

La programmation

Le logiciel "ARDUINO IDE"

L'Arduino est une carte électronique qui ne sait rien faire sans qu'on lui dise quoi faire. Pourquoi ? Eh bien c'est dû au fait qu'elle est programmable. Cela signifie qu'elle a besoin d'un programme pour fonctionner.

Voici un exemple de programme :

Ce programme a été édité avec le logiciel "ARDUINO IDE " qui est disponible à l'adresse suivante :

<https://www.arduino.cc/en/Main/Software>

Le logiciel "ARDUINO IDE" est un logiciel de programmation pour l'Arduino, dans un langage dérivé du C/C++.

IDE veut dire : Integrated Development Environment ou en français, environnement de développement intégré.

Bien-sûr, pour programmer une carte Arduino, celle-ci doit être au préalable connectée à un ordinateur à l'aide d'un câble USB type A-B.

```
1 // définition de la broche 2 de la carte en tant que variable
2 const int led_rouge = 2;
3
4 // fonction d'initialisation de la carte
5 void setup()
6 {
7     // initialisation de la broche 2 comme étant une sortie
8     pinMode(led_rouge, OUTPUT);
9 }
10
11 void loop()
12 {
13     // allume la LED
14     digitalWrite(led_rouge, LOW);
15     // fait une pause de 1 seconde
16     delay(1000);
17     // éteint la LED
18     digitalWrite(led_rouge, HIGH);
19     // fait une pause de 1 seconde
20     delay(1000);
21 }
```

L'Arduino IDE est une interface graphique simple d'utilisation qui regroupe tous les outils qui permettent de programmer l'Arduino

. Installation et exécution du logiciel "ARDUINO IDE"

Une fois téléchargé, double-cliquez sur l'exécutable du fichier et suivez tout simplement le guide d'installation.

Quand le logiciel est installé, ouvrez-le, une fenêtre devrait s'afficher, confirmant ainsi que l'environnement de développement a été correctement installé.

Tout en haut de la fenêtre, vous avez la barre de menus qui regroupe toutes les actions que vous pouvez réaliser avec le logiciel.

Cependant, ce qui nous intéresse, c'est la zone des 5 icônes juste en dessous, ce sont les boutons que vous utiliserez le plus souvent lorsque vous programmerez. Nous avons, de la gauche vers la droite :



- Le bouton "**Vérifier**", pour vérifier votre programme : il faut en effet que le programme que vous avez écrit ne présente pas de bugs afin de s'exécuter correctement,



- Le bouton "**Téléverser**" : en cliquant sur ce bouton, vous transférez votre programme compilé dans la mémoire de votre carte Arduino,



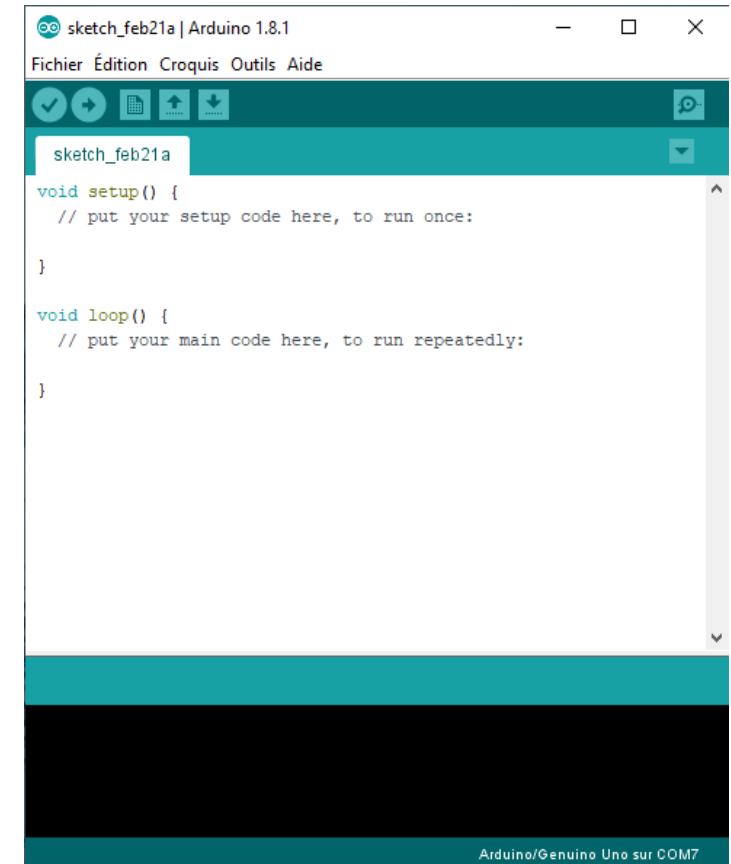
- le bouton "**Nouveau**" : c'est à l'aide de ce bouton que vous créez de nouveaux programmes,



- le bouton "**Ouvrir**" qui vous permet d'accéder aux programmes d'exemple de l'IDE ou aux programmes présents sur votre ordinateur,



- le bouton "**Enregistrer**" par lequel vous pouvez sauvegarder le travail que vous avez réalisé afin d'y revenir quand vous le souhaitez.

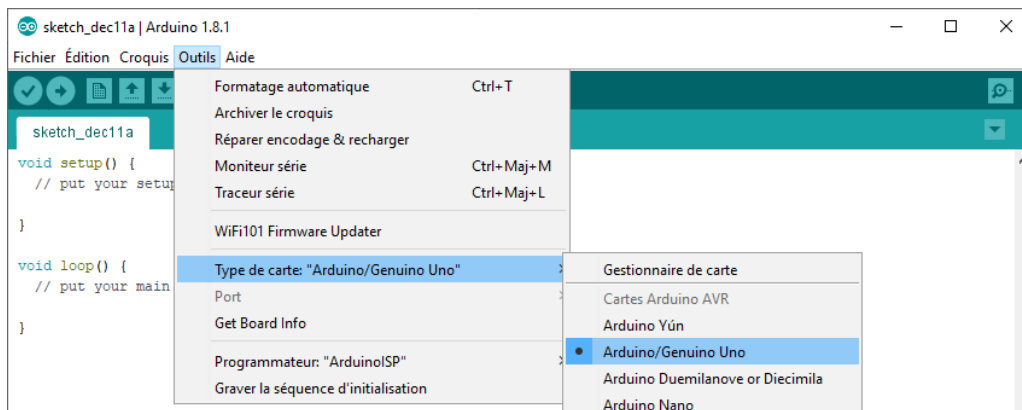


Tout en bas de la fenêtre du logiciel, se trouve la console de débogage : c'est là que vous trouverez les différentes erreurs d'exécution de votre programme le cas échéant. Grâce à cette console, vous vous assurez que votre programme fonctionne comme vous le souhaitez avant de le transférer sur votre carte Arduino.

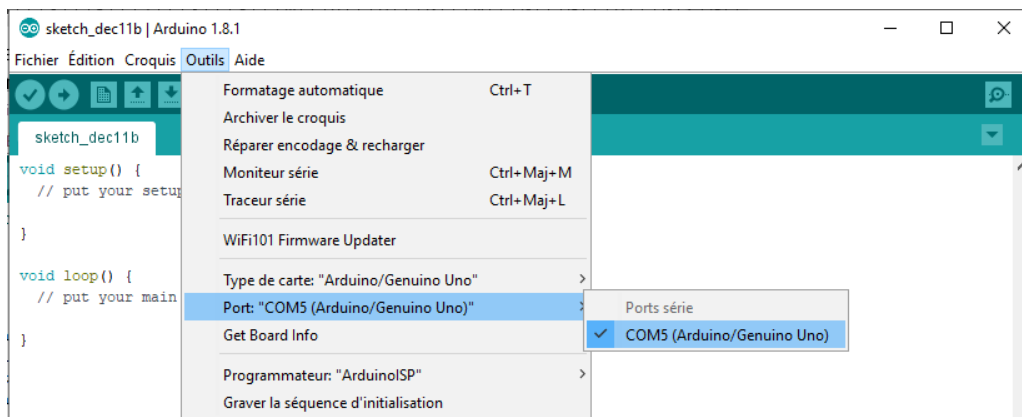
Attention :

Avant de commencer à programmer, il faut d'abord indiquer au logiciel le type de carte qui recevra le programme.

Pour cela, cliquez sur le menu "Outils", puis "Type de carte" et sélectionnez le type correspondant à la carte que vous avez branchée sur votre ordinateur :



Une fois le type sélectionné, vous devez aussi indiquer le port sur lequel votre carte est branchée. Vous le trouverez dans le Menu "Outils" puis "Port" :



Vous pouvez maintenant commencer à programmer et envoyer votre programme sur l'Arduino en cliquant sur le bouton "Téléverser".

Après l'étape de compilation, le programme est transféré sur la carte. Vous aurez, au niveau de la console de débogage, un message vous indiquant que le téléversement est terminé.

Une fois le programme téléversé, il est conservé dans la mémoire de l'Arduino. Et dès que votre carte Arduino est alimentée, celle-ci exécute le programme qui est stocké dans sa mémoire.

Donc, lorsque vous branchez l'Arduino à l'ordinateur, le dernier programme reçu est exécuté. Et avant de commencer un nouveau montage, il est plus que conseillé de téléverser un programme d'initialisation, du modèle de celui-ci :

//Programme vide d'initialisation

```
void setup() {
```

```
}
```

```
void loop() {
```

```
}
```

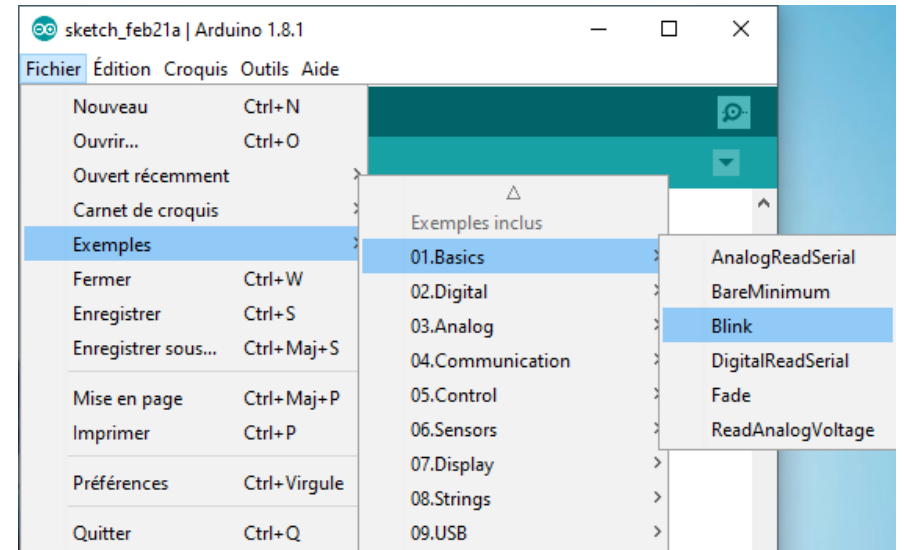
Initiation au langage "ARDUINO"

Dans le logiciel "ARDUINO IDE", de nombreux exemples pré-enregistrés peuvent être chargés et testés. Ils permettent de se familiariser plus rapidement avec le langage "ARDUINO".

Pour commencer, nous allons étudier le programme le plus simple. Il s'agit du programme "Blink" qui fait clignoter la Led L intégrée à la carte Arduino et connectée à la broche 13.

Pour ouvrir le programme : Faire **Fichier > Exemples > 01.Basics > Blink**.

Le programme est alors affiché :



LED_BUILTIN correspond à la led qui se trouve sur la carte
La broche de la Led est initialisée en sortie numérique

```
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(1000);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000);  
}
```

Envoie 5V sur la LED de la carte

Attendre 1000 ms

Remet la tension de la led à 0V

. Structure de base d'un programme

On remarque que ce programme est séparé en 2 parties :

. une fonction **setup()**

Cette fonction est appelée au démarrage du programme. Elle est utilisée pour initialiser des variables préalablement déclarée, configurer le sens des broches. La fonction setup n'est exécutée qu'une seule fois, après chaque mise sous tension ou reset (réinitialisation) de la carte Arduino.

Ici, on définit que la broche 13, reliée à la "LED_BUILTIN", sera utilisée en sortie (OUTPUT).

. une fonction **loop()**

La fonction loop () (boucle en anglais) fait exactement ce que son nom suggère et exécute en boucle sans fin ses instructions pour contrôler la carte Arduino.

Ici elle applique une tension de 5V (HIGH) sur la broche 13, attend 1s (1000 ms), applique ensuite une tension de 0V (LOW) et attend encore 1s.

Syntaxe des fonctions :

```
void setup()  
{  
}
```

```
void loop()  
{  
}
```

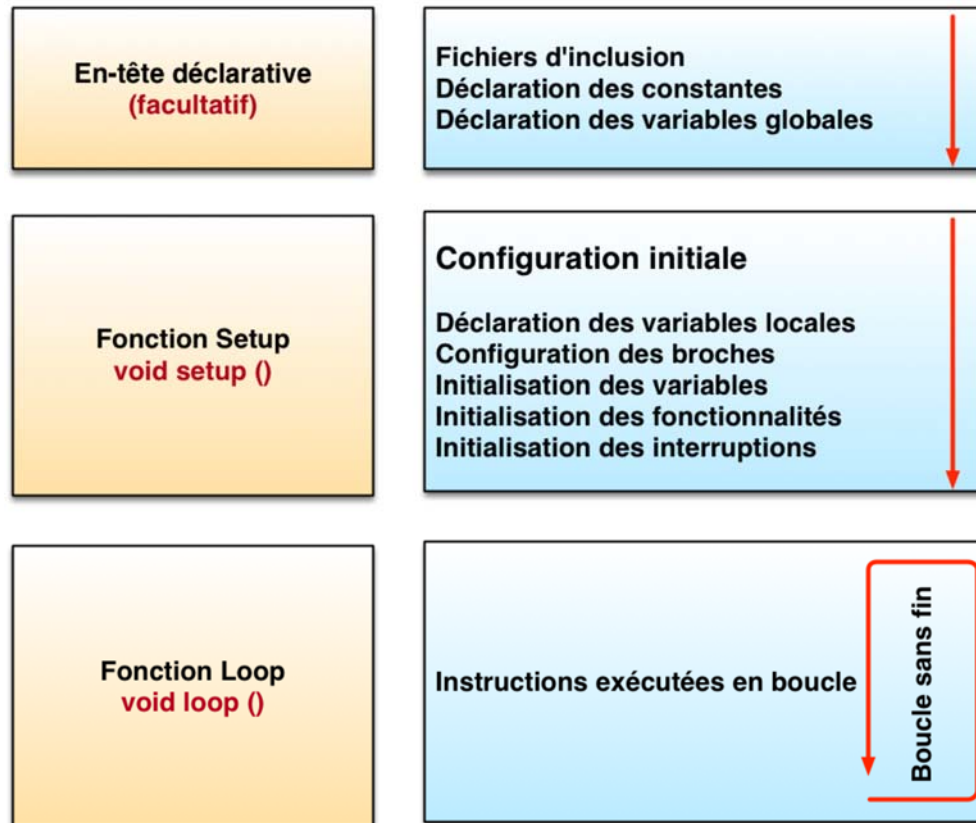
Dans tout programme pour Arduino, codé en langage "Arduino", ces deux fonctions sont obligatoires.

En C/C++, les instructions sont séparées par des points virgules ";" et les blocs de code sont encadrés par des accolades { }.

Une variable doit toujours être déclarée avec son type.

Généralement, en plus des deux fonctions obligatoires, le programme comporte également au début, une partie déclarative (Inclusion des bibliothèques, déclaration des constantes et variables globales)

Déroulement du programme



Types de variables

Nombres entiers

- **boolean** : 1 *bit* [0; 1]
- **byte** : 8 *bits* [0; 255]
- **int** : 16 *bits* [-32 768; 32 767]
- **long** : 32 *bits* [-2 × 10⁹; 2 × 10⁹]

Nombres décimaux

- **float** : 32 *bits* 6 chiffres significatifs

Autres

- **char** : 8 *bits* Code ASCII

Les variables, **déclarées en début de programme et en dehors des fonctions** sont dites **globales** parce qu'elles sont accessibles et modifiables depuis n'importe quel endroit du code.