

Configuring Your ESP8266 Chip

We are now going to configure your ESP8266 chip using the Arduino IDE. This is a great way to use the chip as you will be able to program it using the well-known Arduino IDE, and also re-use several existing Arduino libraries.

If this is not done yet, install the latest version of the Arduino IDE. You can get it from:

<http://www.arduino.cc/en/main/software>

Then, you need to take a few steps to be able to configure the ESP8266 with the Arduino IDE:

- Start the Arduino IDE and open the Preferences window.
- Enter the following URL: `http://arduino.esp8266.com/package_esp8266com_index.json` into *Additional Board Manager URLs* field.
- Open Boards Manager from Tools > Board menu and install the *esp8266* platform.

Now, we are going to check that the Arduino IDE is correctly working, and connect your chip to your local WiFi network.

To do so, we need to write the code first, and then upload it to the board. The code is will be quite simple: we just want to connect to the local WiFi network, and print the IP address of the board. This is the code to connect to the network:

```
1. // Import required libraries
2. #include "ESP8266WiFi.h"
3.
4. // WiFi parameters
5. const char* ssid = "your_wifi_name";
6. const char* password = "your_wifi_password";
7.
8. void setup(void)
9. {
10.     // Start Serial
11.     Serial.begin(115200);
12.
13.     // Connect to WiFi
14.     WiFi.begin(ssid, password);
```

```

15.     while (WiFi.status() != WL_CONNECTED) {
16.         delay(500);
17.         Serial.print(".");
18.     }
19.     Serial.println("");
20.     Serial.println("WiFi connected");
21.
22.     // Print the IP address
23.     Serial.println(WiFi.localIP());
24.
25. }
26.
27. void loop() {
28.
29. }

```

You can simply copy the lines of code above, and copy them into the ESP8266 Arduino IDE that you downloaded before. Of course, put your own WiFi name & password in the code. Save this file with a name of your choice.

Now, also go in Tools>Boards, and select “Generic ESP8266 Module”. Also select the correct Serial port that corresponds to the FTDI converter your are using.

After that, we need to put the board in bootloader mode, so we can program it. To do so, connect the pin GPIO 0 to the ground, via the cable we plugged into GPIO 0 before. Then, power cycle the board but switching the power supply off & then on again.

Now, upload the code to the board, and open the Serial monitor when this is done. Also set the Serial monitor speed to 115200. Now, disconnect the cable between GPIO 0 and GND, and power cycle the board again. You should see the following message:

```

WiFi connected
192.168.1.103

```

If you can see this message and an IP, congratulations, your board is now connected to your WiFi network! You are now ready to build your first projects using the ESP8266 chip & this modified Arduino IDE.

How to Go Further

You now have a completely usable ESP8266 module, so basically what you can do next only depends on your imagination! You can for example use this chip to build a WiFi motion sensor, to control a relay remotely, and also to send data to a remote cloud platform.

I will post articles about applications of this board in the coming weeks, but in the meantime don't hesitate to experiment with this amazing little WiFi board and share your projects in the comments!