

Débuter avec le module WiFi ESP8266

Par TonesB  - f-leb (traducteur)

Date de publication : 17 février 2018

Dernière mise à jour : 10 juin 2018

L'ESP8266 est un circuit intégré avec un microcontrôleur permettant la connexion en WiFi. Les modules intégrant ce circuit sont très utilisés pour contrôler des périphériques par Internet. L'ESP8266 est livré avec un firmware préinstallé vous permettant d'en prendre le contrôle à l'aide de « commandes AT » standards pouvant provenir d'une carte Arduino avec qui il peut communiquer par liaison série. Mais toute la souplesse et la puissance de ce module résident dans le fait que vous pouvez aussi y développer et flasher votre propre code, rendant ainsi le module entièrement autonome.

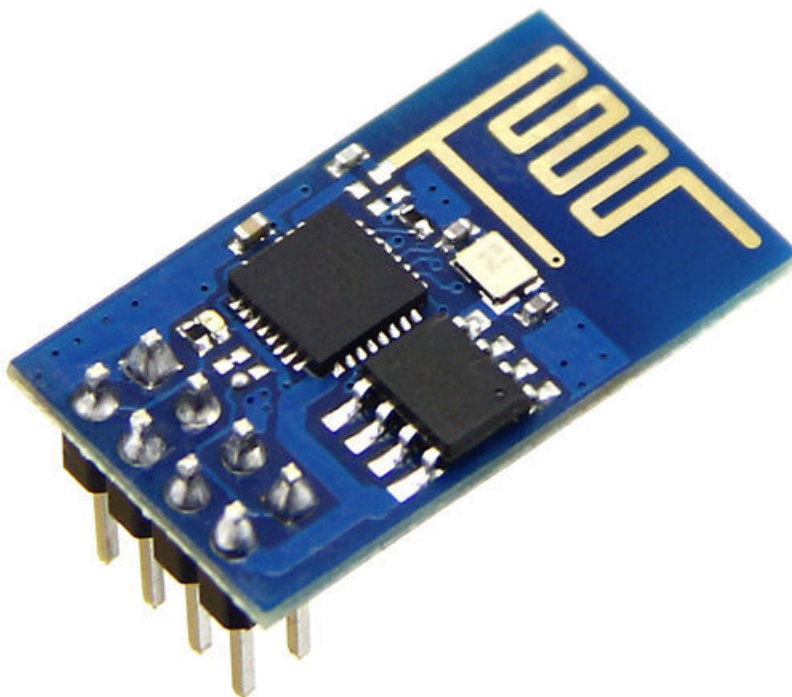
Pour l'Internet des Objets (IdO), ce petit module (25x14 mm) vous rendra de grands services pour quelques euros à déboursier seulement.

Commentez

I - Le module WiFi ESP8266 et l'ESP-01 en bref.....	3
II - Matériel nécessaire.....	5
III - Schéma du montage.....	7
IV - Le brochage de l'ESP-01.....	8
V - Utiliser une carte Arduino comme programmeur série.....	9
VI - Programmer avec l'EDI Arduino.....	9
VI-A - Installation du gestionnaire.....	9
VI-B - Communiquer avec l'ESP8266 à l'aide de commandes AT.....	13
VI-C - Le code.....	14
VI-D - Téléverser le code dans l'ESP8266.....	16
VI-E - L'ESP8266, serveur web.....	17
VII - Sitographie sur l'ESP8266.....	19
VIII - Notes de la rédaction Developpez.com.....	19

I - Le module WiFi ESP8266 et l'ESP-01 en bref

L'aventure ESP8266 a démarré en 2014 et le module a gagné rapidement en popularité. Il existe maintenant **de nombreuses cartes à base de processeur ESP8266**. Leurs caractéristiques diffèrent notamment en termes de nombres de ports d'entrées-sorties ou de taille de la mémoire flash.



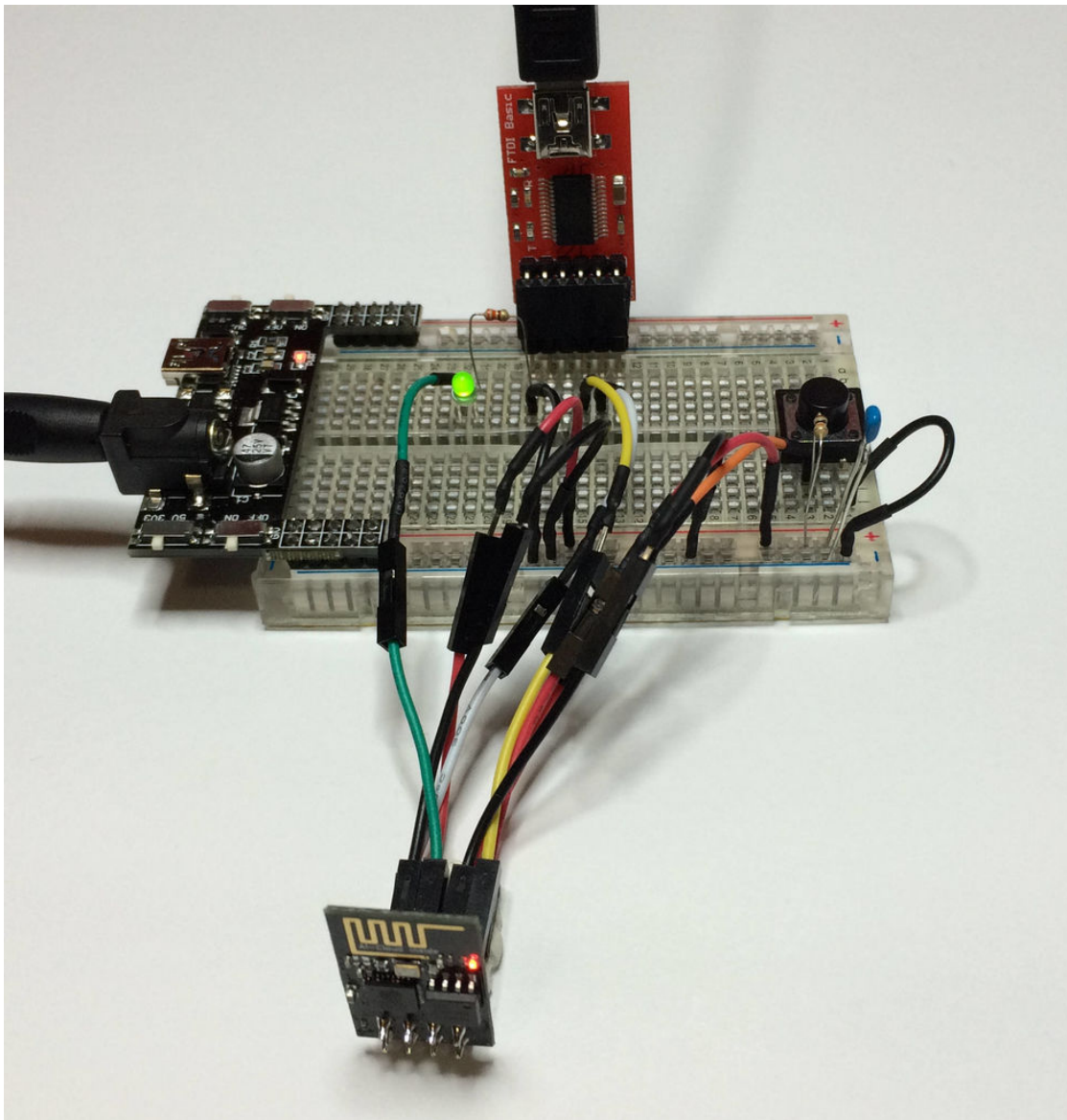
Module ESP-01 à base de processeur ESP8266

Ces circuits ont beaucoup évolué depuis et de nombreuses informations peuvent être trouvées sur le web pour la plus grande satisfaction des utilisateurs. Malheureusement, beaucoup de ces informations sont maintenant périmées ou tout simplement fausses.

Ainsi, il m'a fallu du temps avant d'obtenir un premier résultat exploitable ! D'autres auront eu plus de chance, mais faire fonctionner correctement ce module fut un vrai combat. À tel point que parfois, j'ai même cru avoir détruit le module avant de le faire revivre à force de persévérance dans mes recherches.

J'ai finalement dompté la bête qui s'avère plus simple d'utilisation que je ne pouvais l'imaginer au départ. Après avoir rassemblé toutes les informations utiles, je vous propose d'en faire la synthèse dans ce tutoriel. En espérant qu'il vous permette de débiter rapidement et d'exploiter les fonctionnalités de base dont vous aurez besoin.

Dans ce tutoriel, le module utilisé est un ESP-01, mais l'approche décrite devrait être identique avec la plupart des modules à base de processeur ESP8266 (si ce n'est la totalité).



L'ESP8266 en bref :

- 1 Supporte les protocoles 802.11 b/g/n ;
- 2 Se connecte à votre routeur et fonctionne comme client ou comme point d'accès ou même les deux en même temps !
- 3 Peut fonctionner comme serveur web avec sa propre adresse IP ;
- 4 Le module ESP-01 comprend deux broches numériques qui peuvent être configurées en entrée ou en sortie (pour piloter des LED, des relais, etc.). Ces broches peuvent être utilisées pour générer un signal modulé en largeur d'impulsion (PWM). D'autres versions du module comportent davantage de broches d'entrées-sorties (l'ESP-12 par exemple), mais ils se programment tous de façon similaire ;
- 5 Possède une entrée analogique (ADC/TOUT), mais qui n'est pas câblée sur l'ESP-01 ;
- 6 Peut fonctionner en communiquant avec une carte Arduino ou être programmé pour fonctionner en toute autonomie ;
- 7 Dispose de nombreux outils et environnements de développement (EDI) pour le programmer ;
- 8 Et plein d'autres choses encore...

Dans la démonstration qui suit, l'ESP8266 est configuré en serveur web générant une page web accessible à l'utilisateur par liaison WiFi. Vous pourrez alors vous inspirer de cette démonstration pour développer votre propre projet domotique et contrôler vos objets connectés.

II - Matériel nécessaire

Si vous voulez programmer l'ESP8266 avec votre propre code, ce qu'on propose de faire dans ce tutoriel, vous n'aurez plus besoin du firmware d'origine. Votre code **est** le firmware. Pour revenir en arrière et flasher une version à jour du firmware dans l'ESP8266 : [🇬🇧 Loading new firmware onto an ESP8266](#).

On utilisera l'EDI Arduino comme outil de développement, mais vous n'êtes pas obligé de disposer d'une carte Arduino pour autant. L'ESP8266 est lui-même un microcontrôleur.

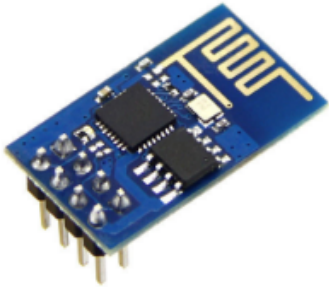
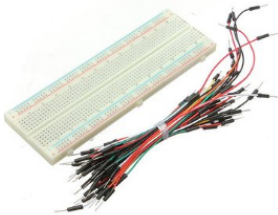
Après avoir téléversé votre propre code dans l'ESP8266, celui-ci ne répondra plus aux commandes AT, car votre code a remplacé le firmware d'origine. Développer son propre code avec l'aide de la bibliothèque ESP8266 de l'EDI Arduino vous offre de nombreuses possibilités. Voir la documentation : [🇬🇧 ESP8266WiFi library](#).

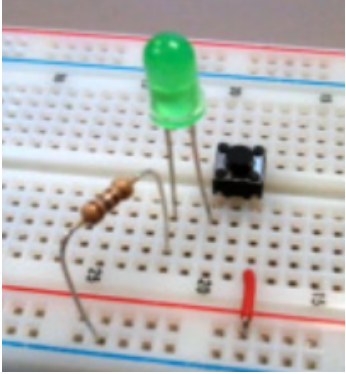

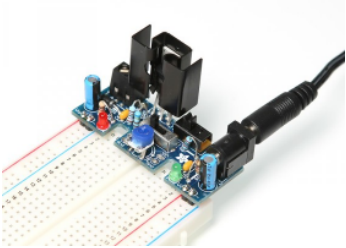
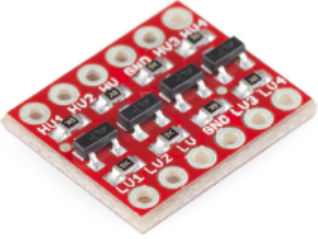



Vous pouvez restaurer le firmware d'origine de l'ESP8266 à tout moment et reprendre le contrôle avec le jeu de commandes AT si vous le souhaitez. Voir [🇬🇧 Loading new firmware onto an ESP8266](#).



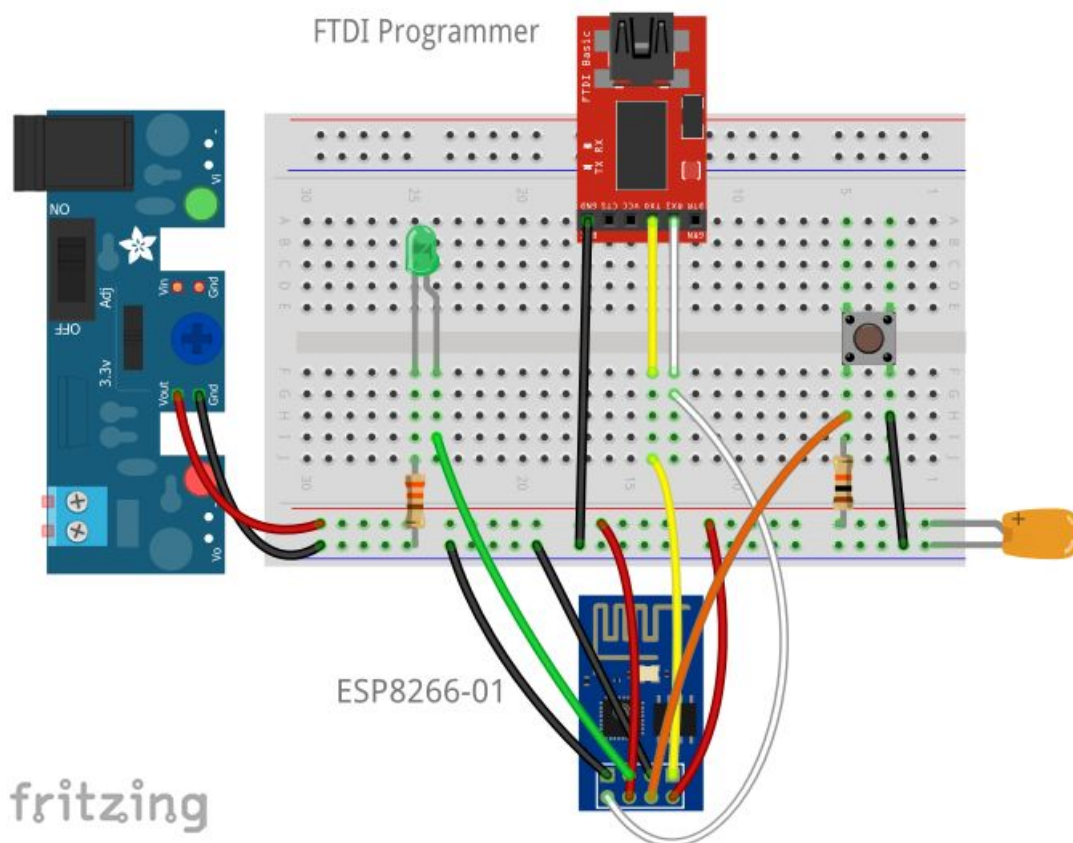
*Le module ESP8266 s'alimente en 3,3 V, **et non en 5 V**. Même si l'ESP-01 est plutôt tolérant, il est fortement recommandé d'adapter le niveau de tension si l'alimentation provient d'une source en 5 V. Sans cette précaution, vous risquez tout simplement d'endommager, voire de détruire le module !*

Matériel	Description et commentaires
<ul style="list-style-type: none"> Module ESP-01 	<ul style="list-style-type: none"> Le module ESP-01 est utilisé dans ce tutoriel, mais cela peut être une autre version moyennant quelques adaptations. Vous découvrirez toute la famille des ESP8266 sur 🇬🇧 le site de référence.
<ul style="list-style-type: none"> Plaquette de câblage rapide (<i>breadboard</i>) et jeu de câbles (<i>jumpers</i>) mâle-mâle et mâle-femelle. 	<ul style="list-style-type: none"> Vous trouverez ces composants électroniques de base avec bien d'autres composants dans la plupart des kits d'électronique pour débutants.
<ul style="list-style-type: none"> LED, bouton-poussoir, résistances 330 Ω et 10 kΩ, condensateur 1 μF 	

	
<ul style="list-style-type: none"> Programmateurs FTDI 3,3 V 	<ul style="list-style-type: none"> Ce genre de modules embarquent une puce de conversion USB-série FTDI232RL permettant d'envoyer et recevoir des signaux UART via le port USB d'un ordinateur et de flasher un programme dans l'ESP. Choisir un module compatible 3,3 V. Certains modules permettent avec un petit interrupteur ou un cavalier (<i>jumper</i>) de choisir entre 5 V et 3,3 V.
<ul style="list-style-type: none"> Module d'alimentation 3,3 V pour plaque de câblage (<i>breadboard power supply</i>) 	<ul style="list-style-type: none"> Une alimentation 3,3 V séparée est fortement recommandée, car l'ESP8266 peut absorber un courant assez élevé pour fonctionner (supérieur à 200 mA par moment). Vous ne devez pas alimenter l'ESP8266 directement à partir de la sortie 3,3 V d'une carte Arduino sous peine d'avoir un comportement erratique de l'ESP (dans le meilleur des cas, parce qu'au pire...). Ce genre de modules peut s'enficher sur les plaquettes de câblage standards. Ils sont munis d'une prise au format <i>jack</i> 3,5 mm pour alimenter le module à partir d'une pile, d'une batterie, d'un adaptateur secteur... (6 à 12 V).
<ul style="list-style-type: none"> Convertisseur de niveau logique bidirectionnel 3,3 V-5 V 	<ul style="list-style-type: none"> Ce module permet de convertir des signaux logiques de façon bidirectionnelle : 5 V vers 3,3 V et 3,3 V vers 5 V simultanément. On le place sur la liaison série (Rx et Tx) entre une carte Arduino ou un programmeur opérant en 5 V et l'ESP-01 opérant en 3,3 V. Bien entendu, si votre programmeur ou votre Arduino opère déjà en 3,3 V, ce module n'est pas utile. <div data-bbox="754 1966 1158 2036">  On trouve beaucoup de montages sur le Net </div>

avec une sortie 5 V du programmeur ou de l'Arduino directement reliée à l'entrée Rx en 3,3 V de l'ESP. Comme dit plus tôt, l'ESP-01 est assez tolérant, c'est à vous de voir si vous prenez le risque... On trouve aussi des montages avec **pont diviseur de tension** pour abaisser du 5 V à 3,3 V. Dans ce cas, attention à la qualité des signaux en sortie du pont, notamment pour des vitesses de transmission élevées (supérieures à 9 600 bauds)

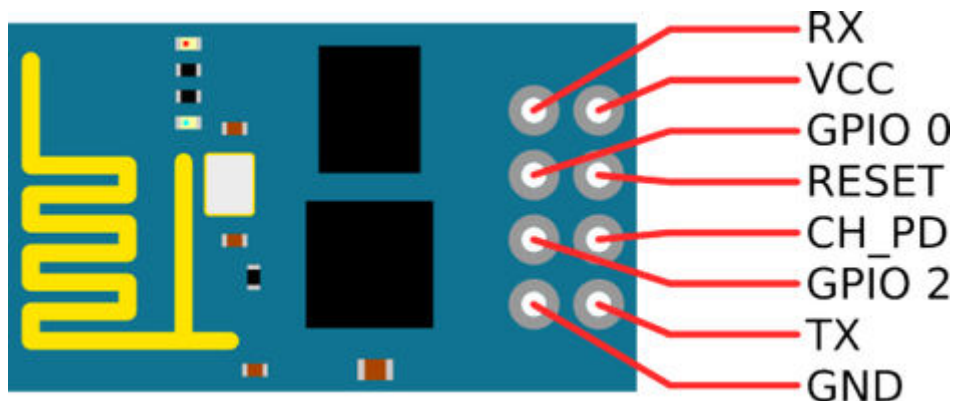
III - Schéma du montage



Le programmeur FTDI est le circuit en rouge en haut du schéma, l'ESP8266-01 est en bas. Seuls trois fils du programmeur FTDI sont connectés : le retour à la masse GND, et les broches de communication Rx et Tx reliées respectivement au Tx et Rx de l'ESP. À gauche du schéma, le module d'alimentation normalement directement enfiché sur la plaque de câblage.

On insiste une fois de plus sur la qualité nécessaire de l'alimentation de l'ESP. Si vous constatez un fonctionnement erratique du système, il s'agit probablement d'un souci d'alimentation. Remarquez le condensateur de découplage entre les lignes 3,3 V et la masse GND afin de combler les éventuelles chutes de tension dues aux pointes de courant demandées par l'ESP.

IV - Le brochage de l'ESP-01



d'après <https://iotbytes.wordpress.com/esp8266-pinouts/>

Quelques remarques sur le câblage :

- la broche GPIO 0 doit être maintenue à la masse (GND) pendant le téléversement du code. Elle doit être déconnectée si vous retirez le FTDI et que l'ESP est en mode d'exécution normal ;
- la broche CH_PD doit toujours être maintenue à l'état haut ;
- la broche RESET est tirée à l'état haut avec une résistance de *pull-up* 10 kΩ, et est reliée à la masse GND sur appui du bouton RESET lorsqu'il faut redémarrer l'ESP. Pressez le bouton RESET à chaque fois avant de téléverser le code, et à chaque fois que vous connectez ou déconnectez la broche GPIO 0. Ce bouton RESET vous évitera bien des tracas.

Quand vous alimentez le circuit, la LED rouge de l'ESP-01 doit s'allumer, et la LED bleue doit s'allumer brièvement. Plus tard, quand vous téléverserez le code dans l'ESP, la LED bleue devrait clignoter pendant le processus. Toute autre combinaison dans l'éclairage des LED suggère un dysfonctionnement.



Si vous souhaitez retirer le programmeur série et démarrer l'ESP en mode d'exécution normal, assurez-vous d'avoir déconnecté la broche GPIO 0 de la masse GND et pressez le bouton RESET, sans cela l'ESP risque de ne pas fonctionner correctement.

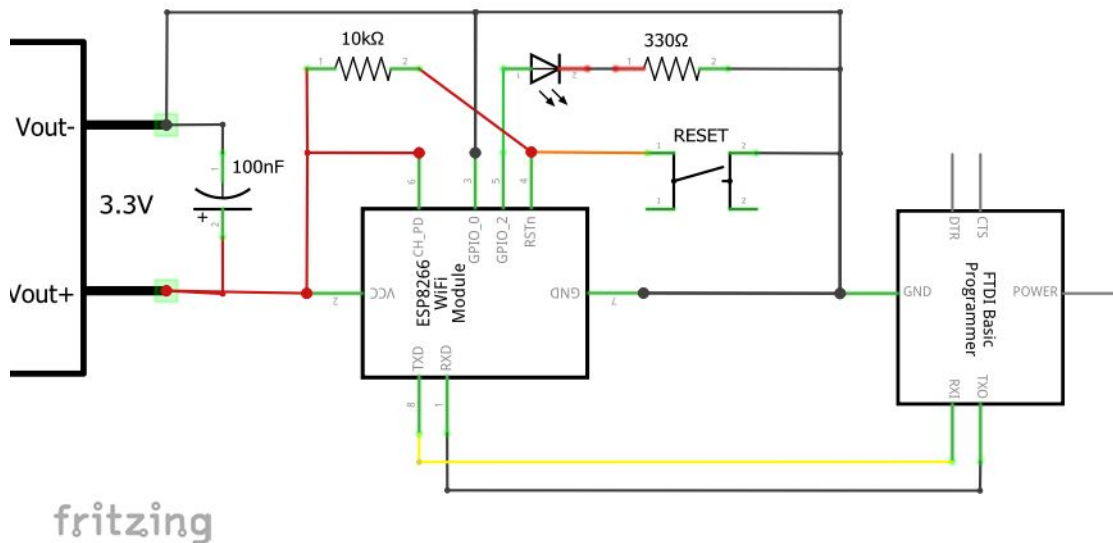


Schéma de câblage

V - Utiliser une carte Arduino comme programmeur série

À la place de votre programmeur USB-série, vous pouvez vous servir de votre carte Arduino.


Quelques remarques :

- 1 Sur une carte Arduino opérant en 5 V (comme la classique Arduino UNO), les sorties logiques seront au potentiel 5 V (logique **TTL**), et vous risquez d'endommager votre ESP8266 en les reliant directement. Vous devez donc utiliser une carte Arduino opérant en 3,3 V ou bien rajouter un convertisseur de niveau logique bidirectionnel 3,3 V-5 V (*bi-directional logic level converter*) entre l'Arduino et l'ESP ;
- 2 La sortie 3,3 V de votre carte Arduino est insuffisante pour alimenter l'ESP qui nécessite un courant élevé. Là encore, pour éviter les comportements erratiques et pour protéger les composants, utilisez une alimentation 3,3 V externe pouvant fournir 0,5 A ;
- 3 Il est recommandé de téléverser un programme vierge dans l'Arduino avant d'y connecter quelques composants, quels qu'ils soient afin d'éviter certains résultats imprévisibles. On entend par programme « vierge » un code ne comprenant qu'un `setup()` et une boucle `loop()` vides ;
- 4 L'Arduino UNO ne comprend qu'un seul port série matériel (connecteurs Rx et Tx). Pour communiquer à la fois avec l'ESP8266 et votre ordinateur via le Moniteur Série, vous devrez passer par un port série logiciel et utiliser la bibliothèque **Software Serial**. Un port série logiciel a forcément quelques limitations, mais la communication devrait théoriquement se dérouler normalement avec un débit allant jusqu'à 115 200 bauds. L'avantage d'une carte Arduino Mega est de disposer de ports série matériels supplémentaires ;
- 5 **Enfin, le piège...** Quand vous utilisez l'Arduino comme programmeur série d'un autre périphérique comme l'ESP, les données (notamment les commandes AT tapées dans le Moniteur Série de l'EDI Arduino) en provenance de votre ordinateur relié à l'Arduino via le câble USB arrivent sur le port Rx de l'Arduino (l'Arduino reçoit bien les données). Et ces données doivent parvenir sur la broche Rx du périphérique qui doit aussi les recevoir. **Dans ce cas précis, il ne faut plus croiser Rx et Tx, vous devez connecter le Rx de l'Arduino au Rx du périphérique, et le Tx de l'Arduino au Tx du périphérique.**

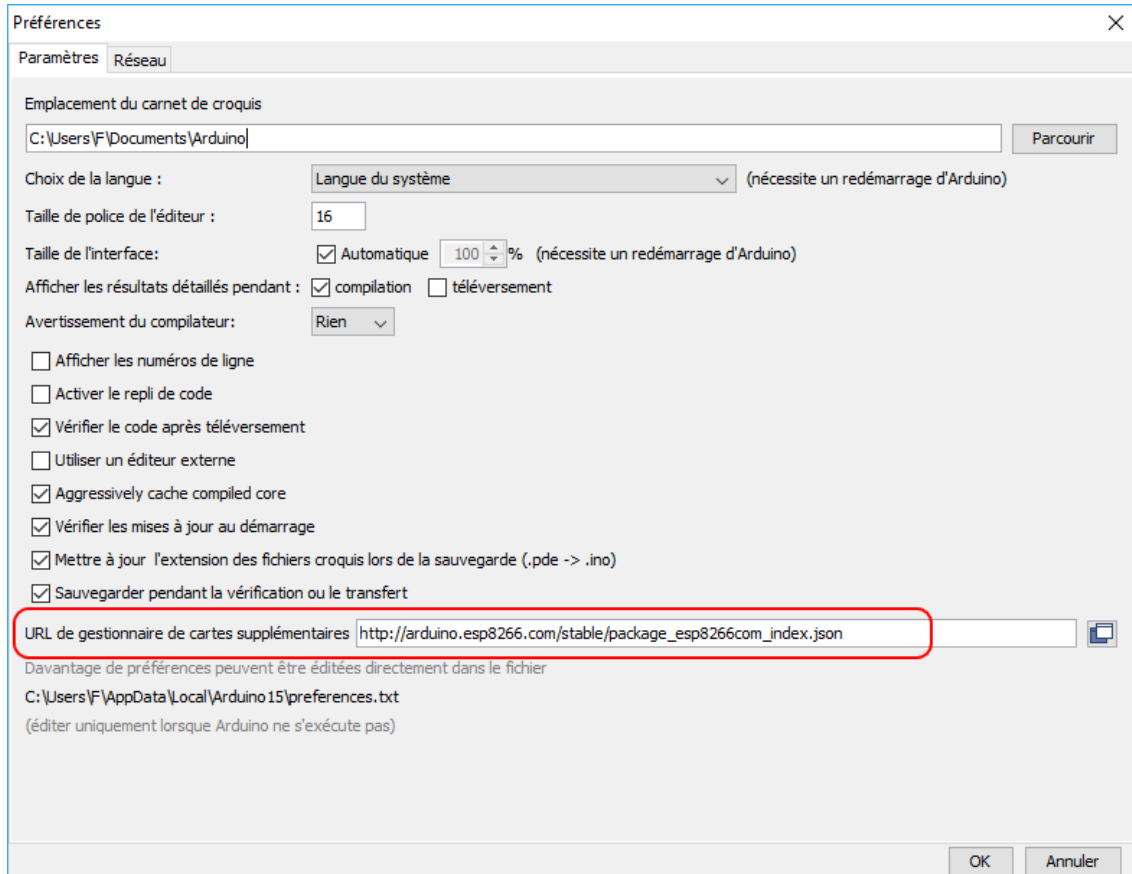
VI - Programmer avec l'EDI Arduino

VI-A - Installation du gestionnaire

De nombreux outils existent pour programmer l'ESP8266, mais l'EDI Arduino (Environnement de Développement Intégré) est sans doute le moyen le plus facile d'y arriver.

Vous aurez besoin de  **la dernière version de l'EDI Arduino** et d'y ajouter le support de l'ESP8266. Notez une fois de plus que même si vous passez par l'EDI Arduino, vous n'êtes pas obligé de posséder une carte Arduino pour autant.

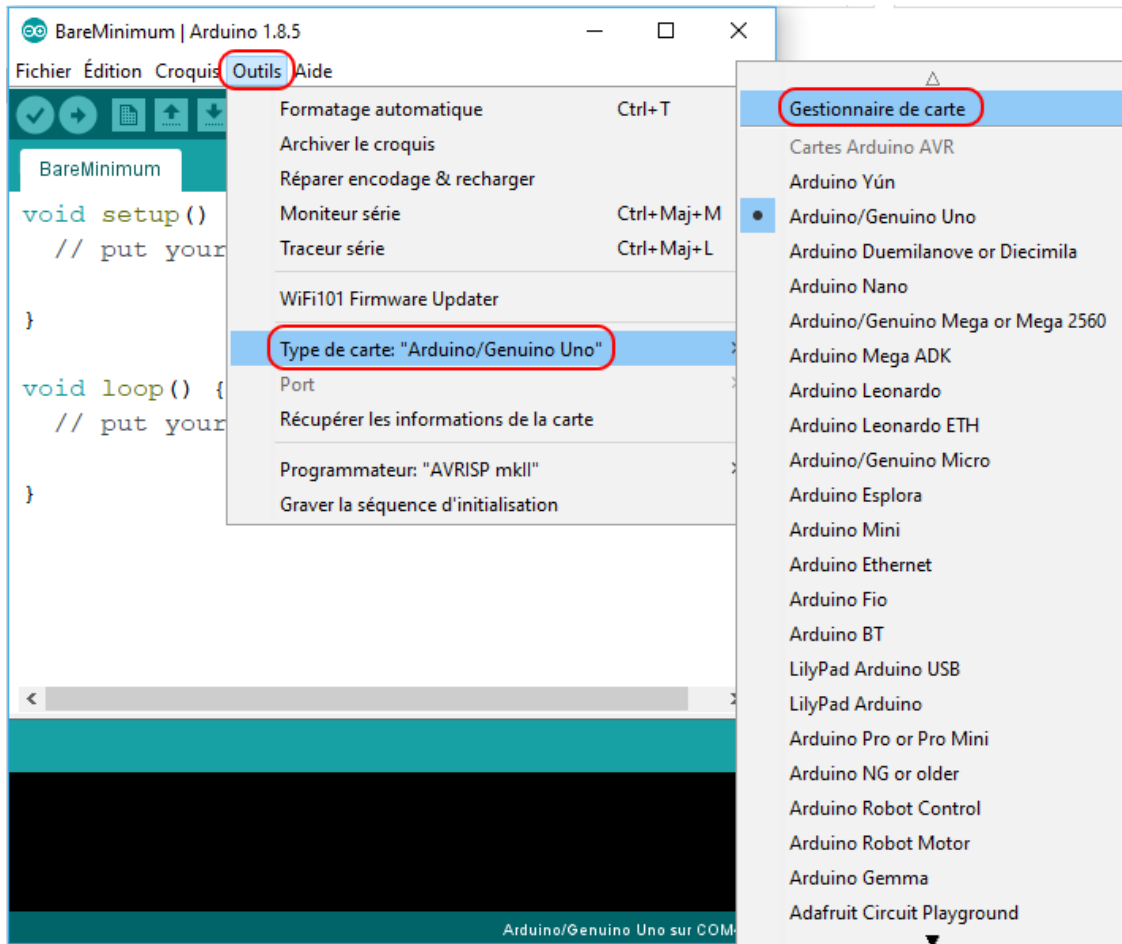
Dans le menu **Préférences** de l'EDI Arduino, il faut rentrer une URL pour gérer les cartes supplémentaires :



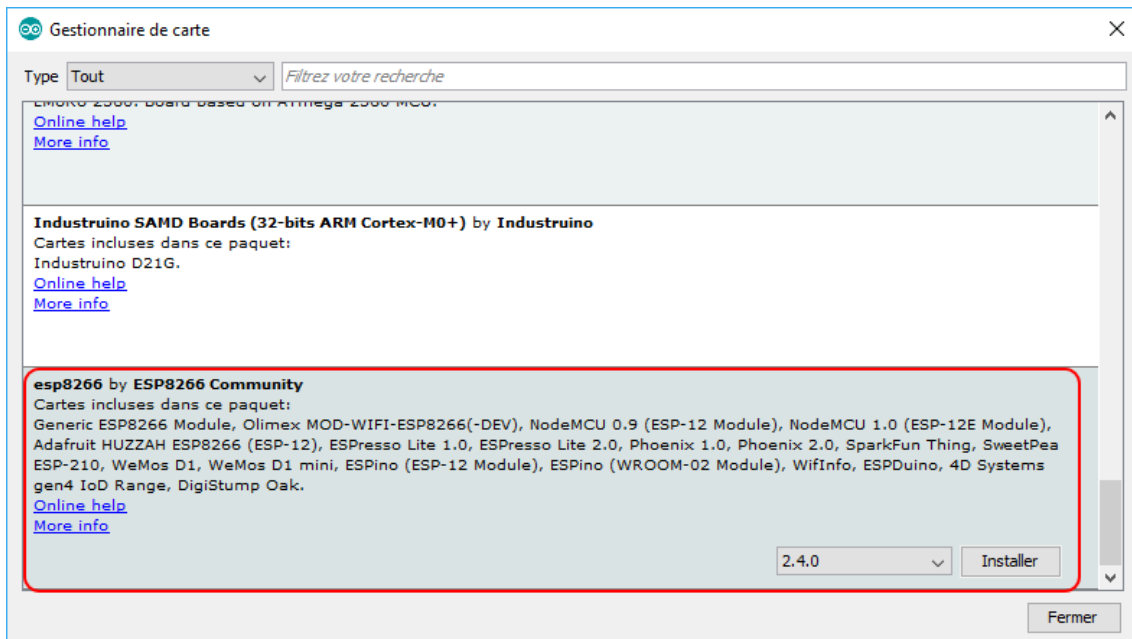
Ici, il faut rentrer l'URL : http://arduino.esp8266.com/stable/package_esp8266com_index.json.

Plus de détails sur le site  **ESP8266 Arduino Core**.

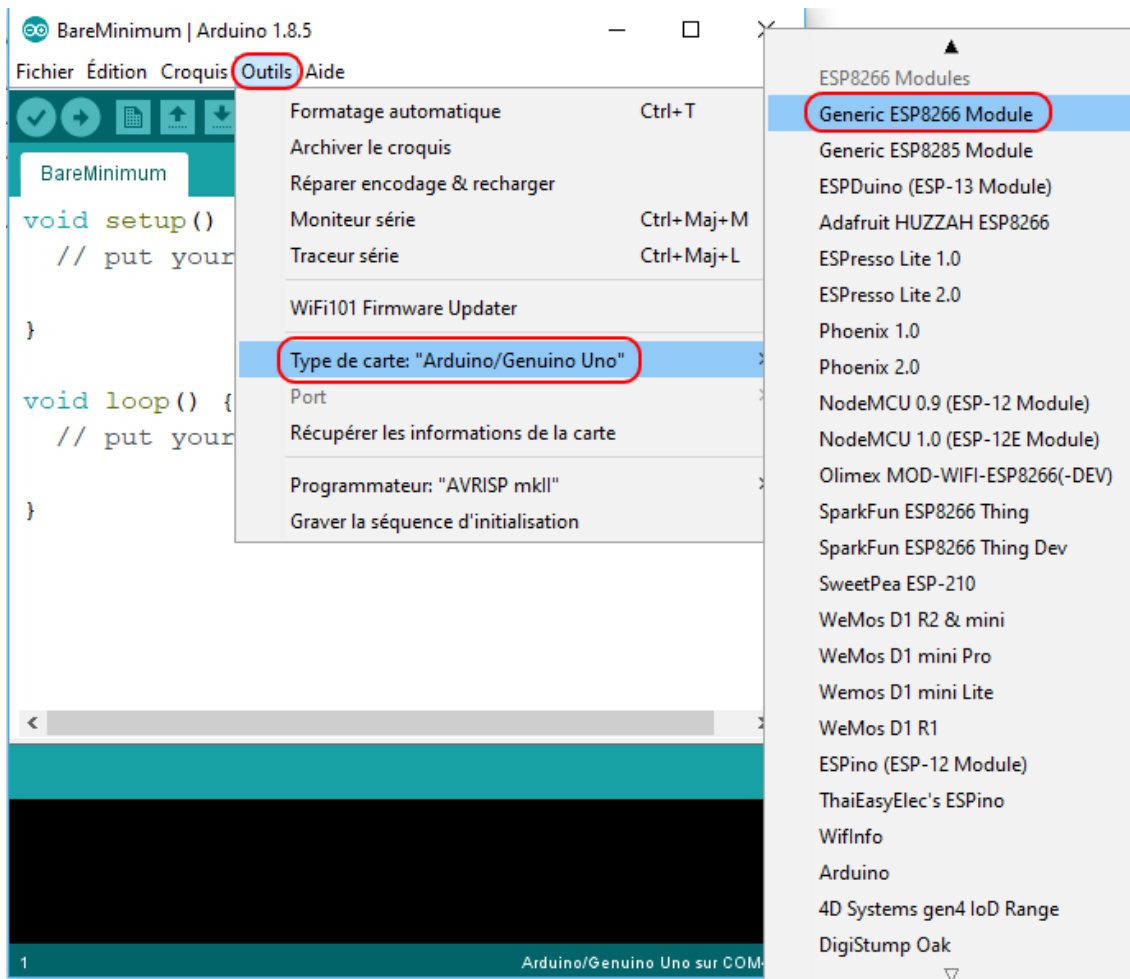
Sélectionnez ensuite le **Gestionnaire de cartes** (*Boards Manager*) :



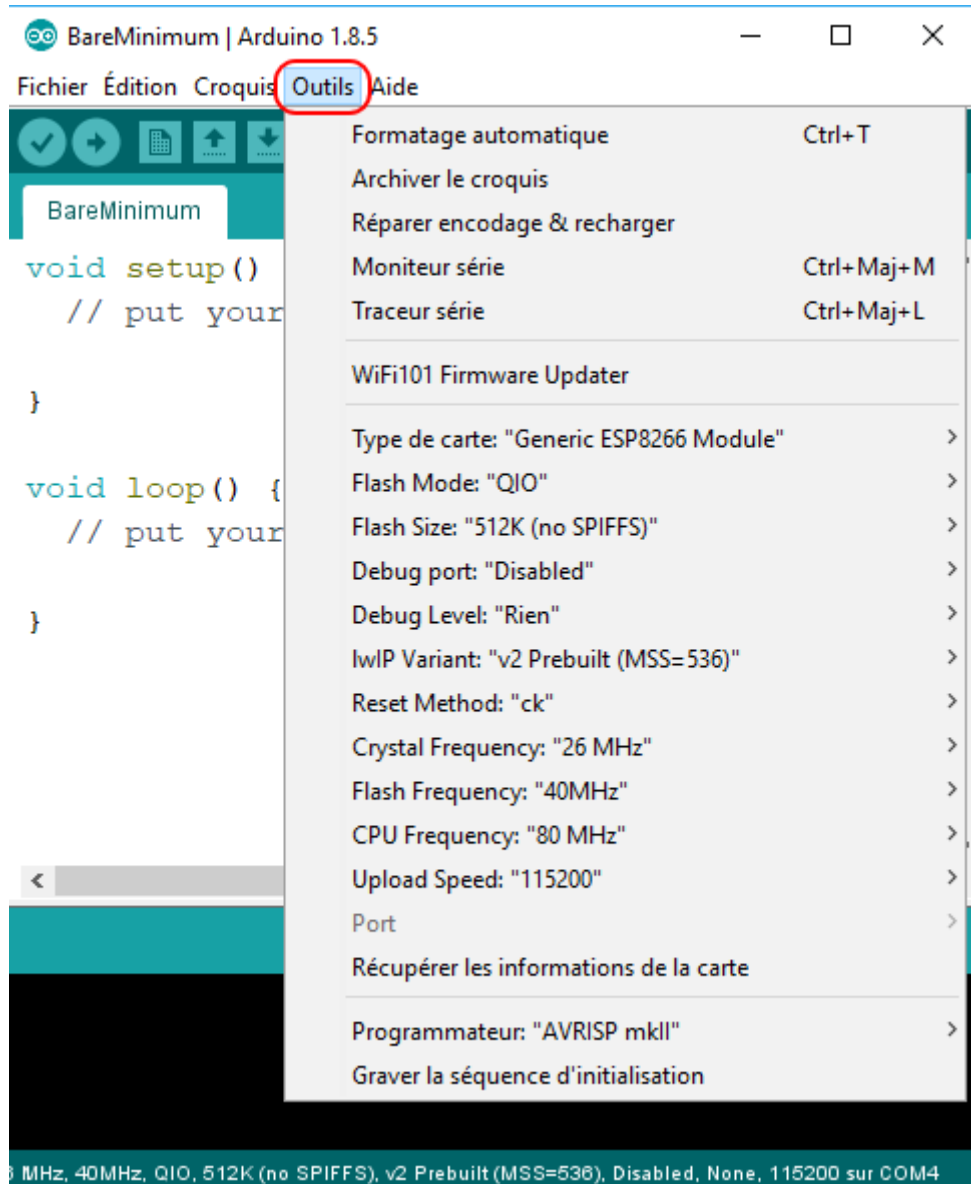
Dans le gestionnaire, sélectionnez le paquet **esp8266** :



Vous pouvez maintenant choisir la cible **Generic ESP8266 Module** :



Des options supplémentaires apparaissent si vous retournez dans le menu **Outils** :

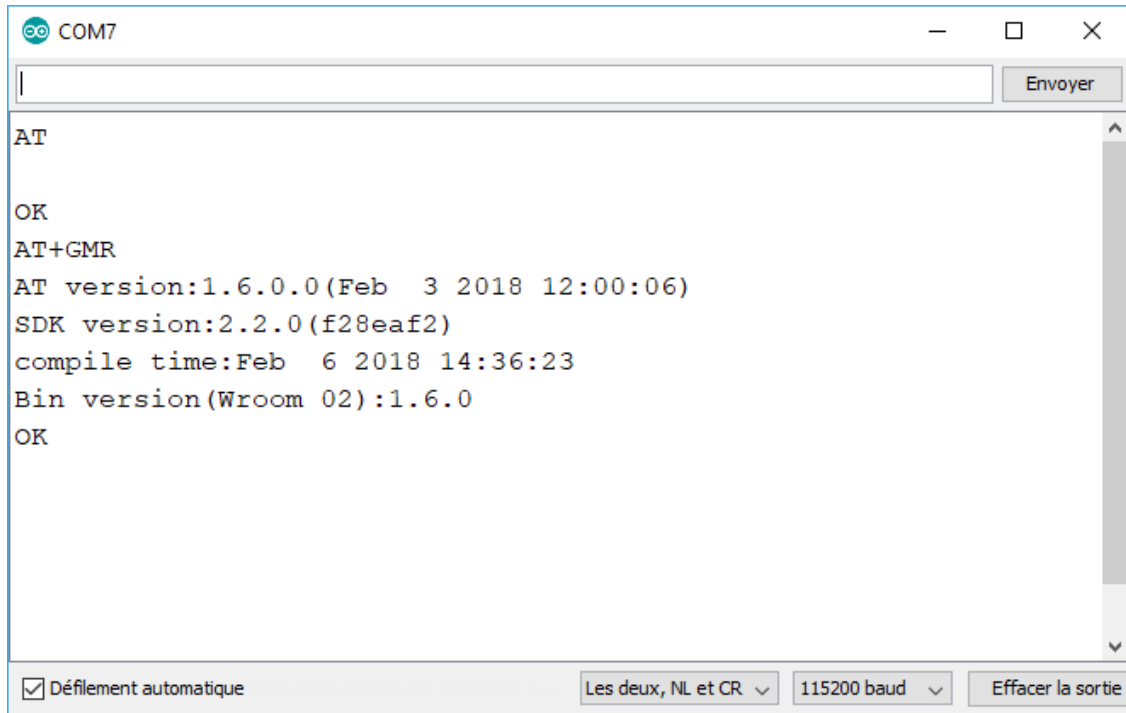


VI-B - Communiquer avec l'ESP8266 à l'aide de commandes AT

Une fois l'ESP connecté à votre poste, sélectionnez le port COM correspondant (menu **Outils/Port...**).

Si le firmware d'origine est toujours présent dans l'ESP8266, vous pouvez communiquer avec lui au travers du Moniteur Série :

- 1 Ouvrez le Moniteur Série (*Serial Monitor*) ;
- 2 Sélectionnez la vitesse de transmission, normalement 115 200 bauds (*baud rate*), mais selon le firmware, elle peut être inférieure ;
- 3 Sélectionnez « Les deux, NL et CR » (*Both NL & CR*), pour envoyer un *New Line* et *Carriage Return* à chaque envoi (bouton *Envoyer*) ;
- 4 Tapez (sans les guillemets) la commande « AT », puis cliquez sur le bouton « Envoyer ». L'ESP devrait renvoyer « OK » dans le Moniteur Série. Si ce n'est pas le cas, pressez le bouton RESET ou essayez à nouveau en redémarrant après avoir débranché le câble USB. Vérifiez aussi que vous avez sélectionné le bon port COM. Si l'ESP ne répond toujours pas, peut-être que le firmware « standard » n'est pas installé et qu'il ne comprend pas les commandes AT. Dans ce cas, cela n'est pas vraiment un problème puisque vous allez réécrire par-dessus le firmware avec votre propre code ;
- 5 Tapez « AT+GMR ». L'ESP retourne le numéro de version du firmware.



VI-C - Le code

Copiez et collez le code ci-dessous dans la fenêtre d'édition de l'EDI Arduino. Il faut modifier les variables **SSID** et **password** pour que l'ESP se connecte à votre routeur.

Notez que la LED est connectée à la broche GPIO 2 de l'ESP, d'où l'initialisation de la variable `int pinGPIO2 = 2;`.

ESP8266_LED_Control.ino

```
1. /*
2.  This is a demo of ESP8266 WiFi Module in standalone mode (without Arduino) controlling an LED via a Web page
3.  This code also demonstrates that PWM is available on the ESP8266, so we can dim the LED.
4.  Analog Input is available on the ESP8266 but this pin is not wired up on ESP-01 board.
5.  The ESP can also support interrupts.
6.
7.  This demo uses the Arduino IDE and the Arduino Boards Manager to upload the program to the ESP8266 with an F
8.  N.B. This overwrites any Firmware that was previously on the ESP8266.
9.
10. See: http://www.instructables.com/id/ESP8266-WiFi-Module-for-Dummies/ for the details on how to set the boar
11. Also see: http://www.instructables.com/id/The-Simple-Guide-to-Flashing-Your-ESP8266-Firmware/
12. */
13.
14. // Include the ESP8266 Library. This library is automatically provided by the ESP8266 Board Manager and does
15. #include <ESP8266WiFi.h>
16.
17. String codeVersion = "Version 1.0 Aug 2016 by TonesB";
18.
19. // WiFi Router Login - change these to your router settings
20. const char* SSID = "YourSSID";
21. const char* password = "YourPassword";
22.
23. // Setup GPIO2
24. int pinGPIO2 = 2; //To control LED
25. int ledStatus = 0; //0=off,1=on,2=dimmed
26.
27. // Create the ESP Web Server on port 80
28. WiFiServer WebServer(80);
29. // Web Client
30. WiFiClient client;
31.
32. void setup() {
33.   Serial.begin(115200);
```


ESP8266_LED_Control.ino

```
34. delay(10);
35. Serial.println();
36. Serial.println();
37. Serial.println(codeVersion);
38.
39. // Setup the GPIO2 LED Pin - this demo also uses PWM to dim the LED.
40. pinMode(pinGPIO2, OUTPUT);
41. digitalWrite(pinGPIO2, LOW);
42. ledStatus = 0;
43.
44. // Connect to WiFi network
45. Serial.println();
46. WiFi.disconnect();
47. WiFi.mode(WIFI_STA);
48. Serial.print("Connecting to ");
49. Serial.println(SSID);
50. WiFi.begin(SSID, password);
51.
52. while (WiFi.status() != WL_CONNECTED) {
53.     delay(500);
54.     Serial.print(".");
55. }
56. Serial.println("");
57. Serial.println("Connected to WiFi");
58.
59. // Start the Web Server
60. WebServer.begin();
61. Serial.println("Web Server started");
62.
63. // Print the IP address
64. Serial.print("You can connect to the ESP8266 at this URL: ");
65. Serial.print("http://");
66. Serial.print(WiFi.localIP());
67. Serial.println("/");
68. }
69.
70. void loop() {
71.     // Check if a user has connected
72.     client = WebServer.available();
73.     if (!client) { //restart loop
74.         return;
75.     }
76.
77.     // Wait until the user sends some data
78.     Serial.println("New User");
79.     while (!client.available()) {
80.         delay(1);
81.     }
82.
83.     // Read the first line of the request
84.     String request = client.readStringUntil('\r\n');
85.     Serial.println(request);
86.     client.flush();
87.
88.     // Process the request:
89.     if (request.indexOf("/LED=ON") != -1) {
90.         analogWrite(pinGPIO2, 1023);
91.         ledStatus = 1;
92.     }
93.     if (request.indexOf("/LED=OFF") != -1) {
94.         analogWrite(pinGPIO2, 0);
95.         ledStatus = 0;
96.     }
97.     if (request.indexOf("/LED=DIM") != -1) {
98.         analogWrite(pinGPIO2, 512);
99.         ledStatus = 2;
100.    }
101.
102.    // Return the response
103.    client.println("HTTP/1.1 200 OK");
104.    client.println("Content-Type: text/html; charset=UTF-8");
```

ESP8266_LED_Control.ino

```

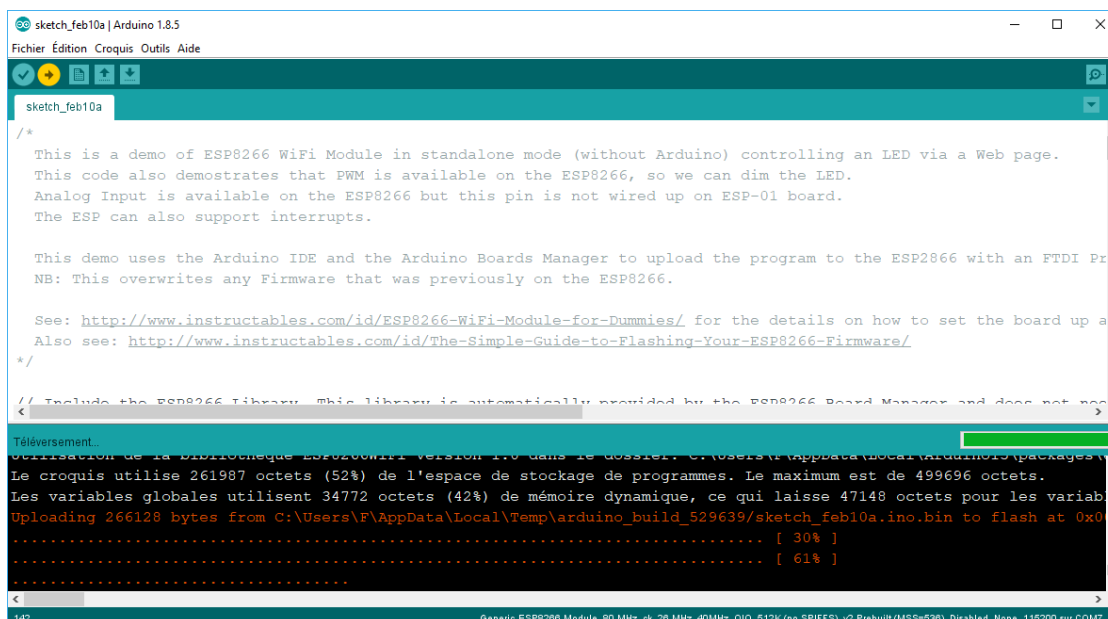
105.  client.println("");
106.  client.println("<!DOCTYPE HTML>");
107.  client.println("<html>");
108.  client.println("<head>");
109.  client.println("<title>ESP8266 Demo</title>");
110.  client.println("</head>");
111.  client.println("<body>");
112.  client.println("<a href=\"\"/>Refresh Status</a>");
113.  client.println("</br></br>");
114.
115.  //check the LED status
116.  if (ledStatus == 0) {
117.    client.print("LED is Off<br>");
118.    client.println("Turn the LED <a href=\"/LED=ON\">ON</a></br>");
119.    client.println("Set LED to <a href=\"/LED=DIM\">DIM</a></br>");
120.  } else if (ledStatus == 1) {
121.    client.print("LED is On<br>");
122.    client.println("Turn the LED <a href=\"/LED=OFF\">OFF</a></br>");
123.    client.println("Set LED to <a href=\"/LED=DIM\">DIM</a></br>");
124.  } else if (ledStatus == 2) {
125.    client.print("LED is Dimmed<br>");
126.    client.println("Turn the LED <a href=\"/LED=OFF\">OFF</a></br>");
127.    client.println("Turn the LED <a href=\"/LED=ON\">ON</a></br>");
128.  }
129.
130.  client.println("</br>");
131.
132.  client.println("<a href=\"http://www.instructables.com/id/ESP8266-WiFi-Module-for-Dummies/\" target=\"_blank\">S");
133.  client.println("<a href=\"http://www.instructables.com/id/The-Simple-Guide-to-Flashing-Your-ESP8266-Firmware/\"");
134.  client.println("</br>");
135.  client.println("</body>");
136.  client.println("</html>");
137.
138.  delay(1);
139.  Serial.println("User disconnected");
140.  Serial.println("");
141.
142. }
```

Vérifiez une dernière fois que votre montage est au point et connectez votre câble USB au programmeur série.

VI-D - Téléverser le code dans l'ESP8266

Quand vous téléversez le code, il faut que la broche GPIO 0 soit reliée à la masse GND. Vous pourrez retirer le fil de câblage correspondant une fois le téléversement effectué. Pressez le bouton RESET avant de téléverser. Si le téléversement échoue, déconnectez et reconnectez le câble USB puis recommencez. En dernier recours, redémarrez votre PC, il arrive parfois que le port COM perde pied.

La progression du téléversement s'affiche dans la fenêtre de notification de l'EDI comme ci-dessous :



```

sketch_feb10a | Arduino 1.8.5
Fichier Édition Croquis Outils Aide

sketch_feb10a

/*
 * This is a demo of ESP8266 WiFi Module in standalone mode (without Arduino) controlling an LED via a Web page.
 * This code also demonstrates that PWM is available on the ESP8266, so we can dim the LED.
 * Analog Input is available on the ESP8266 but this pin is not wired up on ESP-01 board.
 * The ESP can also support interrupts.
 *
 * This demo uses the Arduino IDE and the Arduino Boards Manager to upload the program to the ESP2866 with an FTDI Pr
 * NB: This overwrites any Firmware that was previously on the ESP8266.
 *
 * See: http://www.instructables.com/id/ESP8266-WiFi-Module-for-Dummies/ for the details on how to set the board up a
 * Also see: http://www.instructables.com/id/The-Simple-Guide-to-Flashing-Your-ESP8266-Firmware/
 */

// Include the ESP8266 library. This library is automatically provided by the ESP8266 Board Manager and does not nee

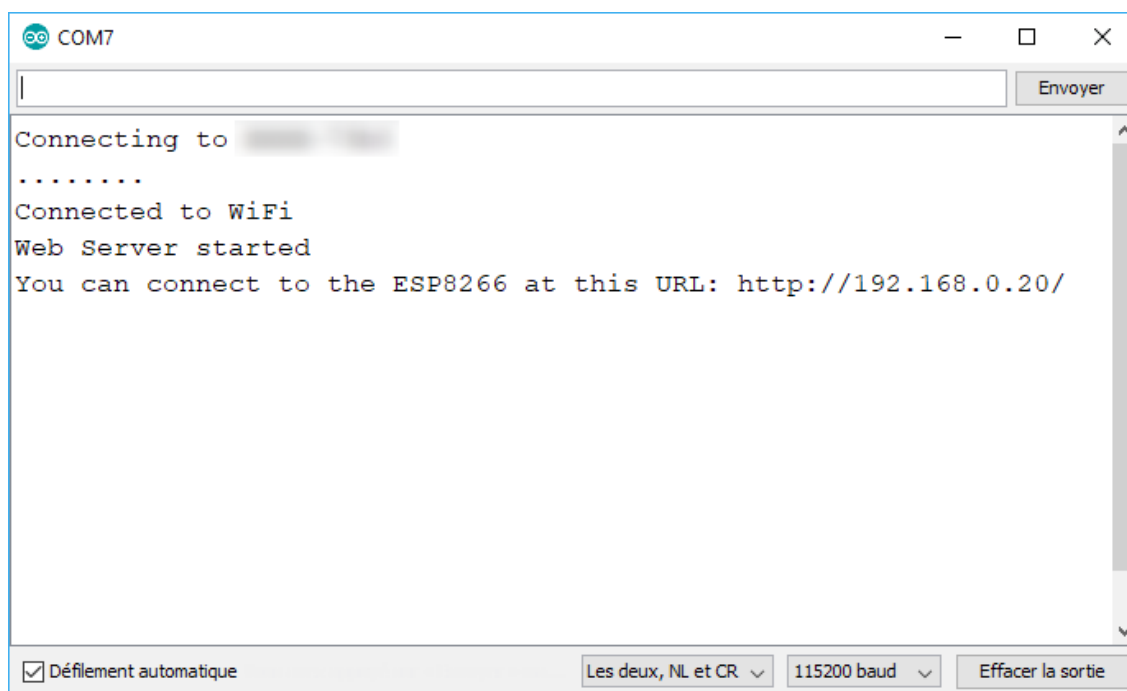
Téléversement...
Installation de la bibliothèque ESP8266WiFi version 1.0 dans le dossier C:\Users\F\AppData\Local\Arduino15\packages\...
Le croquis utilise 261987 octets (52%) de l'espace de stockage de programmes. Le maximum est de 499696 octets.
Les variables globales utilisent 34772 octets (42%) de mémoire dynamique, ce qui laisse 47148 octets pour les variab
Uploading 266128 bytes from C:\Users\F\AppData\Local\Temp\arduino_build_529639/sketch_feb10a.ino.bin to flash at 0x0
..... [ 30% ]
..... [ 61% ]
  
```

La LED bleue de l'ESP doit clignoter pendant le téléversement.

Quand vous retirerez le programmeur série une fois le code téléversé, vous devrez retirer le câble reliant la broche GPIO 0 à la masse, puis presser le bouton RESET pour que l'ESP démarre correctement.

VI-E - L'ESP8266, serveur web

Ouvrez le Moniteur Série, vous devriez voir apparaître l'adresse IP prise par l'ESP. Vous pouvez également regarder dans l'interface d'administration de votre routeur pour découvrir les périphériques qui y sont connectés. L'ESP devrait apparaître dans la liste avec un nom du type : ESP_1A64CA. Tant que vous y êtes, vous pourriez réserver une adresse fixe pour l'ESP et paramétrer le port (*port forwarding*) pour pouvoir vous connecter à l'ESP en dehors de votre domicile sur Internet.



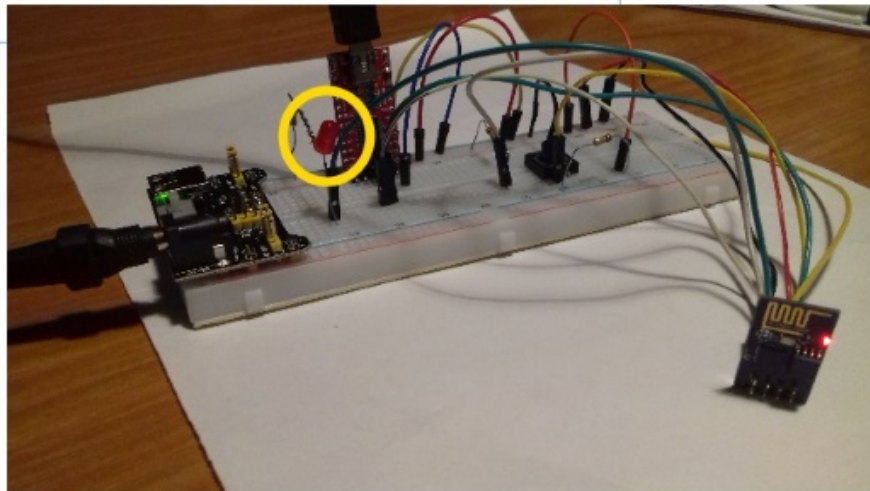
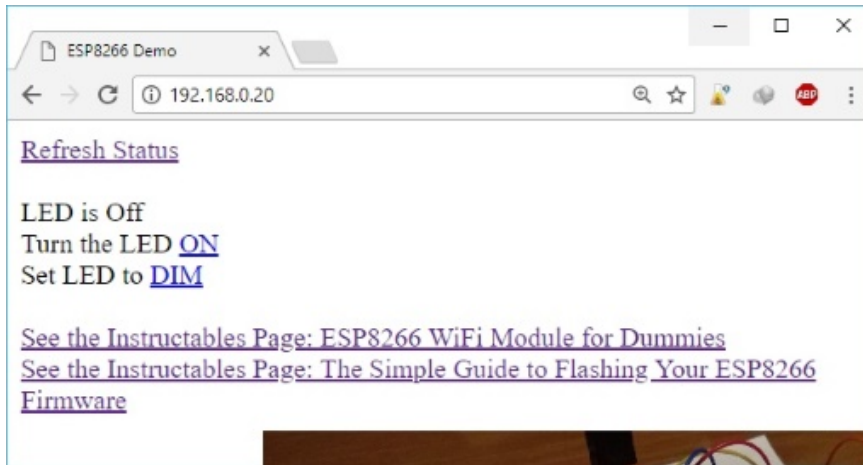
```

COM7

Connecting to .....
Connected to WiFi
Web Server started
You can connect to the ESP8266 at this URL: http://192.168.0.20/

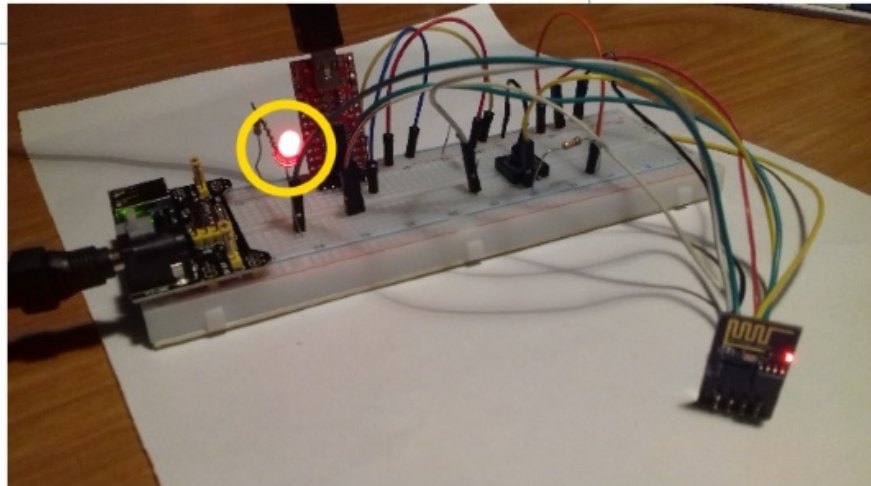
Défilement automatique Les deux, NL et CR 115200 baud Effacer la sortie
  
```

Ouvrez maintenant votre navigateur et saisissez l'adresse IP de l'ESP :



La LED est éteinte...

Testez le bon fonctionnement en cliquant sur les liens ON ou OFF pour allumer ou éteindre la LED. En cliquant sur le lien DIM, la LED devrait s'allumer, mais avec une luminosité plus faible.



... puis on clique sur le lien ON, la LED s'allume !

VII - Sitographie sur l'ESP8266

ESP8266

- Pour les plus motivés d'entre vous, l'ouvrage [🇬🇧 Kolban's book on ESP8266](#) est une mine d'informations inscrite dans un pavé de plus de 400 pages ;
- Le site de référence : [🇬🇧 esp8266.com](#), et le [🇬🇧 Wiki](#) de la communauté ESP8266 ;
- Des ressources sur le site [🇬🇧 expressif.com](#), le fabricant chinois de l'ESP8266 ;
- Pour programmer l'ESP8266 dans l'EDI Arduino : [🇬🇧 ESP8266 Arduino Core](#) ;
- Pour flasher un nouveau firmware dans l'ESP8266 : [🇬🇧 Loading new firmware onto an ESP8266](#).

Commandes AT

- [🇬🇧 un Wiki sur les commandes AT](#).
- [🇬🇧 Le jeu de commandes AT complet de l'ESP8266](#).

Gestionnaire de cartes Arduino (*Arduino boards manager*)

- [🇬🇧 GitHub ESP8266/Arduino](#)

VIII - Notes de la rédaction Developpez.com

Ce tutoriel est basé sur les travaux de [🇬🇧 TonesB](#), membre du site [🇬🇧 instructables.com](#).

Version originale du tutoriel :  **ESP8266 WiFi module for dummies**.

Nous remercions les membres de la Rédaction de **Developpez.com** pour le travail de traduction et de relecture qu'ils ont effectué, en particulier : **f-leb**, **Claude Leloup** et **François DORIN**.