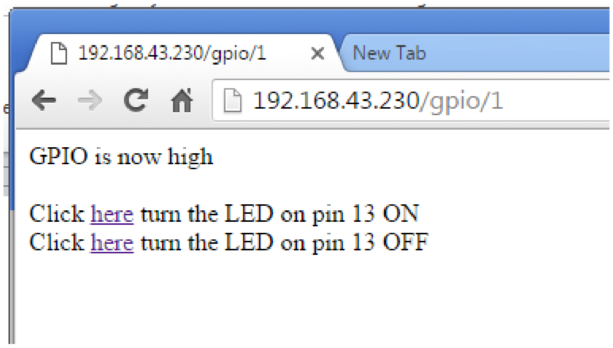
```
rl lœž| Œlà|    Œ lì b|Ž, ì ʼrʼbŒ bŒònnžlnnœâi b pŒŽlrlrlpònà   , l

Connecting to
......
WiFi connected
Server started
192.168.43.230
```

☑ Autoscroll | Both NL & CR | 115200 baud

Now get a wireless-capable device (laptop/smartphone/tablet) and connect to the same WiFi Router access point where your ESP8266 WiFi Dev Board is also connecting.

From your wireless device, open your favorite web browser and type in the format: “IP_Address”/gpio/1, where in this case the IP_Address is **192.168.43.230**.

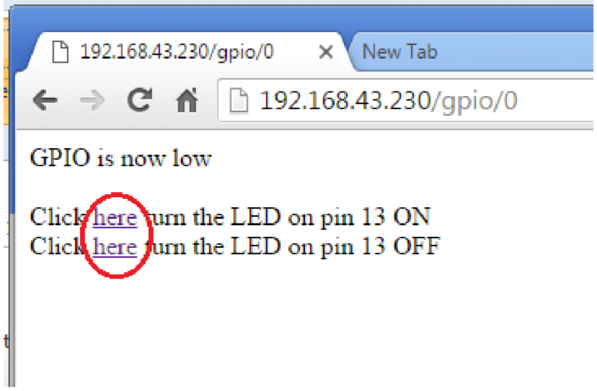
So in the browser we will type in **192.168.43.230/gpio/1** then hit **Enter**.
With this URL you will see that the LED connected to pin **D7 (GPIO13) will turn ON**!
Your browser will also show like below.



GPIO is now high

Click here turn the LED on pin 13 ON
Click here turn the LED on pin 13 OFF

Next, type in **192.168.43.230/gpio/0** and you will see that **D7 (GPIO13) will turn OFF!.**
You can also click the **"Click here"** text to turn the LED on and OFF.



GPIO is now low

Click here turn the LED on pin 13 ON
Click here turn the LED on pin 13 OFF

Hoooray! You can blink the LED using a WiFi connection! In the next example, we will now connect to the internet and get some data from a webpage. Head on!!!

Test Project: Ping a live website from the Internet

For this example we will be connecting to the Internet! And of course through WiFi! We will get a query on a live website on the internet to know that our ESP8266 WiFi Dev Board is connected and can send or receive data over the internet. Simply copy the code below and paste it in your Arduino IDE. Complie and Upload the code to the board just like from the previous example.

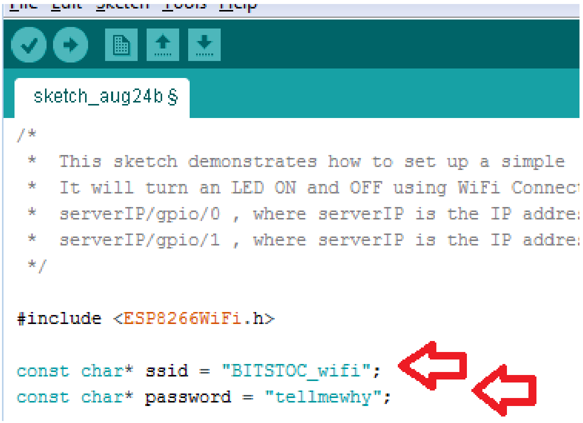
```
/*
 *  Simple HTTP get webclient test
 */

#include <ESP8266WiFi.h>

const char* ssid     = "XXXXX";
const char* password = "XXXXX";

const char* host = "www.example.com";

void setup() {
  Serial.begin(115200);
  delay(100);

  // We start by connecting to a WiFi network

  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

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

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

int value = 0;

void loop() {
  delay(5000);
  ++value;

  Serial.print("connecting to ");
  Serial.println(host);
```

code continuation.......

```
  // Use WiFiClient class to create TCP connections
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
  }

  // We now create a URI for the request
  String url = "/index.html";
  Serial.print("Requesting URL: ");
  Serial.println(url);

  // This will send the request to the server
  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
               "Host: " + host + "\r\n" +
               "Connection: close\r\n\r\n");
  delay(500);

  // Read all the lines of the reply from server and print them to Serial
  while (client.available()) {
    String line = client.readStringUntil('\r');
    Serial.print(line);
  }

  Serial.println();
  Serial.println("closing connection");
}
```

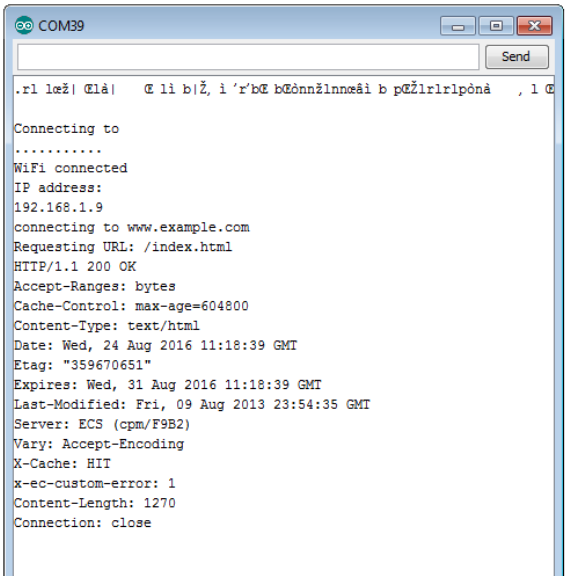
Before you Compile and Upload the code above remember to change the credentials of these line below to your own
<u>**WiFi Name and Password**</u>.

const char* ssid = “(your_wifi_name)”;
const char* password = “(your_wifi_password)”;

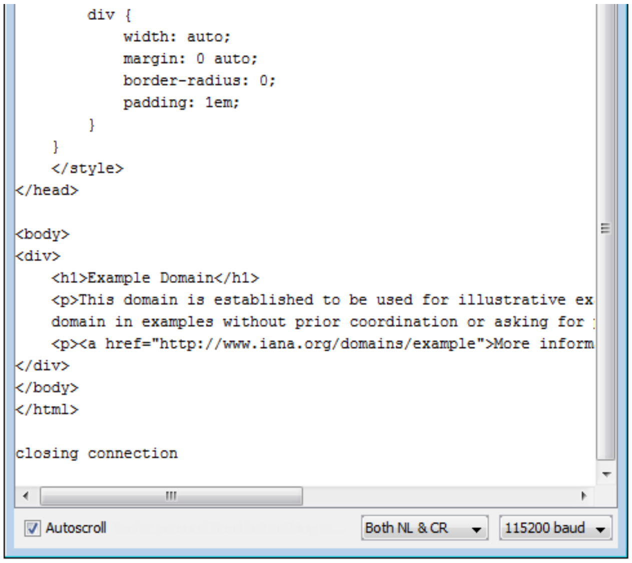


```
/*
 *  This sketch demonstrates how to set up a simple
 *  It will turn an LED ON and OFF using WiFi Connec
 *  serverIP/gpio/0 , where serverIP is the IP addre:
 *  serverIP/gpio/1 , where serverIP is the IP addre:
 */

#include <ESP8266WiFi.h>

const char* ssid = "BITSTOC_wifi";
const char* password = "tellmewhy";
```

After setting up your WiFi name and password in the code, **Compile and Upload the code** to the board. Now open the Serial Monitor from the Arduino IDE by clicking the magnifying glass button at the upper right corner and set the **baud to 115200**. You should see an output like the one below.



```
.rl lœž| Œlà|    Œ lì b|Ž, ì ’r’bŒ bŒònnžlnnœâì b pŒŽlrlrlpònà    , l Œ

Connecting to
............
WiFi connected
IP address:
192.168.1.9
connecting to www.example.com
Requesting URL: /index.html
HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: max-age=604800
Content-Type: text/html
Date: Wed, 24 Aug 2016 11:18:39 GMT
Etag: "359670651"
Expires: Wed, 31 Aug 2016 11:18:39 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (cpm/F9B2)
Vary: Accept-Encoding
X-Cache: HIT
x-ec-custom-error: 1
Content-Length: 1270
Connection: close
```



```
        div {
            width: auto;
            margin: 0 auto;
            border-radius: 0;
            padding: 1em;
        }
    }
    </style>
</head>

<body>
<div>
    <h1>Example Domain</h1>
    <p>This domain is established to be used for illustrative ex
    domain in examples without prior coordination or asking for 
    <p><a href="http://www.iana.org/domains/example">More inform
</div>
</body>
</html>

closing connection
```

☑ Autoscroll   Both NL & CR   115200 baud

In our code, we query a request from the website **www.example.com** and display the text from the website page **www.example.com/index.html.**

Congratulations! Your ESP8266 WiFi Development board is connected to the internet. We are ready to make IoT projects!!!

Going Further

This experiment is just the starting point to get started with the module. Check out the other links from the product page for other example projects and Tutorials and stay tune for more resources that we will add!

Download links for this project

Parts needed for this guide. You can get these items and other information from the link provided

- ESP8266 WiFi Development Board – *http://www.bitstoc.com/index.php?route=product/product&product_id=172*
- USB A to micro B cable – *http://www.bitstoc.com/index.php?route=product/product&product_id=165*
- Breadboard – *http://www.bitstoc.com/index.php?route=product/product&path=73&product_id=144*
- Connecting Wires – *http://www.bitstoc.com/index.php?route=product/product&path=73&product_id=63*
- LED – *http://www.bitstoc.com/index.php?route=product/product&path=77&product_id=147*
- Resistor– *http://www.bitstoc.com/index.php?route=product/product&path=73&product_id=145*

Arduino IDE software 1.6.5
https://www.arduino.cc/en/Main/OldSoftwareReleases#previous

Driver Installer for the USB-to-Serial Chip
https://drive.google.com/drive/folders/0B5RP5NsysFygOFljUmxfbjE1Q2M

ESP8266 WiFi Development Board product page
http://www.bitstoc.com/index.php?route=product/product&product_id=172