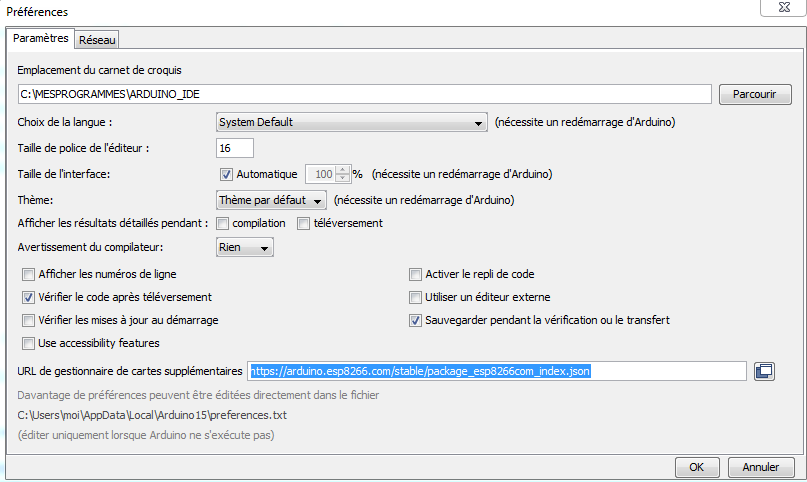
***Travailler avec un module WIFI ESP8266 sur L'IDE Arduino***

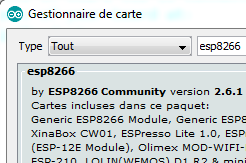
***Exemple du module WeMos D1 mini ou ESP-01***

*1) aller dans Fichier🡪préférences et ajouter le lien ci-dessous à la rubrique: URL de gestionnairede cartes supplémentaires*

*https://arduino.esp8266.com/stable/package\_esp8266com\_index.json*

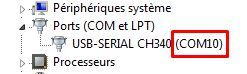
**

*2) aller dans Outils→Type de carte → gestionnaire de cartesetchercher esp8266 et installer le package*

**

*3) Revenir Outils→Type de carteetsélectionner votre module.WeMos D1 R1ou Generic ESP8266 modulePour le module ESP-01*

*4) Une fois la carte connectée au PC, le driver USB-Serie lui affecte un port COM. Voir le gestionnaire de périphériques pour le trouver*

**

*C'est ce port qu'il faut préciser dans Outils→ Port:de l'IDE-Arduino*

*Pour le module* ***ESP-01*** *voir plus bas dans ce document comment il faut le brancher au PC et comment il faut basculer entre mode programmation et mode Exécution*

***Test***

*Pour faire le premier essai, rien de mieux que faire clignoter la LED présente sur le module. Sur le module D1 mini, la LED est connectée sur la pate 2 du ESP8266 qui est elle même connectée à la pate D4 du module. Le mieux est d'utiliser la constante prédéfinie LED\_BUILTIN qui s'adapte au module utilisé*

*#define LED 2*

*void setup() {*

*pinMode(LED\_BUILTIN, OUTPUT);*

*}*

*voidloop() {*

*digitalWrite(LED\_BUILTIN, HIGH);*

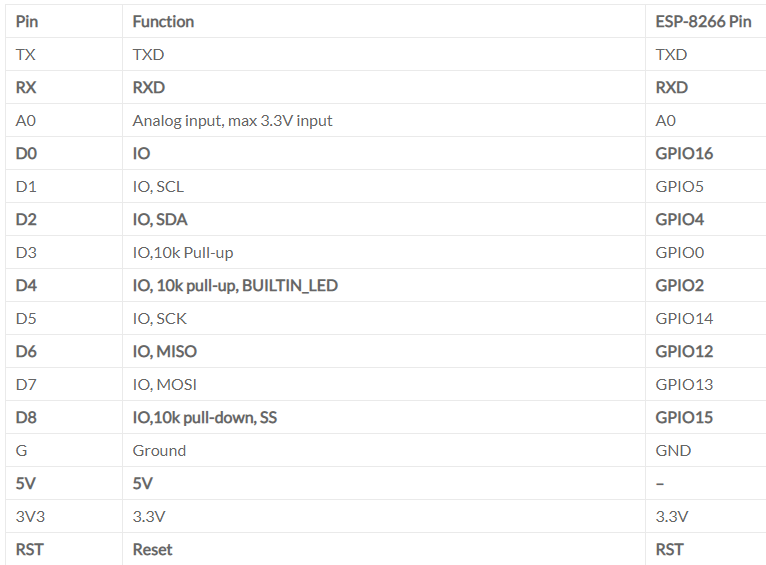
*delay(500);*

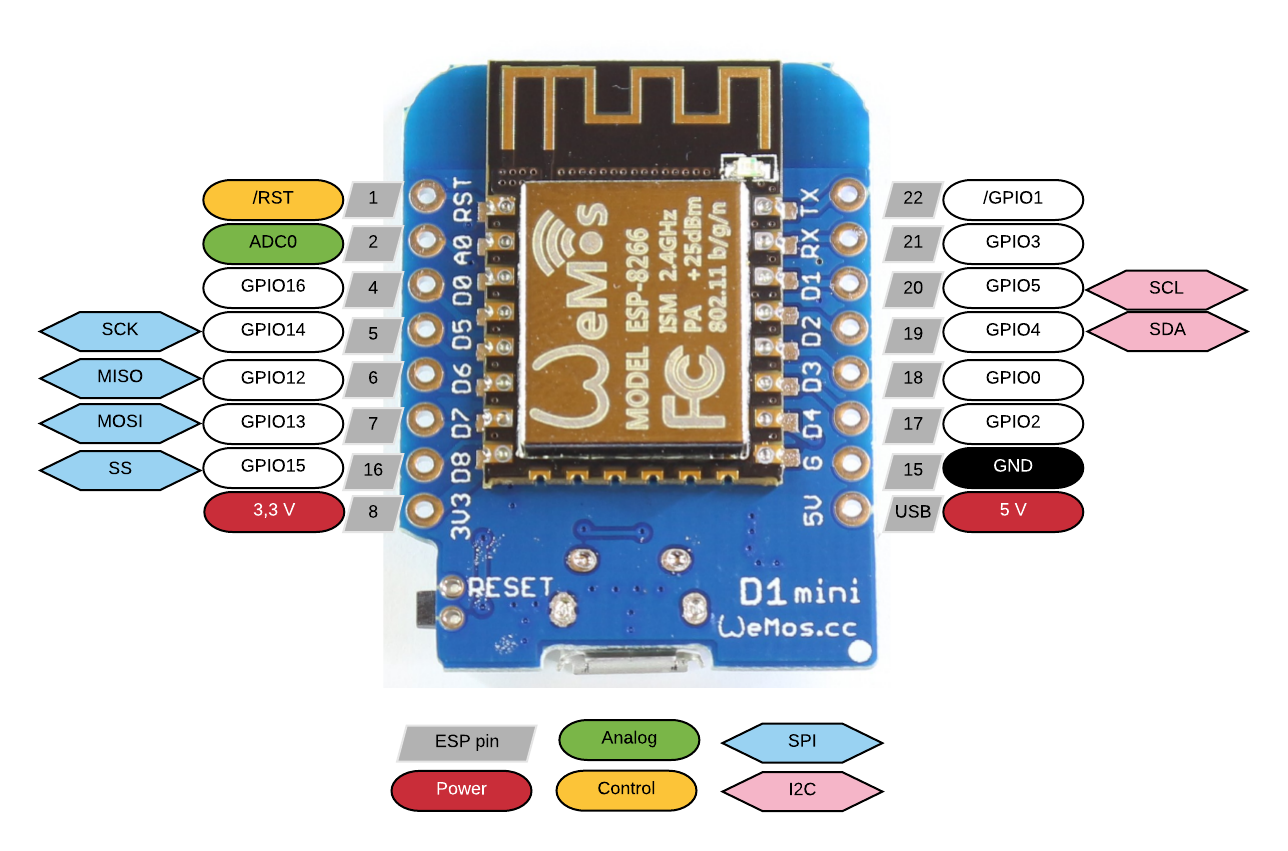
*digitalWrite(LED\_BUILTIN, LOW);*

*delay(500);*

*}*

*Voici le pinout du module D1 mini*

**



*Attention: Toutes les entrées/sorties sont 3.3V*

***Flasher le module ESP-01 avec unfirmware AT***

Il arrive que l'on rencontre quelques ennuis pour utiliser ce module. Normalement, à l'achat, il devrait contenir un *firmware* qui répond aux commande AT, cela n'est pas toujours le cas surtout avec le module noir. Soit qu'il ne contient pas le bon *firmware*, soit qu'il n'est pas en mode RUN. En effet le module possède deux modes:

* Un mode RUN dans lequel il exécute le programme qu'il contient
* Un mode programmation dans lequel il attend de recevoir un programme

Pour passer d'un mode à l'autre il faut utiliser deux boutons poussoir:

Si on un adaptateur USB-Serial, on peut utiliser le module ci-dessous

*3.3V*

*EN*

*RST*

*TX*

*Rx*

*IO2*

*IO0*

*GND*

*3.3V*

*GND*

*TX*

*Rx*

***USB-Serial***

***ESP-01***

***P***

***RST***

*3.3V*

*EN*

*RST*

*TX*

*Rx*

*IO2*

*IO0*

*GND*

*3.3V*

*GND*

*TX*

*Rx*

***USB-Serial***

***ESP-01***

***P***

***RST***



Si on ne dispose pas d'un adaptateur USB-Serial, il faut savoir qu'il y'en a un sur l'Arduino. Mais il faut faire attention car dans ce cas le module ESP-01 et branché en parallèle sur le processeur du arduino, cela a plusieurs conséquence:

1. Il faut brancher le RX avec le RX et le TX avec le TX
2. Il ne faut pas que le processeur du Arduino contienne un programme qui utilise le port série. le mieux est d'y téléverser un programme vide,
3. Avec ce branchement, Si on utilise l'IDE-Arduino, on peut téléverser un programme soit vers le processeur de l'Arduino, soit vers l'ESP-01. Pour programmer l'Arduino il faut choisir *Outils→type de carte→Arduino*… Pour programmer l'ESP-01, il faut choisir *Outil→type de carte→Generic ES8266*(il ne faut pas oublier de placer le module en mode programmation. une fois la programmation terminée, il faut le replacer en mode RUN pour voir l'exécution du programme)

*3.3V*

*EN*

*RST*

***TX***

***RX***

*IO2*

*IO0*

*GND*

*3.3V*

*GND*

*RX*

*TX*

***ARDUINO***

***ESP-01***

***P***

***RST***

***USB/Serial***

***Processeur***

***Atmega***

***TX***

***TX***

***RX***

***RX***

***USB***

***0***

***1***



* Les broches *RST, EN et IO0* disposent de résistances de pull-up internes. Il n'est donc pas nécessaire de rajouter des résistances externes.
* Pour placer le module en mode programmation (en attente de flashage): maintenir P appuyé, cliquer sur RST, relâcher P.
* Pour passer en mode RUN, c.à.d. réinitialiser le module et démarrer le programme qu'il contient: cliquer sur RST

***Remarque:***

***RX***

***ESP-01***

*1K*

*2K*

***USB-Serial***

*GND*

La majorité des modules USB-Serial (y compris celui du Arduino) délivrent un signal TX de 5V alors que le module ESP-01 attend un signal 3.3V sur sa pate RX, il faut prévoir un diviseur de tension pour abaisser la tension (moi je ne l'ai pas fait à mes risques et périls). Le mieux est d'utiliser une USB-Serial qui délivre un TX de 3.3V

Maintenant que le module est branché au PC on peut le flasher avec un firmwarequi interprète les commandes AT. On va procéder comme suit:

1. Télécharger le flasheur de NodeMCUhttps://github.com/nodemcu/nodemcu-flasher
2. Brancher le module à votre PC comme indiqué ci-dessus, placer le en mode programmation. Noter le numéro du port COM affecté par windows à votre USB-Serial ou à votre Arduino (voir gestionnaire de périphérique si nécessaire)
3. Démarrer le flasheur (en chinois, mais il n'y a pas grand-chose à faire) choisir le port COM et cliquer sur le bouton à coté à droite
4. Une fois le flash terminé, fermer le programme et placer le module en mode RUN (bouton RST)
5. Ouvrir un terminal de votre choix, (Le moniteur série de l'IDE\_Arduino ou TeraTerm ou autre), Sélectionner le port COM et la vitesse 9600 baud
6. Taper AT et valider par ↵ le module doit répondre OK
7. taper AT+CWMODE=3↵ le module doit répondre OK
8. à partir de maintenant, le module doit reconnaitre toutes les autres commandes

Pour demander au client de ne pas envoyer la requête ->GET /favicon.ico HTTP/1.1<- qui demande au serveur d'envoyer l'icône de la page web, Il faut placer la ligne suivante dans le header de la page

<link rel="icon" href="data:,">