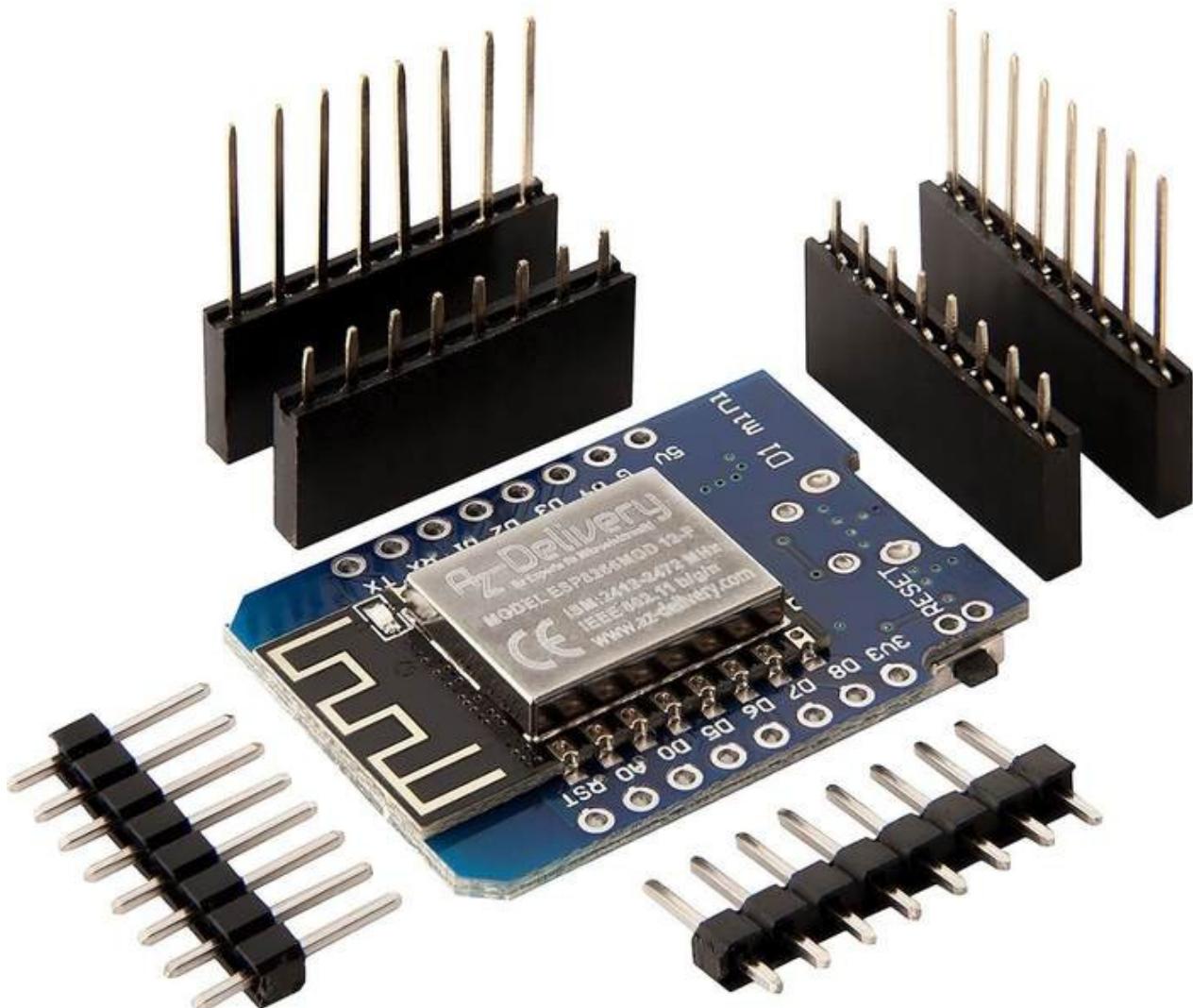




## Bienvenue !

Nous vous remercions d'avoir acheté notre module *AZ-Delivery D1 Mini ESP8266*. Dans les pages suivantes, vous découvrirez comment utiliser et configurer cet appareil pratique.

**Amusez-vous bien !**





## Sommaire

Introduction.....	3
Caractéristiques de l'ESP8266.....	4
Le module D1 Mini.....	5
Caractéristiques techniques.....	5
Broches d'E/S numériques.....	6
PWM.....	7
Entrée analogique.....	7
Serial.....	7
I2C.....	8
SPI.....	8
Le pinout.....	9
Module D1 Mini - Software.....	10
Broches d'E/S numériques.....	10
Broche d'entrée analogique.....	11
Communication sérieelle.....	12
Les interfaces I2C et SPI.....	12
Partage du temps CPU avec la partie RF.....	13
Comment configurer Arduino IDE.....	14
D1 Mini avec Arduino IDE.....	18
Exemples de sketchs Blink + PWM + Serial.....	22
Exemple de sketch Blink.....	22
Exemple de sketch PWM logiciel.....	23
Exemple de croquis de communication série.....	25



## Introduction

Le module ESP8266 est un système sur puce (SoC). Il se compose d'un microcontrôleur 32 bits Tensilica L106 et d'un émetteur-récepteur Wi-Fi. Il possède 11 broches d'entrée/sortie à usage général, ou pour faire court des broches GPIO, et une entrée analogique. Cela signifie qu'il peut être programmé comme n'importe quel autre microcontrôleur. La meilleure chose à propos de l'ESP8266 est qu'il dispose d'une communication Wi-Fi, ce qui signifie qu'il peut être connecté au Wi-Fi, ou à Internet, héberger un serveur web avec de vraies pages web, se connecter à un smartphone, etc. Il prend en charge les protocoles de réseau tels que Wi-Fi, TCP, UDP, HTTP, DNS, etc.

Le module AZ-Delivery D1 Mini est une carte de développement basée sur la puce ESP8266. Il dispose de 11 broches d'entrée/sortie numériques et d'une broche d'entrée analogique. Toutes les broches d'entrée/sortie numériques ont des capacités d'interruption, de pwm, d'I2C et de 1 fil, en logiciel. La gamme de tension d'entrée analogique est comprise entre 0V et 3.3V DC. Le module utilise le port microUSB et la puce CH340G avec un circuit programmateur pour programmer l'ESP8266. De plus, le port microUSB fournit une alimentation au module.



## Caractéristiques de l'ESP8266

- » 802.11 b/g/n
- » MCU 32 bits intégré à faible consommation
- » CAN 10 bits intégré
- » Pile de protocoles TCP/IP intégrée
- » Commutateur TR, balun, LNA, amplificateur de puissance et réseau d'adaptation intégrés.
- » PLL, régulateurs et unités de gestion de l'alimentation intégrés
- » Supporte la diversité d'antenne
- » Wi-Fi 2.4 GHz, support WPA/WPA2
- » Supporte les modes de fonctionnement STA/AP/STA+AP
- » Supporte la fonction Smart Link pour les appareils Android et iOS
- » SDIO 2.0, (H) SPI, UART, I2C, I2S, IRDA, PWM, GPIO
- » STBC, 1x1 MIMO, 2x1 MIMO
- » Agrégation A-MPDU & A-MSDU et intervalle de protection de 0.4s
- » Consommation en sommeil profond <10uA, courant de fuite à la mise hors tension <5uA
- » Réveil et transmission de paquets en < 2ms
- » Consommation en mode veille < 1,0 mW (DTIM3)
- » Puissance de sortie de +20dBm en mode 802.11b
- » Température de fonctionnement : -40 °C ~ 125 °C

## Le module D1 Mini

Le module D1 Mini est livré non soudé avec une paire de connecteurs mâles à huit broches, une paire de connecteurs femelles à huit broches et une paire de connecteurs femelles à huit broches avec des pattes extra longues (que l'on peut voir sur l'image de couverture).

### Caractéristiques

- » Tension de fonctionnement : 3.3V DC
- » Puce principale : ESP8266
- » Vitesse d'exécution : 80MHz / 160MHz
- » Flash : 4MB
- » Broches d'E/S numériques : 11
- » Broches d'entrée analogique : 1
- » Gamme de tension d'entrée analogique : from 0V to 3.3V DC
- » Port: microUSB
- » USB chip: CH340 chip
- » LED embarquée: » connecté à la
- Dimensions: 25 x 35 x 6mm [0.98 x 1.4 x 0.24in]
- » Courant maximal par broche d'E/S numérique unique: 12mA



## Broches d'E/S numériques

Comme toute carte microcontrôleur, le module D1 Mini possède des broches d'entrée/sortie numériques ou GPIO - General Purpose Input/Output pins. Comme leur nom l'indique, elles peuvent être utilisées comme entrées numériques pour lire une tension numérique, ou comme sorties numériques pour délivrer soit 0V (courant d'absorption) soit 3,3V (courant de source).

Le module D1 Mini possède un microcontrôleur qui fonctionne dans une gamme de tension de 0V - 3.3V.

**Le courant maximum qui peut être tiré d'une seule broche GPIO est de 12mA !**

**NOTE : Les broches du module D1 Mini ne sont pas tolérantes à 5V, appliquer plus de 3.6V sur n'importe quelle broche pourrait endommager la puce !**

GPIO1 et GPIO3 sont utilisés comme TX et RX du port série matériel (UART), donc dans la plupart des cas, ils ne peuvent pas être utilisés comme une E/S normale pendant l'envoi/la réception de données série.

Le module D1 Mini possède une LED intégrée connectée à la broche GPIO2.



## PWM

Contrairement à la plupart des puces Atmel (microcontrôleur), le module D1 Mini ne supporte pas le PWM matériel. Cependant, le PWM logiciel est supporté sur toutes les broches numériques. La gamme PWM par défaut est de 10 bits à 1kHz, mais cela peut être modifié (jusqu'à 14 bits à 1kHz).

## Entrée analogique

Le module D1 Mini dispose d'une seule entrée analogique, avec une gamme de tension d'entrée de 0V - 3,0V. Si une tension supérieure à 3,3 V est appliquée, la puce pourrait être endommagée. Le convertisseur analogique-numérique (ADC) a une résolution de 10 bits.

## Série

Le module D1 Mini possède deux UARTS (ports série) matériels : UART0 sur les broches 1 et 3 (TX0 et RX0 respectivement), et UART1 sur les broches 2 et 8 (TX1 et RX1 respectivement). Cependant, GPIO8 est utilisé pour connecter la puce flash. Cela signifie que UART1 ne peut que transmettre des données. Dans la plupart des cas, un seul port UART est plus que suffisant.

UART0 possède également un contrôle de flux matériel sur les broches 15 et 13 (RTS0 et CTS0 respectivement). Ces deux broches peuvent également être utilisées comme alternative aux broches TX0 et RX0.



## L'I2C

Le module D1 Mini n'a pas d'interface matérielle I2C ou TWI (Two Wire Interface), mais elle est implémentée en logiciel. Cela signifie que deux broches numériques quelconques peuvent être utilisées comme broches I2C. Par défaut, la bibliothèque I2C utilise GPIO4 comme SDA et GPIO5 comme SCL (la fiche technique spécifie GPIO2 comme SDA et GPIO14 comme SCL). La vitesse maximale de l'horloge I2C est d'environ 450kHz.

## SPI

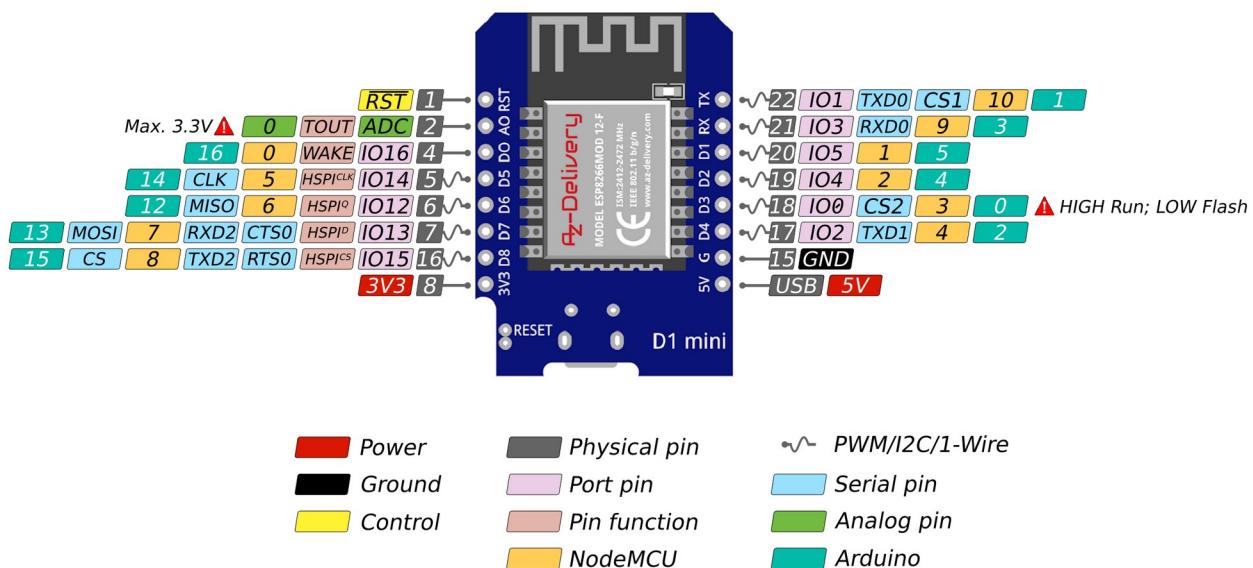
Le module D1 Mini met à la disposition de l'utilisateur une connexion SPI, appelée HSPI. Elle peut être utilisée en mode Slave et Master (en logiciel !).

Elle utilise :

- GPIO14 comme a clock – CLK,
- GPIO12 comme MISO,
- GPIO13 comme MOSI
- GPIO15 comme Slave Select - SS.

## Le pinout

Le module D1 Mini comporte deux rangées de huit broches (seize broches au total). Le brochage du module est présenté sur l'image suivante :





## Module D1 Mini - Logiciel

La plupart des fonctionnalités du microcontrôleur de l'ESP utilisent exactement la même syntaxe qu'une carte microcontrôleur normale, ce qui facilite considérablement la prise en main.

### Broches d'E/S numériques

Tout comme avec une carte microcontrôleur ordinaire, la fonction de la broche peut être définie en utilisant la ligne de code suivante : `pinMode(pin, mode)`

où `pin` est le nom de la broche `GPIO`, et le `mode` peut être soit `INPUT` (qui est le défaut), soit `OUTPUT`, soit `INPUT_PULLUP` pour activer les résistances pull-up intégrées pour les broches `GPIO0` - `15`. Pour activer la résistance pull-down pour `GPIO16` utilisez `INPUT_PULLDOWN_16`.

Pour régler une broche de sortie sur `HIGH` (3.3V) or `LOW` (0V), utilisez la ligne de code suivante : `digitalWrite(pin, value)` où `pin` iest le nom de la broche `GPIO`, et `value` either `1` ou `0` (ou `HIGH` et `LOW`).

Pour lire une entrée, utilisez la ligne de code suivante : `digitalRead(pin)`



Pour activer le PWM sur une certaine broche, utilisez la ligne de code suivante : `analogWrite(pin, value)` où `pin` est le nom de la broche `GPIO`, et `value` un nombre entre `0` et `1023`.

La gamme de la sortie PWM peut être modifiée en utilisant la ligne de code suivante : `analogWriteRange(new_range)`

La fréquence du PWM peut être modifiée en utilisant la ligne de code suivante : `analogWriteFreq(new_frequency)` où `new_frequency` doit être comprise entre `100Hz` et `1000Hz`.

## Broche d'entrée analogique

Comme sur toute carte microcontrôleur, la fonction `analogRead(A0)` peut être utilisée pour obtenir la tension analogique sur l'entrée analogique (`0 = 0V, 1023 = 1.0V`).

Le module D1 Mini peut également utiliser l'ADC pour mesurer la tension d'alimentation (VCC). Pour ce faire, incluez `ADC_MODE(ADC_VCC)` en haut de votre sketch, et utilisez `ESP.getVcc()` pour obtenir réellement la tension `ESP.getVcc()` pour obtenir réellement la tension.

**NOTE:** : Si une broche d'entrée analogique est utilisée pour lire la tension d'alimentation, rien d'autre ne peut être connecté à la broche d'entrée analogique !



## Communication en série

Pour utiliser *UART0* (*TX* = *GPIO1*, *RX* = *GPIO3*), utilisez l'objet *Serial*, comme sur une carte microcontrôleur *ATMega*:

```
Serial.begin(baud_rate)
```

Pour utiliser les broches alternatives (*TX* = *GPIO15*, *RX* = *GPIO13*), utilisez la ligne de code suivante : `Serial.swap()`, après `Serial.begin()`

Pour utiliser *UART1* (*TX* = *GPIO2*), utilisez l'objet *Serial1*.

**NOTE:** Toutes les fonctions de flux microcontrôleur **ATMega328**, comme *read()*, *write()*, *print()*, *println()*, sont également supportées.

## Les interfaces I2C et SPI

Pour l'interface I2C et SPI, utilisez la syntaxe par défaut de la bibliothèque Arduino-library.



## Partage du temps CPU avec la partie RF

Une chose à garder à l'esprit lors de l'écriture de programmes pour le module D1 Mini (ESP8266) est que le sketch doit partager les ressources (temps CPU et mémoire) avec les piles wifi et TCP (le logiciel qui tourne en arrière-plan et gère toutes les connexions wifi et IP). Si le code prend trop de temps à s'exécuter, et ne laisse pas les piles TCP faire leur travail, le programme peut se planter, ou les données peuvent être perdues. Il est préférable de maintenir le temps d'exécution de la boucle sous quelques centaines de millisecondes. Chaque fois que la boucle principale est répétée, un sketch cède au wifi et au TCP pour gérer toutes les demandes du wifi et du TCP. Si la boucle prend plus de temps que cela, le temps CPU doit être explicitement donné aux piles wifi/TCP, en utilisant notamment `delay(0)` or `yield()`. Si le WDT souple est désactivé, après un peu plus de 8 secondes, le WDT matériel réinitialise la puce. Du point de vue du microcontrôleur cependant, 3 secondes est un temps très long (240 millions de cycles d'horloge), donc à moins que des calculs extrêmement lourds ne soient effectués, ou que des chaînes de caractères extrêmement longues ne soient envoyées en série, le sketch ne sera pas affecté par cela. Gardez juste à l'esprit d'ajouter le `yield()` inside the `for` or `while` loops that could take longer than, say 100ms.

## Comment configurer Arduino IDE

Si l'IDE Arduino n'est pas installé, suivez le [link](#) et téléchargez le fichier d'installation pour le système d'exploitation de votre choix.

Download the Arduino IDE

The screenshot shows the official Arduino website's download page. On the left, there's a large teal circular logo with a white infinity symbol containing a minus sign on the left and a plus sign on the right. To its right, the text "ARDUINO 1.8.9" is displayed in bold capital letters. Below this, a paragraph of text describes the software: "The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions." To the right of this text, there are download links for different operating systems. The Windows section includes links for "Windows Installer" (for Windows XP and up) and "Windows ZIP file" (for non-admin install). The Mac OS X section includes a link for "Windows app" (Requires Win 8.1 or 10) with a "Get" button. The Linux section includes links for "Linux 32 bits", "Linux 64 bits", "Linux ARM 32 bits", and "Linux ARM 64 bits". At the bottom of the sidebar, there are links for "Release Notes", "Source Code", and "Checksums (sha512)".

Pour les utilisateurs de Windows, double-cliquez sur le fichier .exe téléchargé et suivez les instructions de la fenêtre d'installation.

# Az-Delivery

Pour les utilisateurs de *Linux*, téléchargez un fichier portant l'extension *.tar.xz*, qui doit être extrait. Lorsqu'il est extrait, allez dans le répertoire extrait et ouvrez le terminal dans ce répertoire. Deux scripts *.sh* doivent être exécutés, le premier appelé *arduino-linux-setup.sh* et le deuxième appelé *install.sh*.

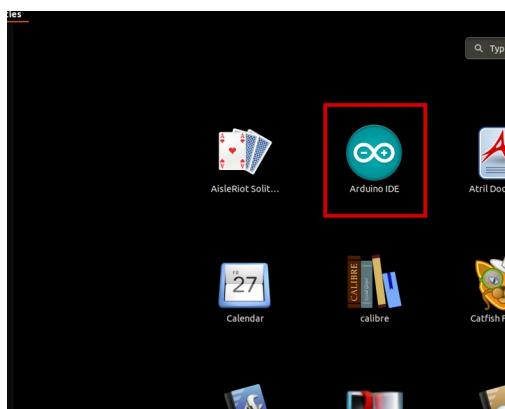
Pour exécuter le premier script dans le terminal, ouvrez le terminal dans le répertoire extrait et exécutez la commande suivante :

**sh arduino-linux-setup.sh user\_name**

*user\_name* - est le nom d'un superutilisateur dans le système d'exploitation *Linux*. Un mot de passe pour le superutilisateur doit être saisi au moment du lancement de la commande. Attendez quelques minutes pour que le script complète tout.

Le deuxième script appelé script *install.sh* doit être utilisé après l'installation du premier script. Exécutez la commande suivante dans le terminal (répertoire extrait) : **sh install.sh**

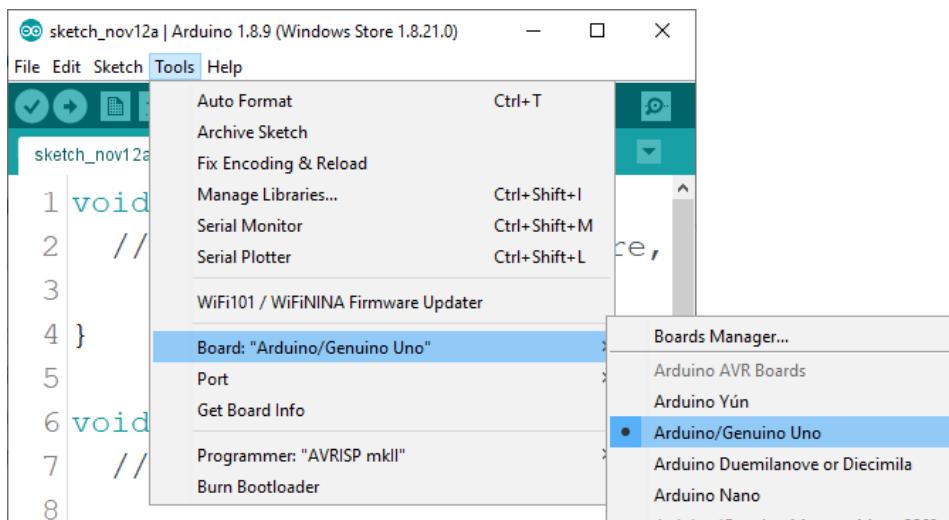
Après l'installation de ces scripts, allez dans le dossier *All Apps*, où est installé l'*Arduino IDE*.



# Az-Delivery

Presque tous les systèmes d'exploitation sont livrés avec un éditeur de texte préinstallé (par exemple, Windows est livré avec Notepad, *Linux Ubuntu* avec *Gedit*, *Linux Raspbian* avec *Leafpad*, etc.) Tous ces éditeurs de texte conviennent parfaitement à l'objectif de l'eBook.

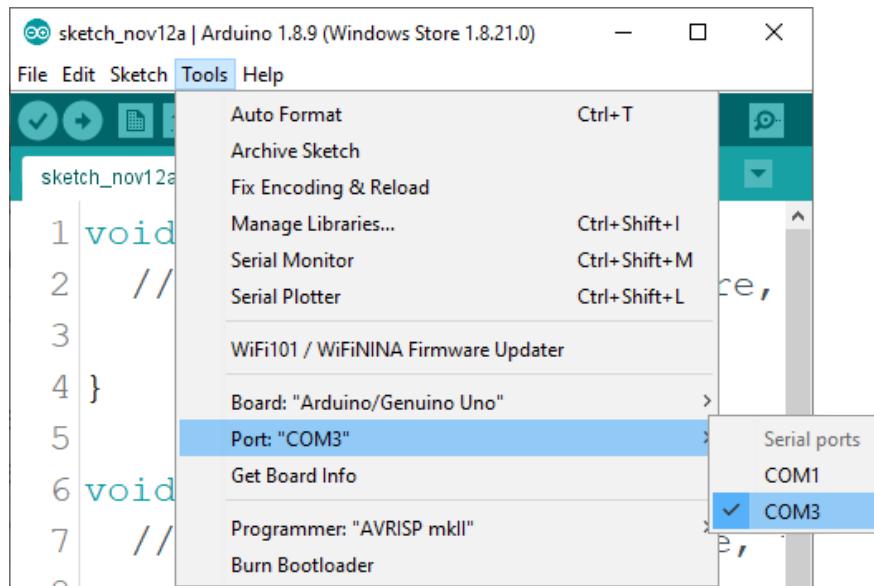
La prochaine étape est de vérifier si votre PC peut détecter une carte Atmega328p. Ouvrez l'IDE Arduino fraîchement installé, et allez dans : *Tools > Board > {your board name here}* devrait être l'*Arduino/Genuino Uno*, comme on peut le voir sur l'image suivante :



Le port auquel la carte microcontrôleur est connectée doit être sélectionné. Aller à : *Tools > Port > {le nom du port va ici}* et lorsque la carte microcontrôleur est connectée au port USB, le nom du port peut être vu dans le menu déroulant de l'image précédente.

# Az-Delivery

Si l'IDE Arduino est utilisé sous Windows, les noms des ports sont les suivants :



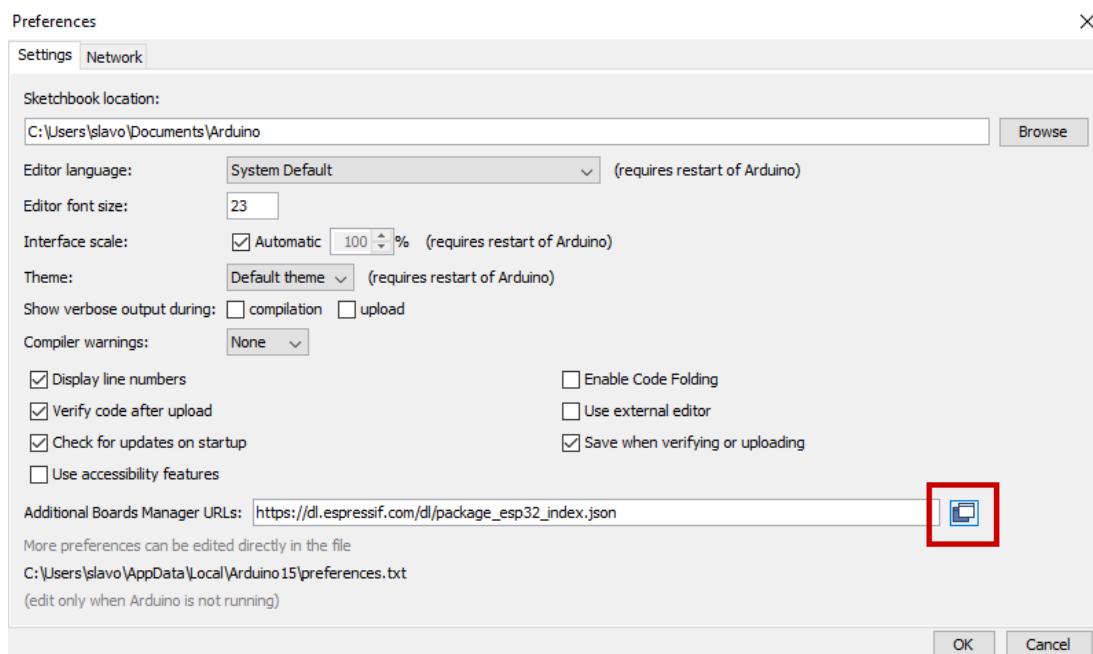
Pour les utilisateurs de Linux, par exemple, le nom du port est `/dev/ttUSBx`, où x représente un nombre entier entre 0 et 9.



## D1 Mini avec Arduino IDE

Pour configurer l'Arduino IDE afin de pouvoir programmer le module D1 Mini via l'Arduino IDE, suivez les étapes suivantes.

Tout d'abord, installez le processeur ESP8266. Pour l'installer, ouvrez l'IDE Arduino et allez dans : *File > Preferences* et trouvez le champ *Additional URLs*.



Ensuite, copiez l'URL suivante :

[https://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](https://arduino.esp8266.com/stable/package_esp8266com_index.json)

et le coller dans le champ *Additional URLs*.

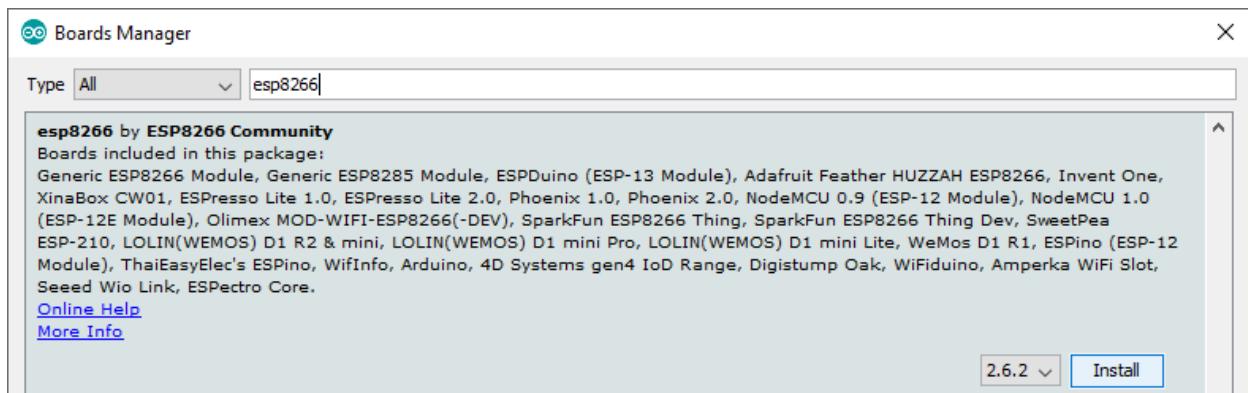
# Az-Delivery

S'il y a un ou plusieurs liens dans le champ *Additional URLs*, ajoutez simplement une virgule après le dernier lien, collez un nouveau lien après une virgule et cliquez sur le bouton *OK*. Fermez ensuite l'IDE Arduino.



Ouvrez à nouveau microcontrôleur IDE et allez à :  
*Tools > Board > Boards Manager*

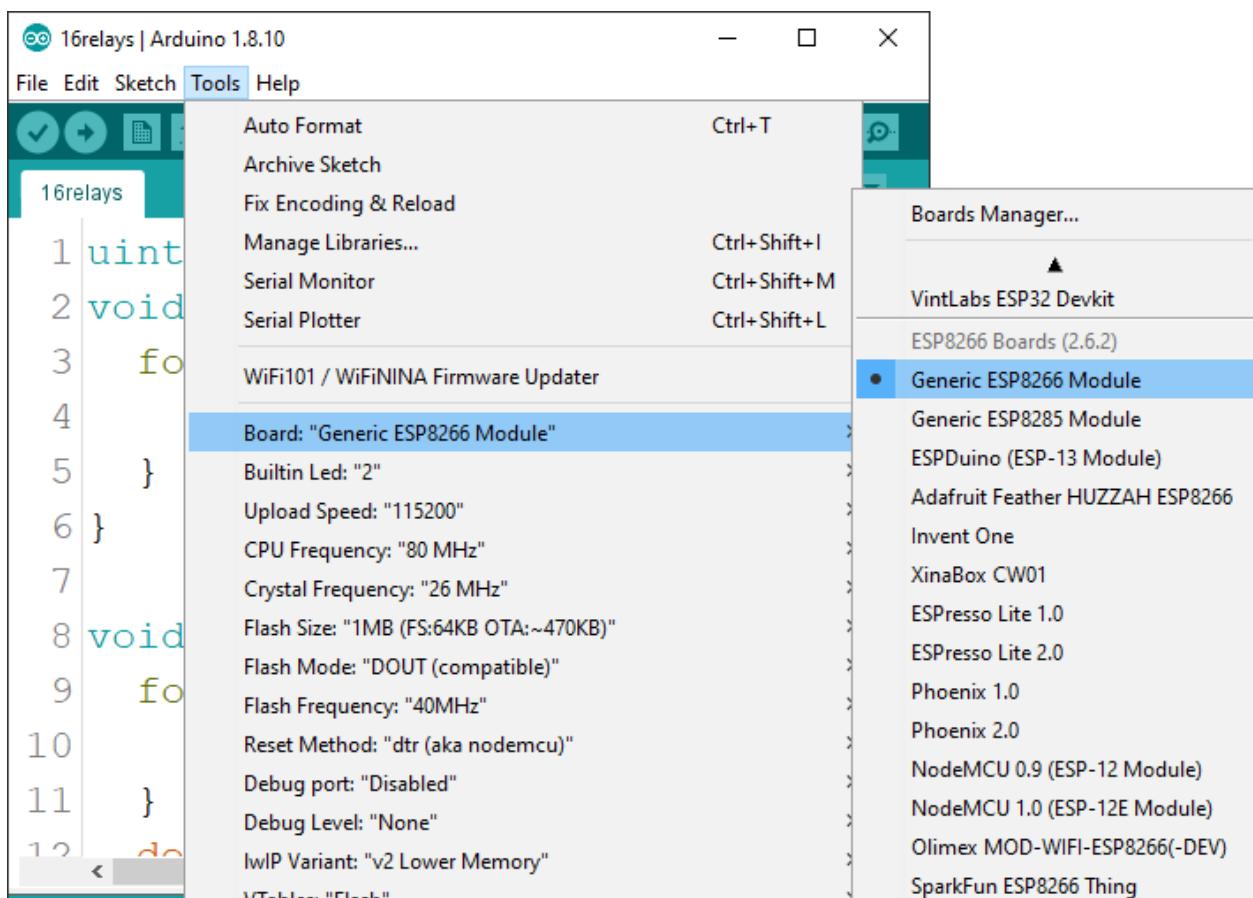
Lorsque la nouvelle fenêtre s'ouvre, tapez *esp8266* dans la boîte de recherche et installez la carte appelée *esp8266* fabriquée par la *ESP8266 Community*, comme le montre l'image ci-dessous :



Maintenant, le core ESP8266 est installé.

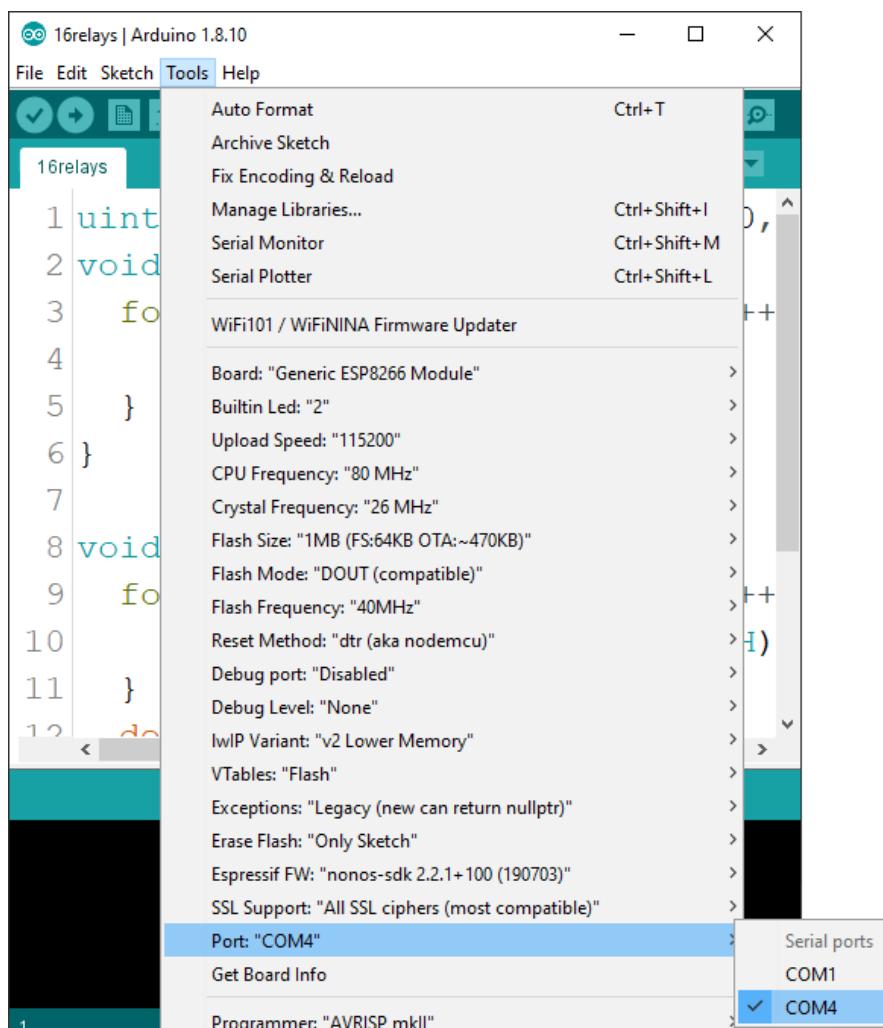
# Az-Delivery

L'étape suivante consiste à sélectionner la bonne carte dans l'IDE Arduino. Ouvrez Arduino IDE et allez dans : *Tools > Board > {board name}* et sélectionnez le premier module *ESP8266* générique comme indiqué sur l'image suivante :



# Az-Delivery

Ensuite, sélectionnez le port sur lequel la carte du module D1 Mini est connectée. Allez sur : *Tools > Port > {le nom du port va ici}* Si la carte du module D1 Mini est connectée sur le port USB, il devrait y avoir des noms de port. Dans cet eBook, l'IDE Arduino est utilisé sous Windows, les noms de port sont les suivants :



Pour les utilisateurs de Linux, le nom du port est `/dev/ttyUSBx` par exemple, où "x" représente un nombre entier spécifique compris entre 0 et 9.



## Blink + PWM + Serial exemples de sketchs

### Exemple de sketch Blink

Il existe un exemple de sketch blink fourni avec la bibliothèque de la carte *ESP8266*. Pour l'ouvrir, allez dans : *Files > Examples > ESP8266 > Blink*

```
void setup() {  
    pinMode(LED_BUILTIN, OUTPUT);      // Initialize the LED_BUILTIN  
}  
  
void loop() {  
    digitalWrite(LED_BUILTIN, LOW);    // Turn the LED on  
    // Note that LOW is the voltage level  
    // but actually the LED is on; this is because  
    // it is active LOW on the ESP  
    delay(1000);                    // Wait for a second  
    digitalWrite(LED_BUILTIN, HIGH);   // Turn the LED off  
    // by making the voltage HIGH  
    delay(2000);                    // Wait for two seconds  
}
```

Pour le module D1 Mini, *LED\_BUILTIN* est égal au nombre 2, ce qui signifie que la *LED* embarquée est connectée à la broche *GPIO2*. Pour allumer la *LED*, la broche *GPIO2* doit être mise à l'état *LOW*, et pour l'éteindre, la broche *GPIO2* doit être mise à l'état *HIGH*.

# Az-Delivery

## Exemple de sketch PWM logiciel

```
int brightness = 1; // do not set it to the zero
                     // zero disables the PWM on a specific pin
uint8_t fadeAmount = 5;
void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
    analogWrite(LED_BUILTIN, 200); // high brightness
    delay(1000);
    analogWrite(LED_BUILTIN, 500);
    delay(1000);
    analogWrite(LED_BUILTIN, 800);
    delay(1000);
    analogWrite(LED_BUILTIN, 1000); // low brightness
    delay(1000);

    // fading led
    while (1) {
        analogWrite(LED_BUILTIN, brightness);
        brightness = brightness + fadeAmount;
        if (brightness < 0 || brightness >= 1023) {
            fadeAmount = -fadeAmount;
        }
        delay(10);
    }
}
```

# Az-Delivery

Pour utiliser le PWM sur le module D1 Mini (ESP8266), on utilise la même fonction que sur les cartes microcontrôleur :

*analogWrite(pin, value)* où *pin* est le nom de la broche *GPIO*, toute broche *GPIO* libre du module D1 Mini. *Value* est le cycle d'utilisation, une valeur entre.

**N'utilisez pas le chiffre zéro comme *value*, car il désactive la fonction PWM sur une broche spécifique !**

Plus la valeur zéro est proche, plus le niveau de luminosité d'une *LED* est élevé et plus la valeur 1023 est proche, plus le niveau de luminosité d'une *LED* est faible.

Pour que la *LED* s'éteigne, il faut utiliser la boucle *while(1)* à l'intérieur de la fonction *loop()*, sinon cela ne fonctionnera pas.

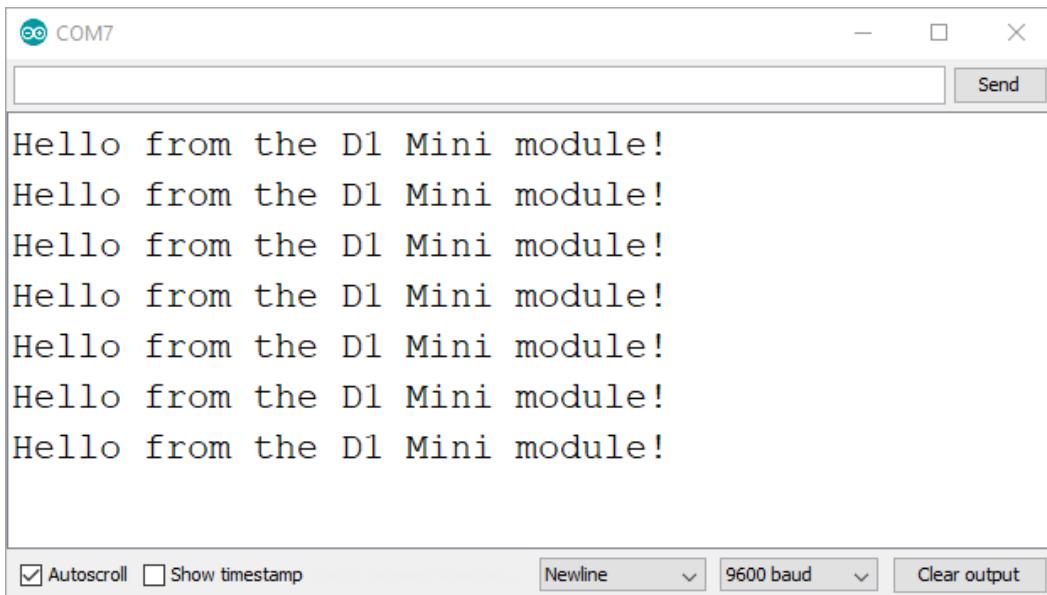
# Az-Delivery

## Exemple de sketch de communication série

```
void setup() {
    Serial.begin(9600);
}

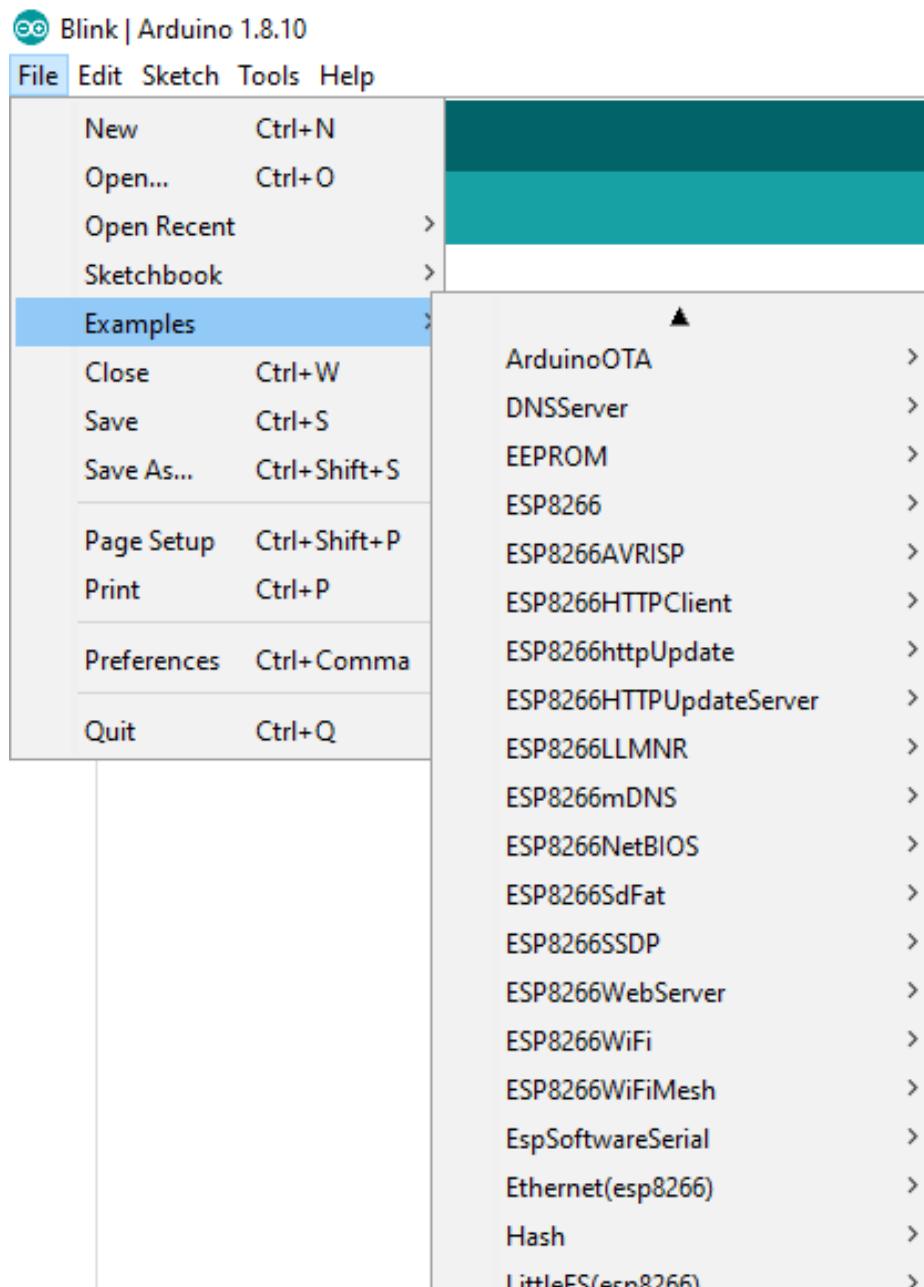
void loop() {
    Serial.println("Hello from the D1 Mini module!");
    delay(1000);
}
```

Téléchargez le sketch sur le module D1 Mini et ouvrez le (*Tools > Serial Monitor*). Le résultat devrait ressembler à l'image suivante :



# Az-Delivery

La bibliothèque de la carte *ESP8266* contient de nombreux autres exemples de sketchs. La partie wifi du module *D1 Mini* pourrait être testée à partir de là. Elle n'est pas couverte dans cet eBook.





Il est maintenant temps d'apprendre et de réaliser vos propres projets. Vous pouvez le faire à l'aide de nombreux exemples de scripts et autres tutoriels, que vous trouverez sur Internet.

**Si vous êtes à la recherche de microélectronique et accessoires de haute qualité, AZ-Delivery Vertriebs GmbH est l'entreprise idéale pour vous les procurer. Vous recevrez de nombreux exemples d'application, des guides d'installation complets, des eBooks, des bibliothèques et l'assistance de nos experts techniques.**

<https://az-delivery.de>

Amusez-vous !

Impressum

<https://az-delivery.de/pages/about-us>