

Comment installer ARDUINO Gnu/Linux

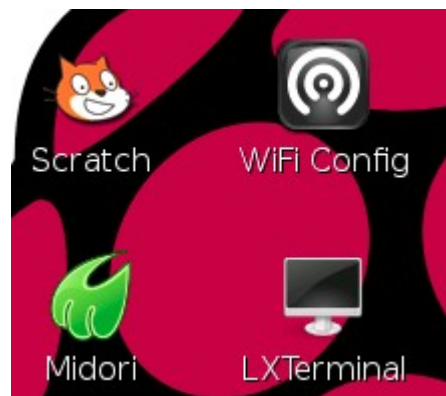
Préambule

Sous Ubuntu ou sur d'autres distributions (Debian, Linux Mint, Raspbian...) il est possible d'installer directement l'environnement de développement depuis les dépôts. Or la version étant assez vieille (1:1.0.5) Il est préférable d'utiliser une version plus récente du [site officiel](#) selon la méthode ci-dessous.

1) Le terminal pour saisir des commandes

Pour l'installation, nous allons essentiellement utiliser le terminal en saisissant des commandes. Soyez rassurés, les commandes seront simples et bien expliquées. Vous pouvez faire du copier coller avec la souris.

Si vous avez le système d'exploitation installé [Raspbian](#), alors il suffit de cliquer sur l'icône **LXTerminal** sur le bureau :



Il est également possible de le retrouver dans le menu des programmes, dans la catégorie « Accessoires ».

Une fois ouvert, vous devriez vous retrouver face à une **fenêtre noire** comme celle-ci :



Vous êtes maintenant prêt à entrer des commandes.

2) Installer l'environnement Arduino (ancienne version, installation rapide)

Il faut que votre Raspberry Pi ou ordinateur soit connecté à Internet.

Taper les commandes suivantes et valider par la touche entrée :

```
sudo apt-get update
```

puis

```
sudo apt-get install arduino -y
```

```
root@raspberrypi-fm:~# apt-get install arduino
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  arduino-core avr-libc avrdude binutils-avr ca-certificates ca-certificates-java default-jre
  default-jre-headless extra-xdg-menus gcc-avr icedtea-6-jre-cacao icedtea-6-jre-jamvm
  icedtea-netx icedtea-netx-common java-common libatk-wrapper-java libatk-wrapper-java-jni
  libftdi1 libjna-java libnspr4 libnss3 libnss3-ld librxtx-java openjdk-6-jre
  openjdk-6-jre-headless openjdk-6-jre-lib tzdata-java
Paquets suggérés :
  arduino-mk avrdude-doc task-c-devel gcc-doc gcc-4.2 equivs libjna-java-doc icedtea-plugin
  libnss-mdns sun-java6-fonts fonts-ipafont-gothic fonts-ipafont-mincho ttf-wqy-microhei
  ttf-wqy-zenhei ttf-indic-fonts
Les NOUVEAUX paquets suivants seront installés :
  arduino arduino-core avr-libc avrdude binutils-avr ca-certificates-java default-jre
  default-jre-headless extra-xdg-menus gcc-avr icedtea-6-jre-cacao icedtea-6-jre-jamvm
  icedtea-netx icedtea-netx-common java-common libatk-wrapper-java libatk-wrapper-java-jni
  libftdi1 libjna-java libnspr4 libnss3 libnss3-ld librxtx-java openjdk-6-jre
  openjdk-6-jre-headless openjdk-6-jre-lib tzdata-java
Les paquets suivants seront mis à jour :
  ca-certificates
1 mis à jour, 27 nouvellement installés, 0 à enlever et 154 non mis à jour.
Il est nécessaire de prendre 57,7 Mo dans les archives.
Après cette opération, 168 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer [O/n] ?
```

L'installation est presque terminée.

Il reste une dernière étape pour que la carte puisse bien communiquer avec l'ordinateur. Il faut ajouter l'utilisateur au group dialout. En considérant que l'utilisateur s'appelle paul.

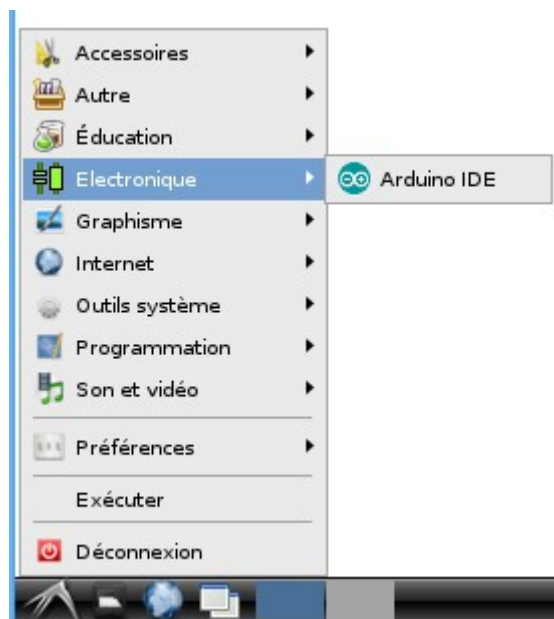
Pour connaître le nom de l'utilisateur, dans le terminal taper : `whoami`

Puis exécuter les deux commandes :

```
sudo usermod -a -G dialout paul
```

```
sudo usermod -a -G tty paul
```

Voilà l'installation est terminée :)



Un nouveau menu apparaît

Voilà maintenant vous pouvez programmer votre carte Arduino à partir de votre Raspberry ou d'un ordinateur.

Testons la carte avec un exemple.

Connecter la carte ARDUINO à l'ordinateur à l'aide du câble USB.

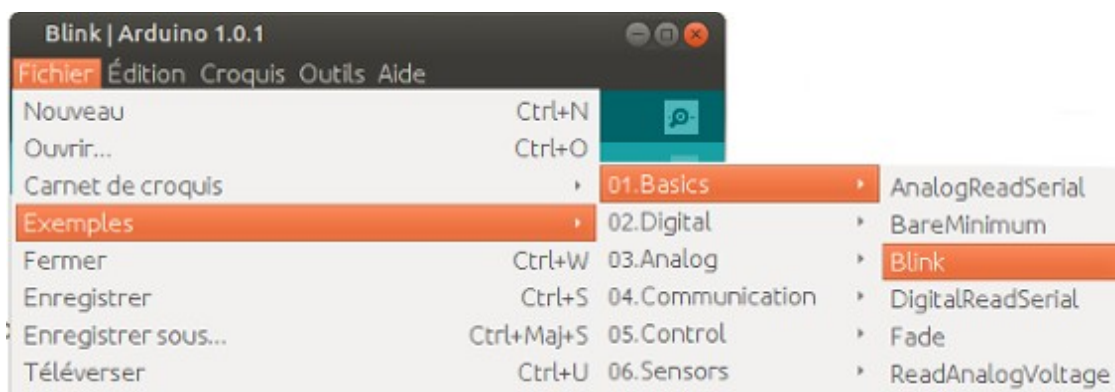


Une LED doit s'allumer sur la carte, intitulé «ON».

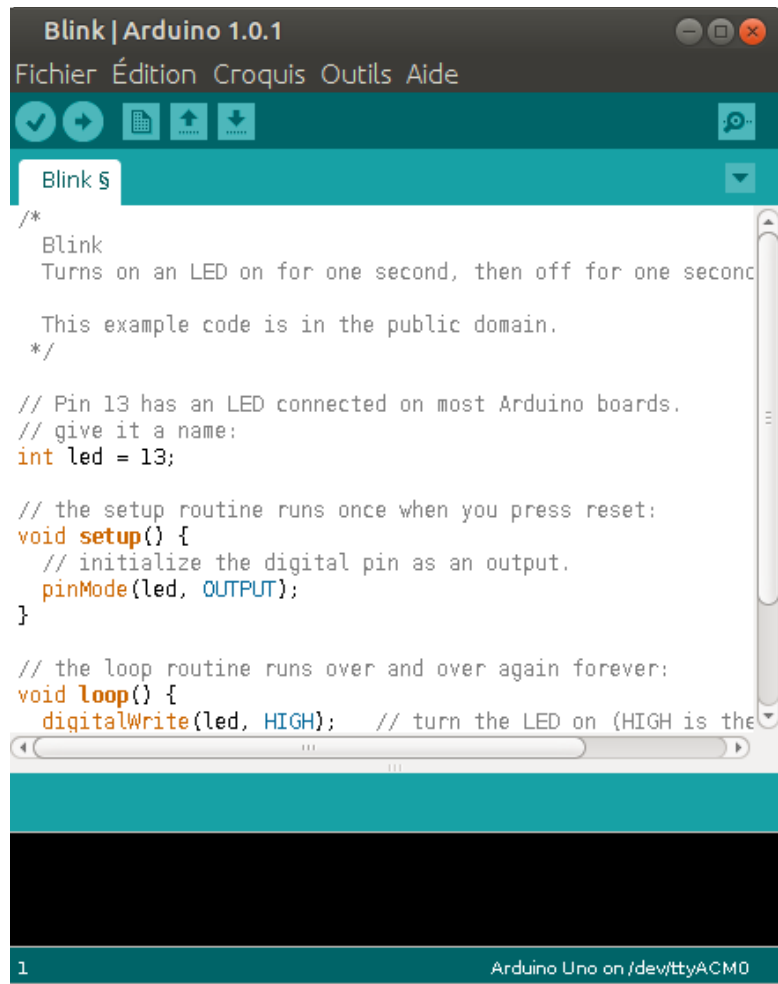
Lancer l'IDE Arduino.

Ouvrir l'exemple Blink.

Ouvrir l'exemple LED Blink : Fichier → Exemples → 1. Basics → Blink.



Voilà le programme Blink est chargé.



The screenshot shows the Arduino IDE interface with the 'Blink' program loaded. The title bar reads 'Blink | Arduino 1.0.1'. The menu bar includes 'Fichier', 'Édition', 'Croquis', 'Outils', and 'Aide'. The toolbar contains icons for saving, running, uploading, and downloading. The main editor area displays the following code:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second
 *
 * This example code is in the public domain.
 */

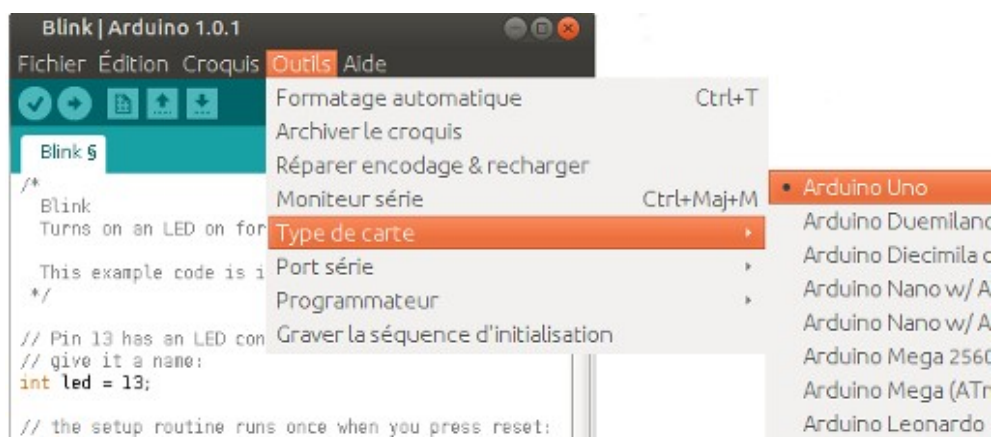
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the
```

Choisir la carte

Choisir la carte Arduino Uno dans le menu «outils».



The screenshot shows the Arduino IDE interface with the 'Outils' menu open. The menu items are:

- Formatage automatique (Ctrl+T)
- Archiver le croquis
- Réparer encodage & recharger
- Moniteur série (Ctrl+Maj+M)
- Type de carte (highlighted)
- Port série
- Programmeur
- Graver la séquence d'initialisation

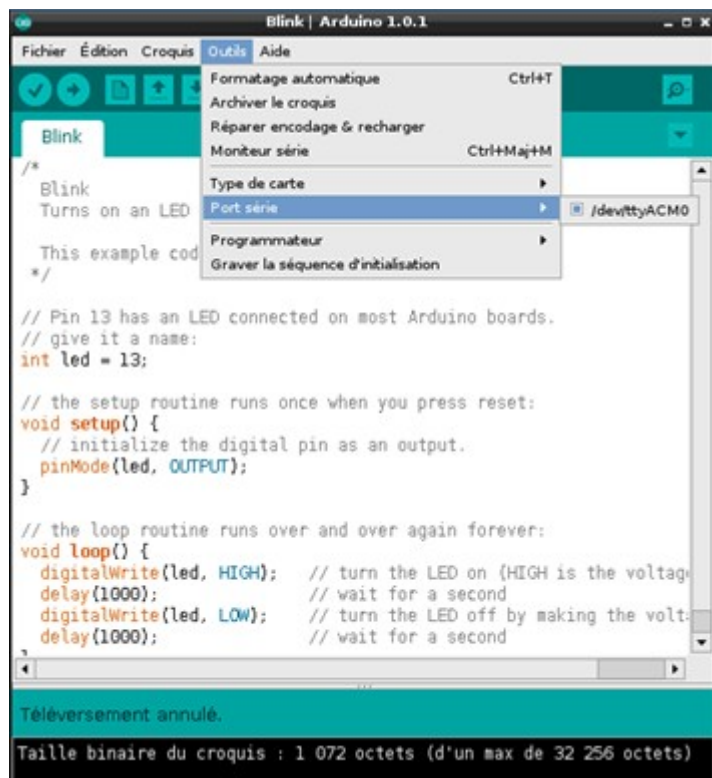
The 'Type de carte' submenu is open, showing the following options:

- Arduino Uno (highlighted)
- Arduino Duemilano
- Arduino Diecimila
- Arduino Nano w/ ATmega168
- Arduino Nano w/ ATmega328P
- Arduino Mega 2560
- Arduino Mega (ATmega1280)
- Arduino Leonardo

Choisir le port

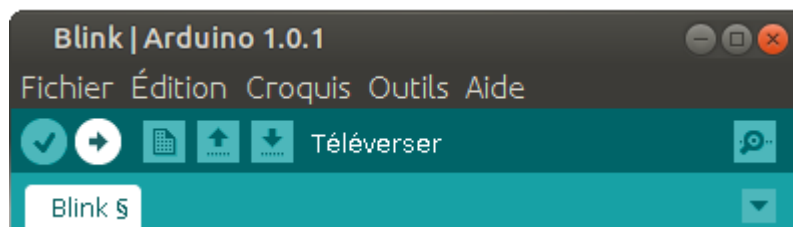
Choisir le port dans le menu «outils». Cela sera sûrement /dev/ttyACM0.

Pour savoir, tu peux déconnecter la carte Arduino et rouvrir le menu ; option grisée si la carte est débranchée.



Téléverser le programme

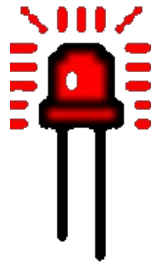
Maintenant, cliquer simplement sur le bouton «Téléverser».



Attendre quelques secondes, les leds TX et RX vont clignoter sur la carte. Si le téléversement a été réussi, le message « téléversement terminé » s'affichera sur la barre.

```
Téléversement terminé
Taille binaire du croquis : 1 084 octets (d'un max de
32 256 octets)
1 Arduino Uno on /dev/ttyACM0
```

Quelques secondes plus tard, la pin 13 (L) LED sur la carte commencera à clignote.



Félicitations !!! Arduino est prêt !!!

Description de la barre des boutons



	Vérifier/compiler : Vérifie le code à la recherche d'erreur.
	Transférer vers la carte : Compile votre code et le transfère vers la carte Arduino. Voir ci-dessous "Transférer les programmes" pour les détails.
	Nouveau : Crée un nouveau code (ouvre une fenêtre d'édition vide)
	Ouvrir : Ouvre la liste de tous les programmes dans votre "livre de programmes". Cliquer sur l'un des programmes l'ouvre dans la fenêtre courante. Note: en raison d'un bug dans Java, ce menu ne défile pas. Si vous avez besoin d'ouvrir un programme loin dans la list, utiliser plutôt le menu File > Sketchbook .
	Sauver : Enregistre votre programme.
	Moniteur Série : Ouvre la fenêtre du moniteur (ou terminal) série.

3) Installer l'environnement Arduino (version récente)

Vérifier la présence de Java

Taper la commande suivante :

```
arnaud@portable ~ $ java -version
java version "1.8.0_74"
Java(TM) SE Runtime Environment (build 1.8.0_74-b02)
Java HotSpot(TM) 64-Bit Server VM (build 25.74-b02, mixed mode)
```

Tout est parfait. S'il n'est pas présent, voici les commandes pour installer Java 8. La commande `sudo` permet de passer en mode super utilisateur. Un mot de passe vous sera demandé. Il est l'équivalent du compte administrateur sous Windows.

```
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update
$ sudo apt-get install oracle-java8-installer
```

Installation des dépendances

```
sudo apt-get install gcc-avr avr-libc
```

Installer l'IDE Arduino

Aller sur la [page de téléchargement d'Arduino](#) et récupérer la dernière version qui correspond à votre architecture, 32 bits ou 64 bits.

Pour connaître la version de votre système, dans votre terminal taper la commande suivante :

```
$ uname -m
```

Retourne `x86_64` si le système est en 64 bits ou `i686`, `i386` pour une version 32 bits.

Nous allons installer une version 64 bits.

Le nouveau processeur du Raspberry Pi 3 fonctionne en 64 bits.



ARDUINO 1.6.8

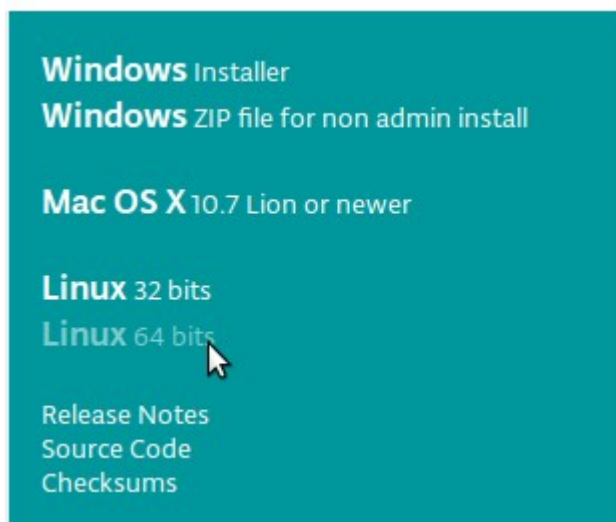
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

Windows Installer
Windows ZIP file for non admin install

Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits

Release Notes
Source Code
Checksums



Windows Installer
Windows ZIP file for non admin install

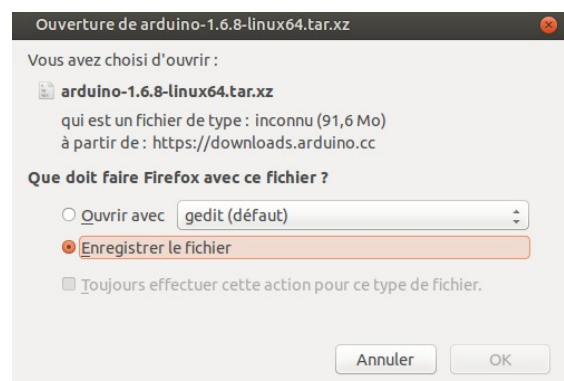
Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits

Release Notes
Source Code
Checksums

A droite on clique sur Linux 64 bits puis sur **JUST DOWNLOAD**.

Le téléchargement peut commencer. Par défaut le fichier est sauvegardé dans le dossier Téléchargement.



Ouverture de arduino-1.6.8-linux64.tar.xz

Vous avez choisi d'ouvrir :

arduino-1.6.8-linux64.tar.xz
qui est un fichier de type : inconnu (91,6 Mo)
à partir de : https://downloads.arduino.cc

Que doit faire Firefox avec ce fichier ?

Ouvrir avec

Enregistrer le fichier

Toujours effectuer cette action pour ce type de fichier.

Annuler OK

Lors de la rédaction de ce tutoriel, la dernière version était la 1.6.8. Le fichier téléchargé se nomme **arduino-1.6.8-linux64.tar.xz**.

Installation dans le dossier \$HOME/arduino

\$HOME est une variable d'environnement qui correspond à /home/utilisateur.

```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
arnaud@portable ~ $ $HOME
bash: /home/arnaud : est un dossier
arnaud@portable ~ $
```

En résumé, dans cet exemple l'environnement de développement Arduino (IDE) sera installé dans le dossier **/home/arnaud/arduino/arduino-1.6.8**.

Nous allons utiliser le terminal pour rentrer des commandes. Voir la première partie (*Le terminal pour saisir des commandes*).

Après le \$, entrer les commandes suivantes :

création du dossier arduino..

```
mkdir ~/arduino
```

On se déplace dans le dossier Téléchargement.

```
cd ~/Téléchargements
```

On décompresse le fichier dans le dossier arduino

```
tar -C ~/arduino -xvzf arduino-1.6.8-linux64.tar.xz
```

On va dans le dossier `/home/arduino/nom_utilisateur/arduino-1.6.8/`.

```
cd ~/arduino/arduino-1.6.8/
```

Ajouter l'utilisateur au group dialout

Il faut ajouter l'utilisateur au group dialout. En supposant que l'utilisateur s'appelle paul.

Pour connaître le nom de l'utilisateur, dans le terminal taper : `whoami`

Puis exécuter les deux commandes :

```
sudo usermod -a -G dialout paul
```

```
sudo usermod -a -G tty paul
```

Connecter la carte et lancer l'IDE

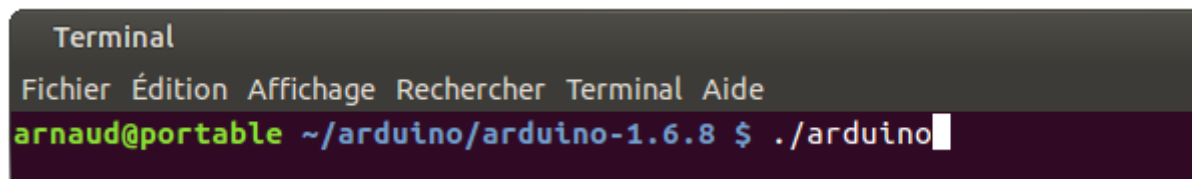
Connecter la carte Arduino à la raspberry ou à l'ordinateur à l'aide du câble USB.

Maintenant ouvrons l'IDE, depuis le dossier contenant l'extraction de l'archive, lancer la commande **arduino** :

```
cd ~/arduino/arduino-1.6.8/
```

Puis on lance l'IDE Arduino avec cette commande.

```
./arduino
```



```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
arnaud@portable ~/arduino/arduino-1.6.8 $ ./arduino
```

Cela ouvre une fenêtre comme celle de la capture suivante :



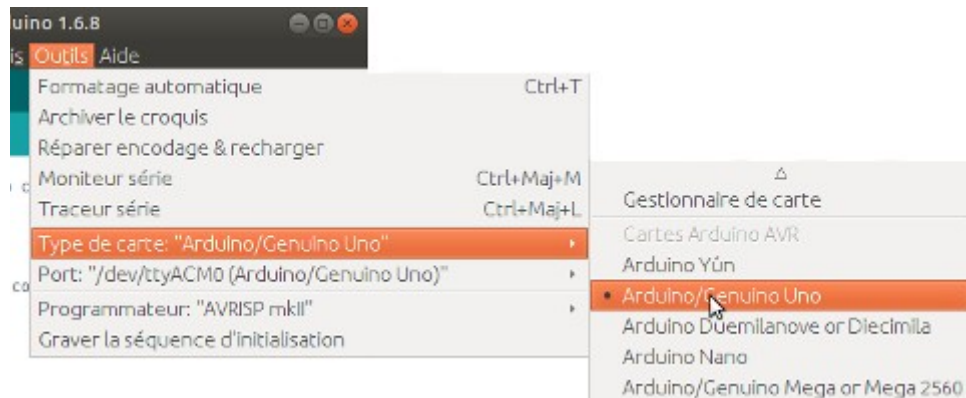
```
sketch_may05a | Arduino 1.6.8
Fichier Édition Croquis Outils Aide

sketch_may05a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }

Arduino/Genuino Uno sur /dev/ttyACM0
```

Pour modifier le port, dans le menu “Outils > Port” choisir celui qui vous intéresse. Dans mon cas, ce sera /dev/ttyACM0.

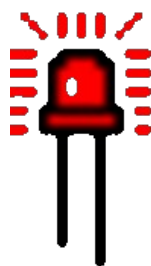
Dans le menu "Outils > "Type de carte" sélectionner Arduino/Genuino Uno



Vous pouvez maintenant charger des sketches (programmes) sur votre carte Arduino... Vous pouvez commencer par l'exemple **blink** fourni avec l'IDE, il fait clignoter la LED intégrée à la carte.

Pour téléverser le programme Blink, se référer à la partie **Installer l'environnement Arduino (ancienne version)**.

Et voilà, votre carte Arduino peut être utilisée sur un système Gnu/Linux (Ubuntu, Debian, Raspbian...)



Félicitations !!! Arduino est prêt !!!