

Installer le logiciel Arduino IDE sous Linux

Cet article a été testé avec les distributions suivantes

- **Linux Mint 19 64 bits**
- **Linux Debian 9.6 (stretch) 64 bits**
- **MX Linux 18.3** basé sur Linux Debian 9
- **Zorin OS 16.1** basé sur Linux Debian



Je privilégie une installation "manuelle" de l'application **Arduino IDE** afin d'utiliser une des dernières versions. Certaines options et paramètres n'apparaissent pas dans les versions packagées des distributions. J'ai eu le cas avec **Debian 9.6** où je ne pouvais pas utiliser des cartes ESP 8266.

Date de relecture : 15 août 2022

Il faut télécharger la version 1.8.19. Vous pouvez vérifier la dernière version disponible depuis le site <https://www.arduino.cc/en/Main/Software>. Adaptez les informations ci-dessous suivant les informations trouvées sur le site arduino.cc.



1. Télécharger Arduino IDE Software

Depuis mon dossier **home** ou un autre dossier personnel, je télécharge le programme **Arduino IDE**. Je préfère l'effectuer depuis une ligne de commande.

```
wget -O arduino.tar.xz  
https://downloads.arduino.cc/arduino-1.8.19-linux64.tar.xz
```



```
cedric@qigajside:~$ wget -O arduino.tar.xz https://www.arduino.cc/download.php?f=/arduino-1.8.7-linux64.tar.xz
--2018-11-16 07:16:39-- https://www.arduino.cc/download.php?f=/arduino-1.8.7-linux64.tar.xz
Résolution de www.arduino.cc (www.arduino.cc) : 18.215.226.55, 18.235.138.153
Connexion à www.arduino.cc (www.arduino.cc)[18.215.226.55]:443_ connecté.
requête HTTP transmise, en attente de la réponse_ 302 Moved Temporarily
Emplacement : https://downloads.arduino.cc/arduino-1.8.7-linux64.tar.xz [suivant]
--2018-11-16 07:16:39-- https://downloads.arduino.cc/arduino-1.8.7-linux64.tar.xz
Résolution de downloads.arduino.cc (downloads.arduino.cc) : 137.74.24.249
Connexion à downloads.arduino.cc (downloads.arduino.cc)[137.74.24.249]:443_ connecté.
requête HTTP transmise, en attente de la réponse_ 200 OK
Taille : 117717256 (112M) [application/octet-stream]
Sauvegarde en : = arduino.tar.xz =

arduino.tar.xz 100%[=====>] 112,26M 895KB/s in 2m 18s
2018-11-16 07:18:49 (883 KB/s) = arduino.tar.xz = sauvegardé [117717256/117717256]

cedric@qigajside:~$
```

2. Déployer Arduino IDE Software

ligne 1 : déployer Arduino IDE dans un dossier accessible à tous : /opt/. L'archive **tar** va extraire les fichiers dans un dossier comportant le numéro de version.

ligne 3 : si un lien /opt/arduino/ existait, détruit le lien sur /opt/arduino avec **unlink**

ligne 4 : réaliser un **alias** vers le dossier /opt/arduino. De cette manière, on s'affranchit du numéro de version. Nous pouvons également déployer de nouvelles versions sans perturber la précédente.

ligne 5 : exécuter le script `install.sh` qui permet de créer un raccourci sur le bureau graphique.

1. `sudo tar -xvf arduino.tar.xz -C /opt`
2. `sudo chown -R root:root /opt/arduino-1.8.19`
3. `sudo unlink /opt/arduino`
4. `sudo ln -s /opt/arduino-1.8.19/ /opt/arduino`
5. `sudo /opt/arduino/install.sh`

3. Problèmes rencontrés après le déploiement

3.1 Droits sur tty et dialout

Il faut ajouter l'utilisateur de l'application **Arduino IDE** aux groupes `tty` et `dialout`. L'exemple suivant permet d'ajouter l'utilisateur `cedric` aux groupes `tty` et `dialout`.

```
sudo usermod -a -G tty $USER
sudo usermod -a -G dialout $USER
```

Si vous venez de saisir les lignes de commandes `usermod`, vous devez redémarrer.

3.2 Version de Java incorrecte

Le message suivant peut apparaître dans l'application **Arduino IDE** :

```
java.lang.NullPointerException thrown while loading gnu.io.RXTXCommDriver
```

Il se peut que la version de Java ne soit pas bonne. Vous pouvez le vérifier avec `java -version`. Toutefois, un java est intégré avec l'application **Arduino IDE**. (`/opt/arduino/java/bin/java -version` me renvoie la version java version "1.8.0_191").

Il faudra peut être toutefois changer les paramètres java de l'OS. J'ai testé avec OpenJDK Java 8 avec succès. Par contre Java 10 et 11 ne sont pas compatibles.

Pour procéder à l'installation de Java 8 et choisir par défaut cette version :

```
sudo apt-get update

sudo apt-get install openjdk-8-jdk

sudo update-alternatives --config java
```

```
cedric@gigajside: ~
Fichier Édition Affichage Rechercher Terminal Aide
cedric@gigajside:~$ sudo update-alternatives --config java
Il existe 1 choix pour l'alternative java (qui fournit /usr/bin/java).

  Sélection  Chemin                                     Priorité  État
-----
0           /usr/lib/jvm/java-8-oracle/jre/bin/java    1081     mode automatique
* 1         /usr/lib/jvm/java-8-oracle/jre/bin/java    1081     mode manuel

Appuyez sur <Entrée> pour conserver la valeur par défaut[*] ou choisissez le numéro sélectionné :1
cedric@gigajside:~$
```

3.3 Serial port not selected

Dans certains cas, sous Mint, il faut désinstaller `brltty`

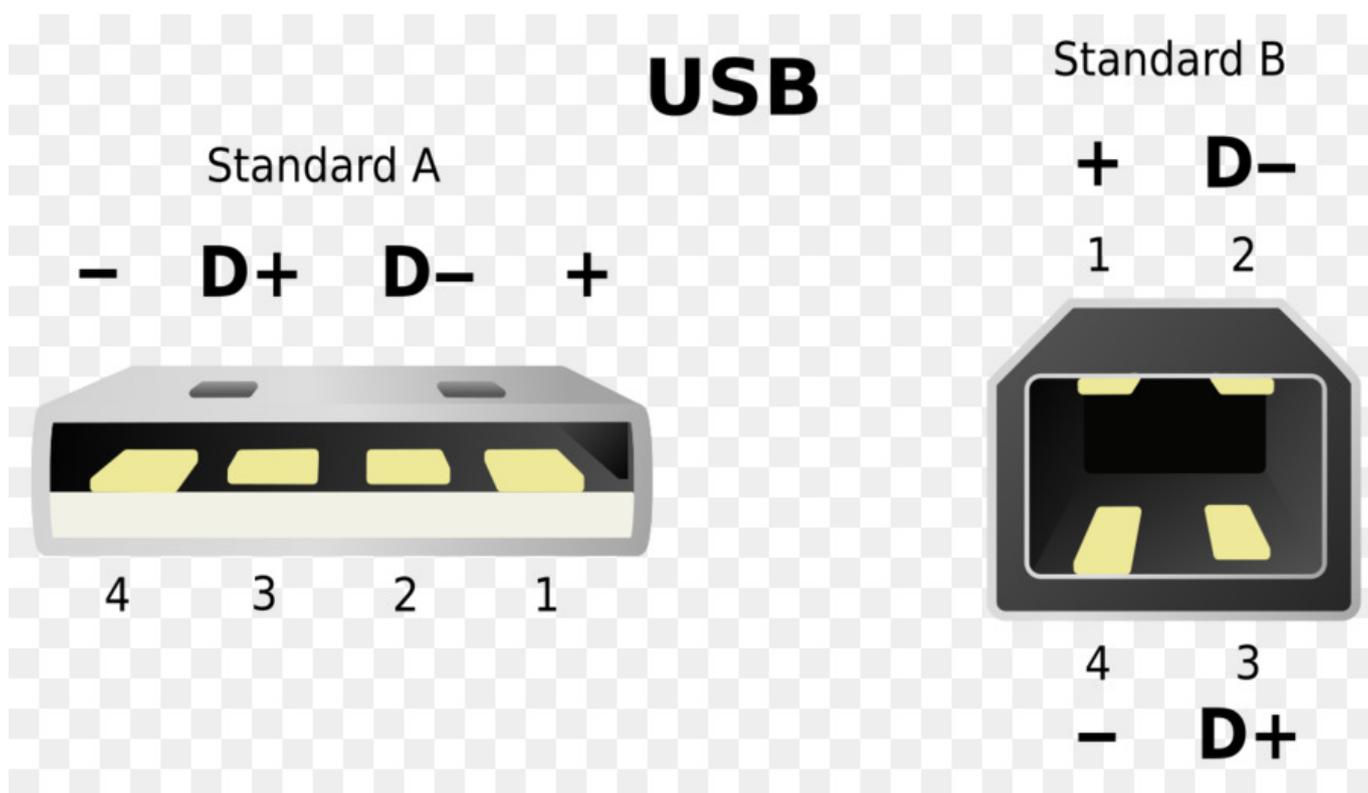
```
sudo apt-get remove brltty
```

3.4 Aucune carte n'est détectée



Le câble USB peut être également non câblé pour les données (data)

Pour rappel, voici le cablage d'un port USB



4. Exécuter votre premier programme

Voici donc notre premier programme (celui qui est décrit dans la vidéo).

Cette suite d'instructions va faire clignoter une LED branchée sur la broche 13 de l'Arduino toutes les secondes.

Il n'est pas nécessaire de brancher une LED, car rappelez-vous, la LED sur la carte, mentionnée dans la [présentation de l'arduino](#), servira dans notre exemple sans forcément réalisée un montage électronique.

Lorsque vous utilisez le logiciel Arduino, il peut être trouvé en cliquant sur Fichier → Exemples → 01.Basics → Blink.

Vous pouvez également copier le code suivant dans votre éditeur :

[basics-blink.c](#)

```
1. /*
2.   Clignotement
3.   Allume la LED pendant 1 seconde,
4.   puis l'éteint pendant 0,5 seconde.
5. */
6.
7. // Numéro de la broche à laquelle est
```

```
8. // connectée la LED
9. int led = 13;
10.
11. // le code dans cette fonction est exécuté une fois au début
12. void setup() {
13.   // indique que la broche de la LED une sortie :
14.   // on va modifier sa tension
15.   pinMode(led, OUTPUT);
16. }
17.
18. // le code dans cette fonction est exécuté en boucle
19. void loop() {
20.   digitalWrite(led, HIGH); // allumer la LED (tension 5V sur la
    broche)
21.   delay(1000);           // attendre 1000ms = 1s
22.   digitalWrite(led, LOW); // éteindre la LED (tension 0V sur la
    broche)
23.   delay(1000);           // attendre à nouveau 1seconde
24. }
```

Deux déclaration de fonctions sont obligatoires :

- setup
- loop

Dans ce bout de code, nous appelons trois fonctions :

- digitalWrite
- pinMode
- delay

Pour exécuter ce bout de code, il faudra brancher l'Arduino sur port USB et sélectionner dans le menu Outils ⇒ Port, sélectionner /dev/ttyACM0.

From:

<https://www.abonnel.fr/> - **notes informatique & technologie**

Permanent link:

<https://www.abonnel.fr/electronique/esp/installer-logiciel-arduino-ide>

Last update: **2020/11/12 13:36**

