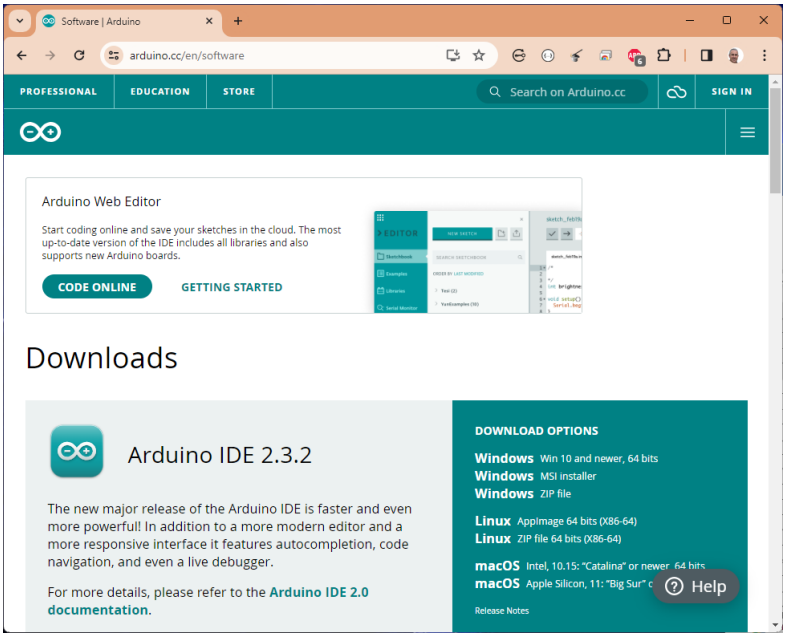

# TP 1 : initiation à Arduino

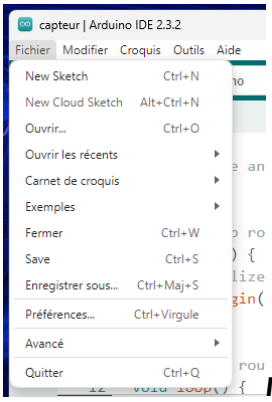
## 0. déroulement des travaux pratiques

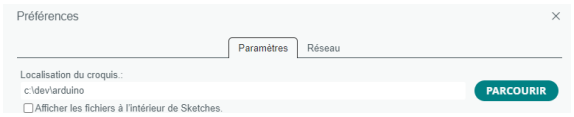
1. Avant de commencer, il faut installer sur le poste de travail l'environnement de travail (IDE) arduino (**http://www.arduino.cc). Supposons que l'IDE soit installé sur le répertoire C:/langages)**
   *Dans le cas contraire, il faut :*
   - *Télécharger le logiciel (version 2.3.2)*
   - *Créer deux répertoires sur **C:/ langages** et sur **C:/dev***
   - *Décompresser l'archive téléchargée dans **C:/langages/arduino-2.x.x** (où x.x représente le numéro de version)*




**Téléchargement de l'IDE Arduino (v. 2.3.2)**

1. Brancher sur un port usb une plaque arduino et installer le pilote (driver) si nécessaire
2. Lancer l'IDE arduino en cliquant sur **arduino.exe** dans son répertoire
3. Dans le répertoire **dev, créer un répertoire arduino**
4. Modifier dans **Fichier | Préférences** l'emplacement du carnet de croquis vers **C:/dev/arduino**. Appuyer sur ok et redémarrer **arduino.exe**




**Modification des préférences (emplacement des réalisations)**

5. **Nous pouvons commencer à travailler** ☺ Nous installerons plus tard de nouvelles librairies et de nouveaux langages (Processing.org par exemple) pour contrôler arduino et le logiciel (Fritzing) pour se rappeler de nos montages arduino facilement.

# 1. introduction

### 1.1 Histoire d'arduino (*http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino/0*)

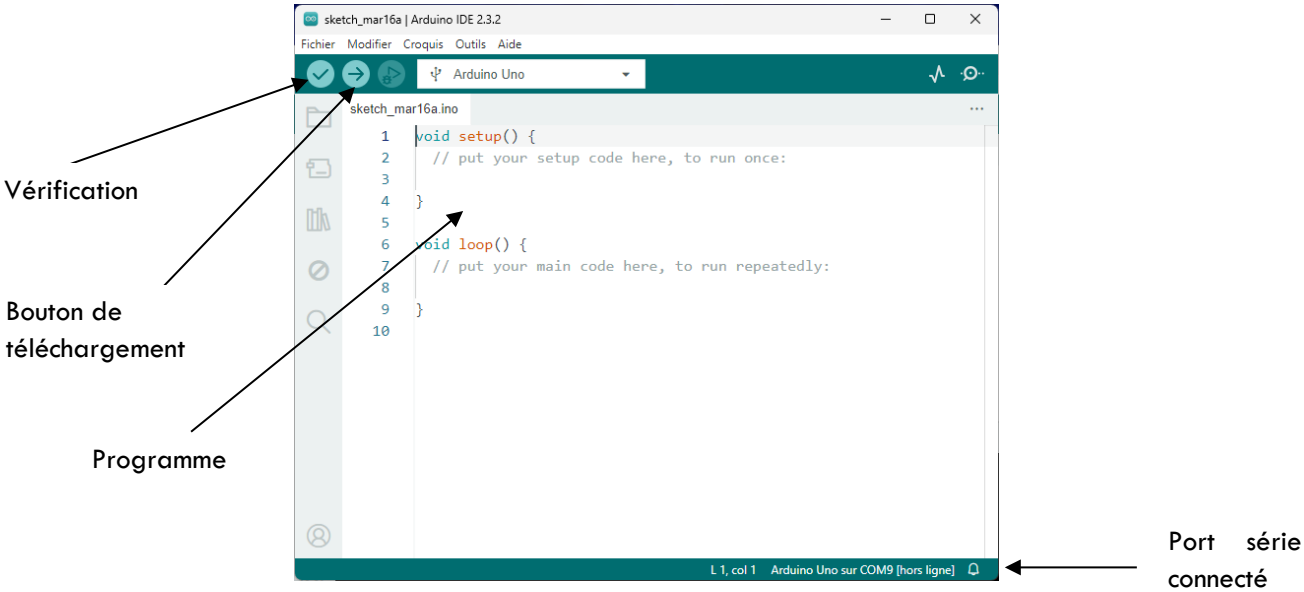
C'est à Ivrea en Italie, terres du roi Arduin (Arduino en italien) vers l'an mil que commence l'histoire de cette plateforme électronique. Créé en 2005 comme outil pour les étudiants de l'Interaction Design Institute d'Ivrea, Arduino est devenu en moins de 8 ans le projet de plus influent de l'électronique moderne.

Sous licence Creative Commons (les plans sont libres), arduino peut permettre d'effectuer des tâches extrêmement diversifiées comme des tâches domotique ou robotique.

Il existe de nombreux matériels compatibles Arduino comme ESP32 (https://www.espressif.com/en/products/socs/esp32) , teensy (https://www.pjrc.com/teensy), …

### 1.2 IDE Arduino

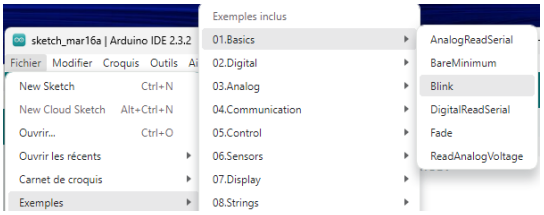
L'IDE (Environnement de Développement) permet de préparer ses programmes (appelés « *sketch* »), vérifier la syntaxe et télécharger le programme sur la plateforme arduino.




**Vue de l'IDE Arduino**

Il y a plusieurs manières de programmer pour arduino. La première est d'utiliser le langage (la syntaxe est proche du langage C).

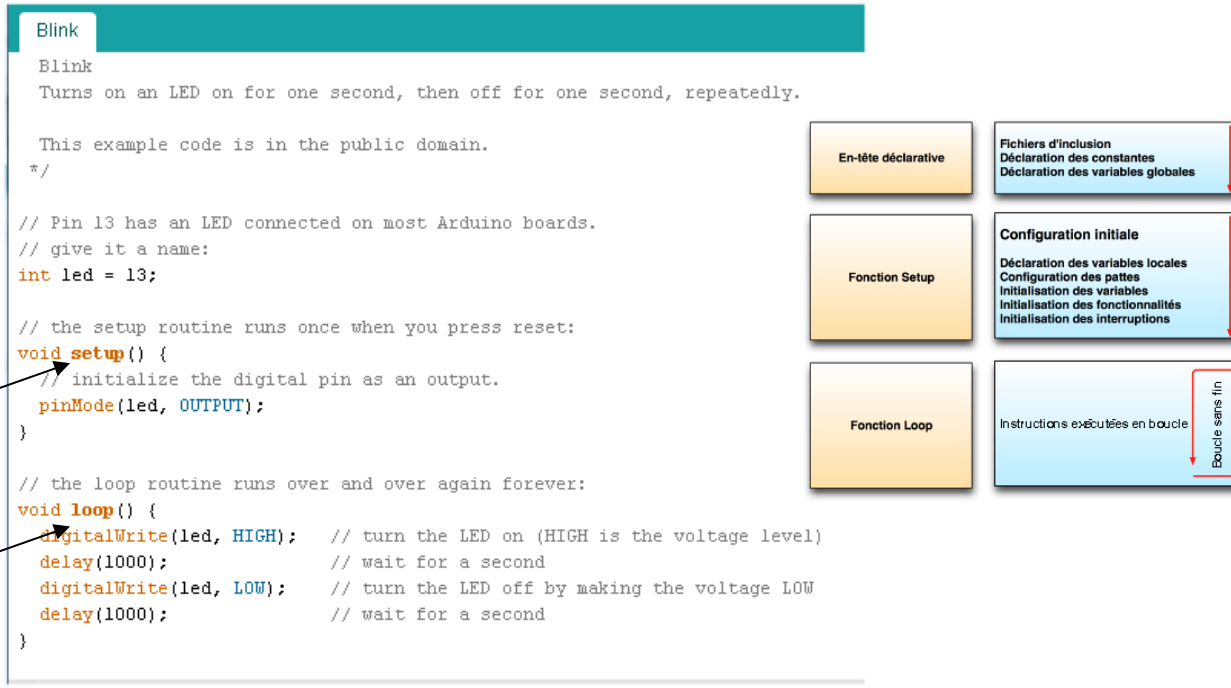
Prenons un exemple. Ouvrons le programme « **blink** » situé dans « **Fichier | Exemples | 01.Basics | Blink** ». Le programme va faire clignoter une led que nous allons placer sur le pin 13.



**Ouverture de l'exemple « blink »**

Le programme arduino est divisé en deux parties (fonctions) par défaut :

- **setup()** qui initialise un certain nombre de valeurs
- **loop()** qui correspond à une boucle sans fin et qui exécute le code contenu dans la fonction.



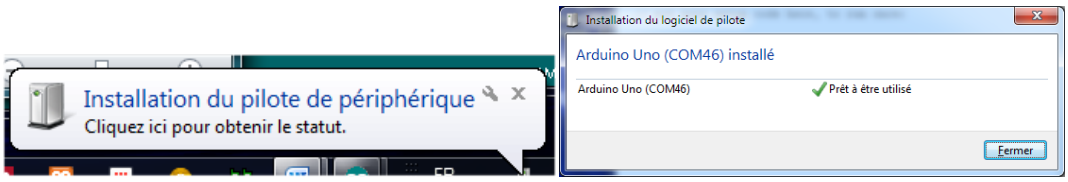

initialisation

Code du programme : allumer et éteindre la led

**Exemple de clignotement de led**

**Mettre une led sur la plaque arduino : grande patte (+) sur le pin 13[1], petite sur le pin GND (-)**

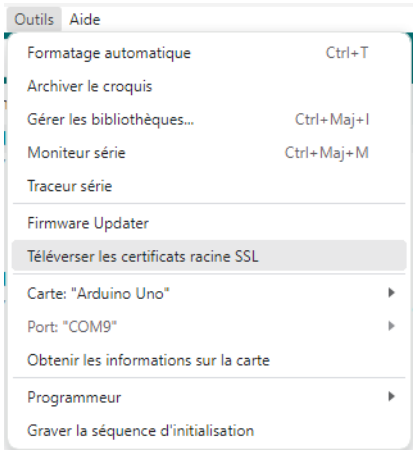
Branchez la plaque arduino : à la première utilisation, le pilote de périphérique s'installe. Attendez qu'un port série soit affecté à notre plaque




**Installation du pilote de périphérique**

Toujours lors de la première utilisation, il faut configurer la version de votre plaque et le numéro de port série. Pour ce faire, aller dans le menu « **Outils** » puis successivement « **Carte** » pour configurer le type de votre plaque arduino et « **Port Série** » pour le numéro de port (choisissez le numéro de port préalablement donné lors de l'installation du pilote)

[1] *Le pin 13 possède une résistance interne de 220 ohms*




**Configuration de la plaque et du port série**

Vérifiez le programme en appuyant sur le bouton  puis téléchargez votre code en appuyant sur le bouton  . **C'est prêt !** La led devrait clignoter …

Ce rapide aperçu laisse entrevoir de très nombreuses possibilités simples à mettre en œuvre : contrôle de capteurs, d'effecteurs (led, buzzer, moteurs, …), contrôle à distance sans fil, etc., etc.

# 2. A vous de jouer !!!

## 2.1 une led comme capteur de lumière

Nous allons utiliser une led infrarouge pour allumer un éteindre une led branchée sur le pin 13.

Branchez la led infrarouge sur l'entrée analogique **A0** et **GND**.

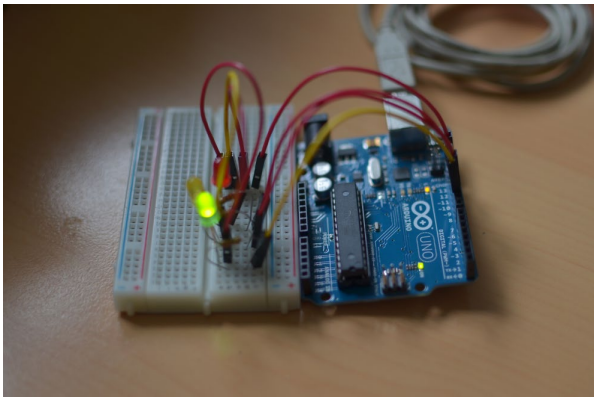
Ecrire un programme qui lit une valeur sur l'entrée **A0** et allume la led sur le pin « 13 » si une valeur de seuil est dépassée.

## 2.2 un capteur de température

Utilisons maintenant un module de température KY-013 (**https://sensorkit.joy-it.net/en/sensors/ky-013**)

Câblez votre capteur et codez la fonction de transformation qui permettra de renvoyer la valeur de température en degrés Celsius sur la liaison série en fonction des valeurs lues sur le pin analogique.

## 2.3 feux tricolores



Un feu tricolore peut être dans l'état rouge, orange, vert. Dans le cas classique, le feu est successivement vert, puis orange, puis rouge, puis de nouveau vert, etc.

Dans ce cas, il reste 6 secondes au vert, 1 seconde à l'orange, et 7 secondes au rouge (histoire de ne pas attendre trop longtemps ☺)

Déterminez le montage électronique (**Attention** : pensez à utiliser une résistance de 220 ohms si vous utilisez un autre pin que le pin 13), codez et testez !

**Nota** : vous pouvez utiliser le logiciel Fritzing (**http://fritzing.org/download**) pour dessiner et sauver votre montage !

### 2.4 changement d'état via un bouton physique (feux … suite)

Modifiez votre montage et votre code de telle manière que l'appui sur un bouton physique permette de changer de l'état nominal à l'état clignotant (le feu orange clignote à une fréquence de clignotement toutes les deux secondes) et inversement.

### 2.5 contrôle de moteurs (feux … encore)

Modifiez encore votre montage afin d'intégrer une barrière (ouverte/fermée) à partir d'un servomoteur sur une des intersections.

### 2.6 contrôle par le PC (feux … toujours)

Modifiez encore le résultat de telle sorte qu'un envoi de données par la liaison série (un caractère par exemple) à l'arduino permette ce changement d'état (contrôle par le PC).

## 3. Allez plus loin … en codant une interface sur le PC

Nous pouvons aller plus loin en contrôlant la carte depuis des interfaces sur le PC

Pour ce faire, vous aurez sans doute à télécharger le logiciel suivant :

- **https://arduino.cc/en/Reference/HomePage** : la page de référence du langage

- **Processing.org** : **https://processing.org/download**
  Processing est le langage qui a inspiré Arduino. Même IDE, syntaxe proche (même si Processing est issu du langage Java), Arduino et Processing forme une combinaison « magique » !

Développez un système de gestion de feux tricolores (une intersection par exemple) avec une interface graphique qui commande les feux sur Arduino.