



## **Atelier Arduino Débutants**

### **Sommaire**

- 1- Présentation de la carte Arduino UNO**
- 2- Notions d'électricité et d'électronique**
- 3- Installation et prise en main de l'IDE Arduino**
- 4- Faire clignoter une LED, étude du programme**
- 5- Le microcontrôleur**
- 6- Système décimal, binaire, hexadécimal – Code ASCII**
- 7- Ecrire un programme qui lance un SOS – les fonctions**
- 8- Organisation de la mémoire – les variables**
- 9- Les opérateurs – les conditions et structures**
- 10- L'Organigramme ou l'algorigramme**
- 11- Les sorties PWM**
- 12- Les entrées numériques**
- 13- Les entrées analogiques**
- 14- Etude et réalisation des différents projets**

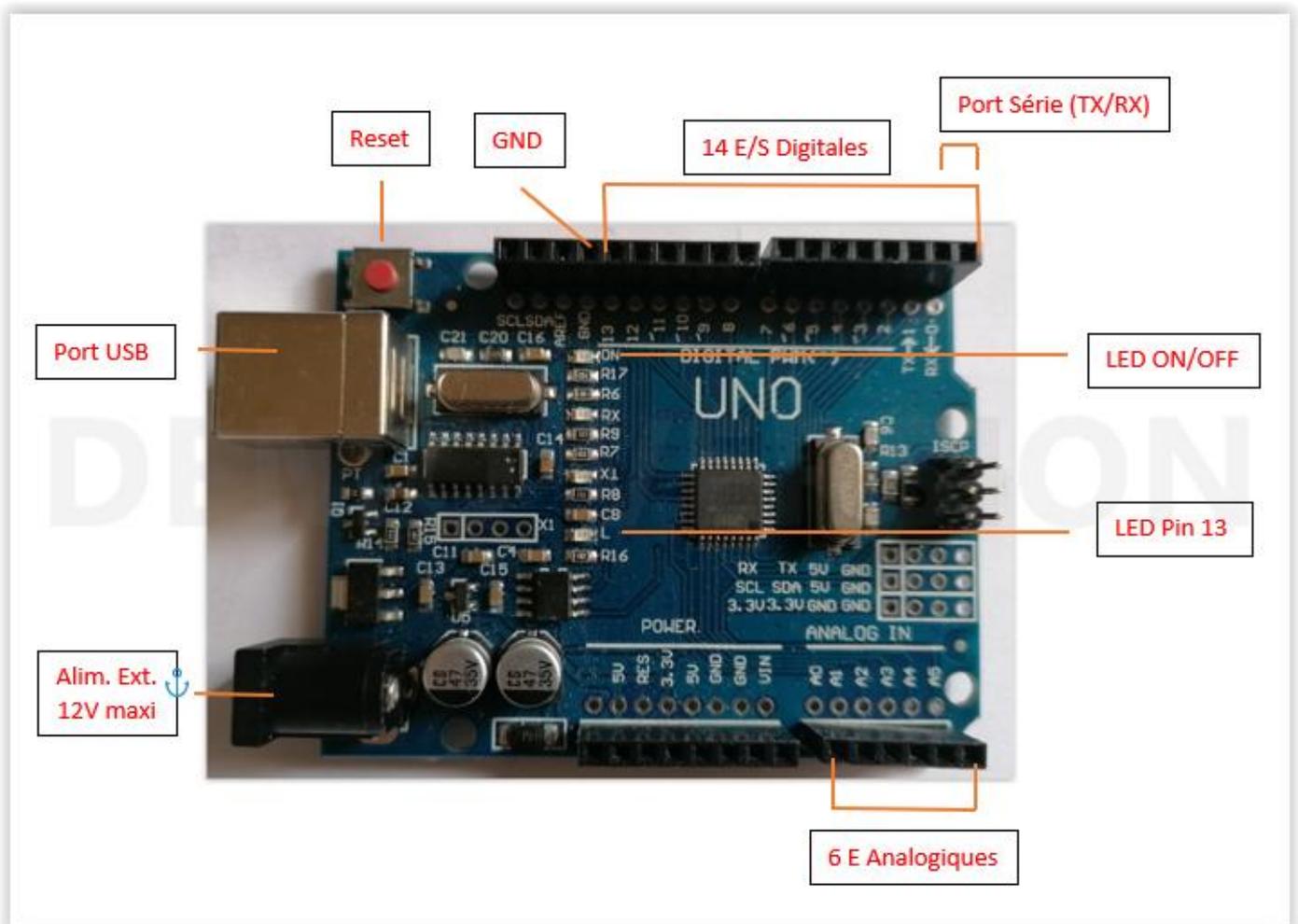
# 1- Présentation de la carte Arduino UNO

Définition :

Une carte Arduino est une plateforme électronique dotée d'un microcontrôleur.

Ce microcontrôleur permet de programmer des actions en réponse à des événements détectés par des capteurs.

En résumé, la carte Arduino sert d'interface programmable pour la création et la gestion de systèmes interactifs.



Nous détaillerons les différentes broches de la carte plus tard au fur et à mesure de notre progression.

## 2- Notions d'électricité et d'électronique

- Tension
- Intensité
- Puissance
- Résistance
- LED
- Potentiomètre
- Transistor

### Tension :

La **tension** est la différence de potentiel électrique entre deux points dans un circuit. Elle agit comme une "force" qui pousse les électrons à travers un conducteur, un peu comme une pression qui fait circuler de l'eau dans un tuyau. La tension est mesurée en volts (V). Plus la tension est élevée, plus elle peut faire circuler de courant dans le circuit.

### Intensité :

**L'intensité** est la quantité de courant électrique qui circule dans un circuit. Elle représente le flux d'électrons qui passent par un point donné en une seconde, un peu comme la quantité d'eau qui coule dans un tuyau. L'intensité se mesure en ampères (A). Plus l'intensité est élevée, plus le courant qui circule dans le circuit est important.

### Puissance :

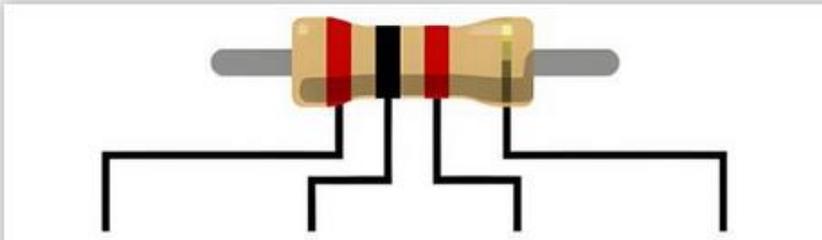
La **puissance** électrique est la quantité d'énergie consommée ou produite par un appareil en un temps donné. Elle indique la vitesse à laquelle un appareil utilise ou génère de l'électricité. La puissance est mesurée en watts (W). Plus la puissance est élevée, plus l'appareil consomme ou produit d'énergie rapidement.

## Résistance :

Une **résistance** est un composant électronique qui limite le passage du courant électrique dans un circuit. Elle est caractérisée par sa valeur de résistance, mesurée en ohms ( $\Omega$ ), et elle joue un rôle crucial dans la régulation du courant, la protection des autres composants et le réglage de la tension.

Pour déterminer la valeur d'une résistance, il y a plusieurs méthodes.

- La plus courante consiste à utiliser les bandes de couleur présentes sur la résistance elle-même. Chaque couleur correspond à un chiffre spécifique, et en les lisant dans l'ordre, tu peux déterminer la valeur de la résistance.



1st digit	2nd digit	Multiplier	Tolerance
0	0	x 1	
1	1	x 10	±1%
2	2	x 100	±2%
3	3	x 1K	
4	4	x 10K	
5	5	x 100K	
6	6	x 1M	
7	7		
8	8	x 0.1	±5%
9	9	x 0.01	±10%

- Tu peux également mesurer la valeur d'une résistance avec un multimètre en mode "ohmmètre" :
  - Branche les deux pointes du multimètre sur les deux bornes de la résistance.
  - La valeur de la résistance s'affiche directement en ohms ( $\Omega$ ).



Un **multimètre** est un **outil polyvalent** qui te permet de mesurer :

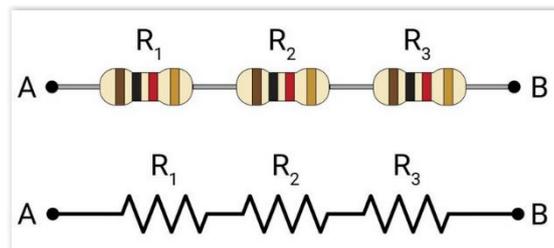
- **La tension** pour savoir s'il y a une différence de potentiel.
- **Le courant** pour mesurer le flux d'électricité dans un circuit.
- **La résistance** pour vérifier si un composant limite bien le passage du courant.
- **La continuité** pour t'assurer qu'un circuit est bien connecté.

C'est un instrument essentiel pour tout projet électronique ou électrique !

## Résistances en série

- **Comment ça marche** : Les résistances sont en ligne, comme des perles sur un fil.
- **Courant** : Le même courant passe à travers chaque résistance.
- **La résistance équivalente (Re)** : La résistance totale est la somme des résistances individuelles.

$$R_e = R_1 + R_2 + R_3 + \dots$$

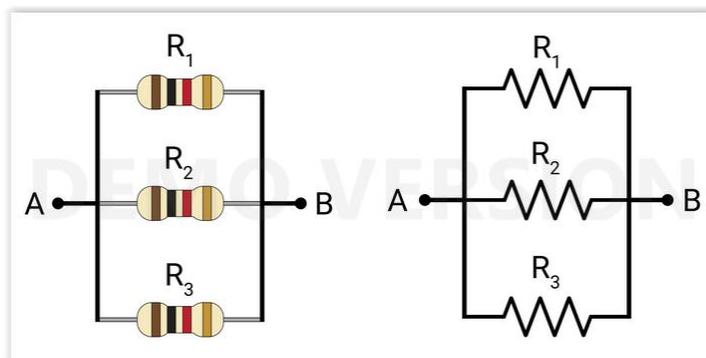


## Résistances en parallèle :

- **Configuration** : Les résistances sont connectées aux mêmes bornes, formant des chemins multiples pour le courant.
- **Résistance équivalente (Re)** : La résistance totale est donnée par la formule :

$$1/R_e = 1/R_1 + 1/R_2 + 1/R_3 + \dots$$

- **Courant** : Le courant se divise entre les chemins, donc chaque résistance peut avoir un courant différent.
- **Tension** : La tension est la même à travers chaque résistance.



La loi d'Ohm est une relation fondamentale en électricité qui relie la tension, le courant et la résistance dans un circuit électrique. Elle s'exprime par la formule suivante :

$$U = R \times I$$

**U** : Tension en volts (V)

**R** : Résistance en ohms ( $\Omega$ )

**I** : Courant en ampères (A)

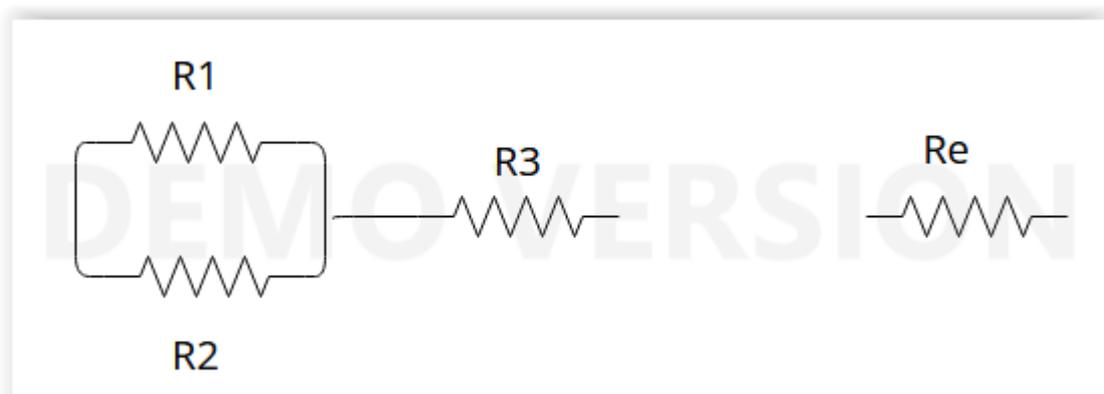
La puissance électrique représente la quantité d'énergie fournie ou consommée par un composant électrique.

$$P = U \times I \quad \text{ou} \quad P = R \times I^2$$

**P** : Puissance en watts (W)

**Exercice** : Calculer la résistance équivalente  $R_e$

$$R_1 = 300\Omega \quad R_2 = 300\Omega \quad R_3 = 1.5K\Omega$$

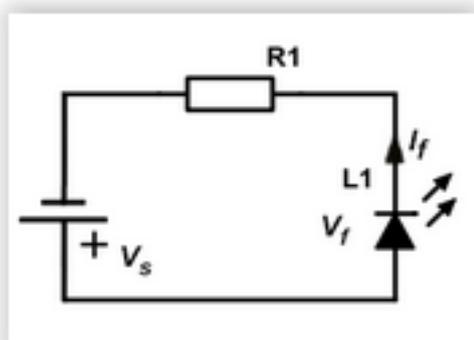


Une **LED** (Light Emitting Diode) est une diode électroluminescente qui émet de la lumière lorsqu'un courant électrique la traverse.



**Exercices :**      **Calculer la résistance R1 et la puissance consommée P1 pour :**

- Luminosité faible : 5 mA
- Luminosité moyenne : 10 mA
- Luminosité forte : 15 mA



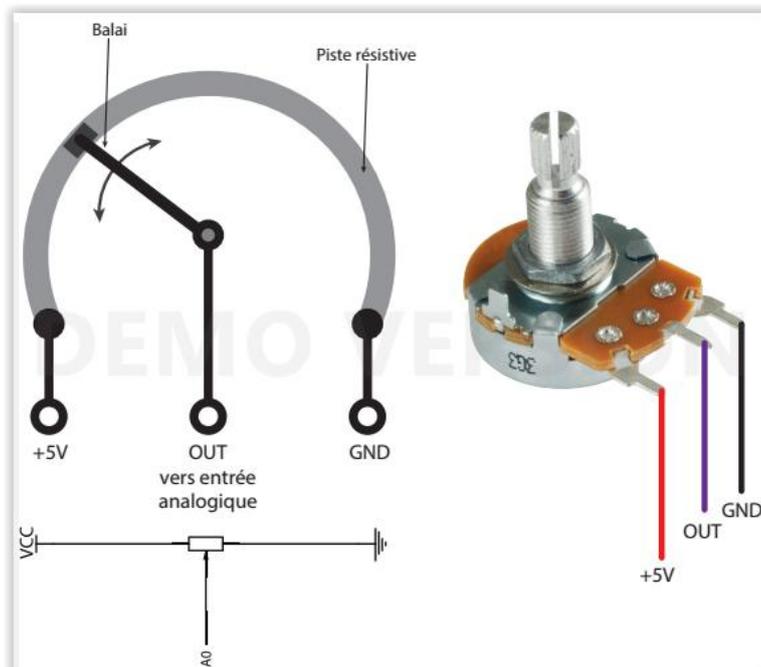
Couleur de LED	Tension admissible (Volt)
Rouge	1,8 à 2,1
Ambre	2 à 2,2
Orange	1,9 à 2,2
Jaune	1,9 à 2,2
Vert	2 à 3,1
Bleu	3 à 3,7
Blanc	3 à 3,4

## Potentiomètre :

Un potentiomètre est un composant électronique qui fonctionne comme une résistance variable.

Il a trois bornes : deux aux extrémités d'une piste résistive et une borne centrale appelée "curseur".

En tournant ou en déplaçant le curseur, on modifie la position sur la piste résistive, ce qui change la valeur de la résistance entre le curseur et les deux autres bornes.



## Transistor :

Un transistor, c'est un petit composant dans un circuit qui a deux utilisations principales :

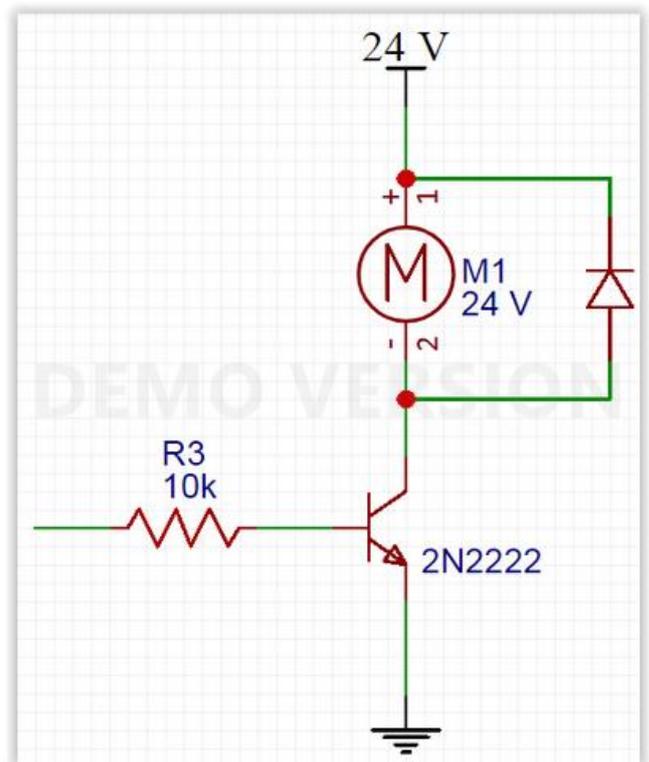
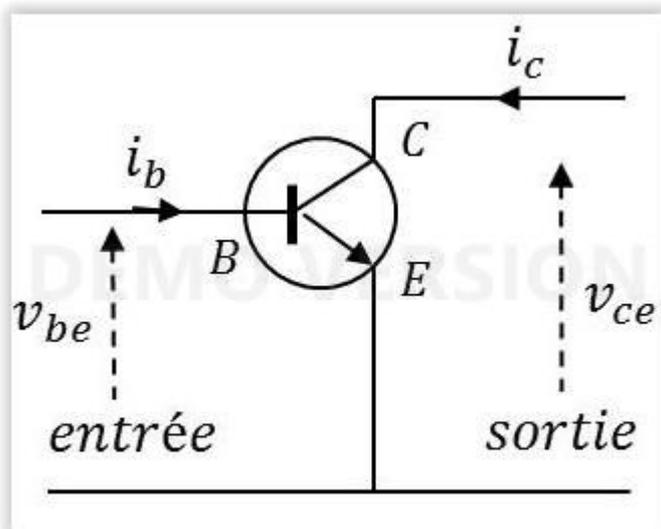
### 1. Agir comme un interrupteur :

Quand le signal est envoyé à la base, c'est comme si on appuyait sur un interrupteur, et le courant peut circuler.

Quand il n'y a pas de signal à la base, c'est comme si l'interrupteur était ouvert, et le courant ne passe pas.

### 2. Amplifier un signal :

Il peut rendre un petit signal (comme un son faible) plus fort.



### 3 Installation et prise en main de l'IDE Arduino

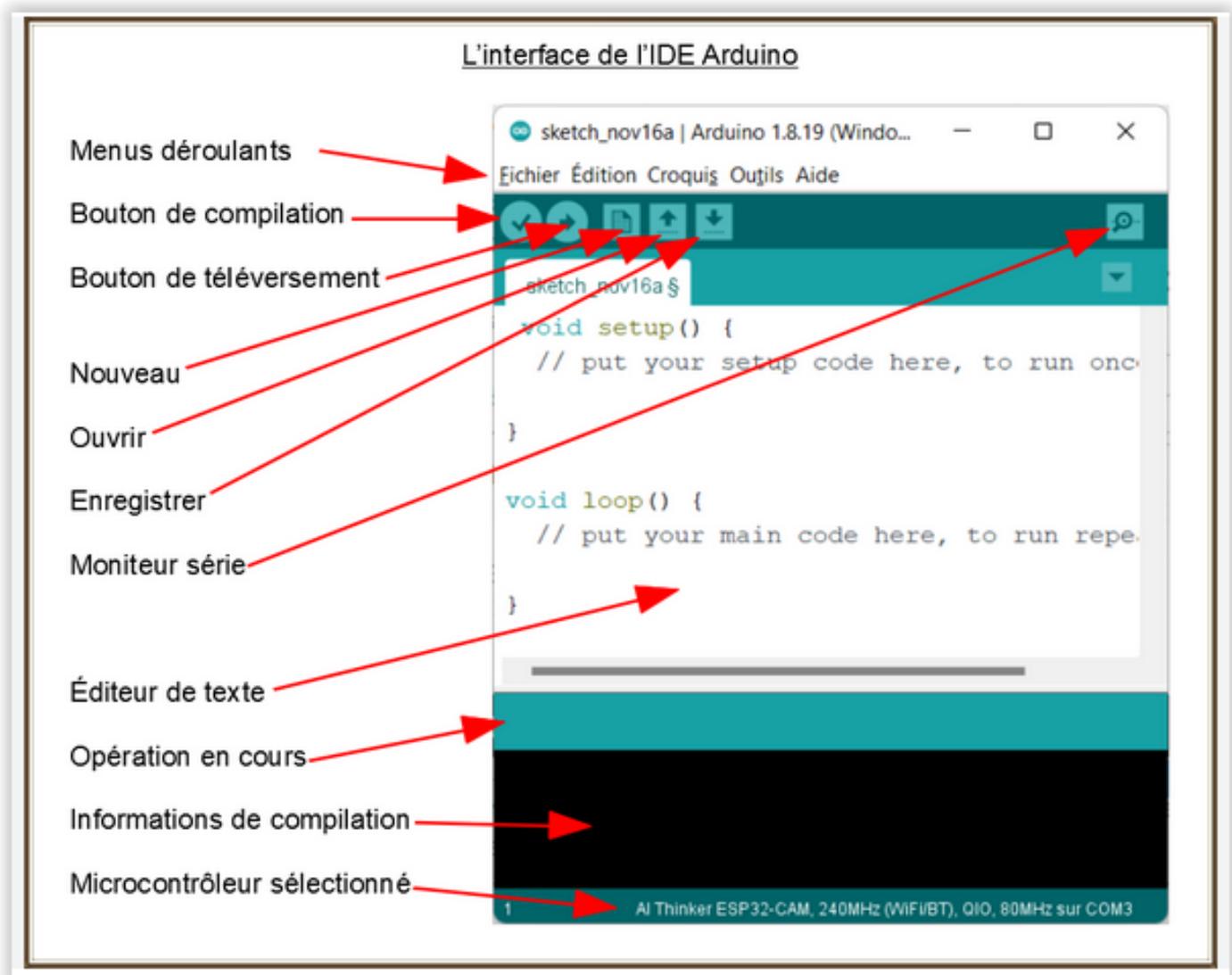
- Découverte de l'IDE Arduino :

L'IDE Arduino (Integrated Development Environment) est le logiciel principal que tu vas utiliser pour programmer les cartes Arduino. C'est un outil simple et puissant, idéal pour débiter avec la programmation de microcontrôleurs et la création de projets électroniques. Voici un aperçu pour bien comprendre cet environnement et ses principales fonctions.

- Téléchargement et installation de l'IDE Arduino :

<https://www.arduino.cc/en/software>

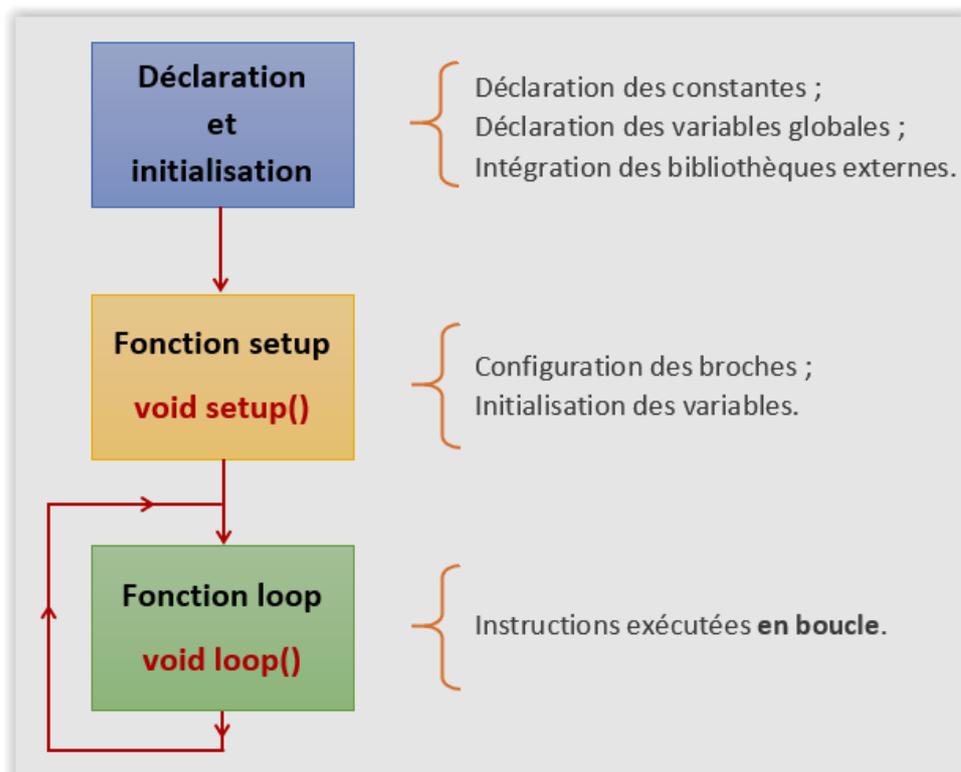
#### Lancez l'interface Arduino



L'interface de l'IDE Arduino est conçue pour être simple d'utilisation. Voici les principales parties :

- **Barre d'outils** : Elle contient des boutons pour les actions principales :
  - **Vérifier (✓)** : Vérifie ton code pour s'assurer qu'il n'y a pas d'erreurs.
  - **Téléverser (→)** : Envoie (téléverse) ton code à la carte Arduino.
  - **Nouvelle fenêtre** : Crée un nouveau fichier de code.
  - **Ouvrir** : Ouvre un projet existant.
  - **Sauvegarder** : Sauvegarde ton projet actuel.
  - **Moniteur Série** : Ouvre une fenêtre pour voir les données envoyées ou reçues par la carte Arduino.
- **Éditeur de code** : La grande zone blanche où tu écris ton programme (appelé un "sketch"). C'est ici que tu écris les instructions que ta carte Arduino va exécuter.
- **Fenêtre de messages** : En bas, tu verras les messages d'erreur ou de succès après avoir vérifié ou téléversé ton code.

## Squelette d'un programme Arduino



Nous allons d'abord regarder la structure générale du programme avant de nous pencher sur les détails comme la déclaration des variables.

Dans tout programme Arduino, il y a deux fonctions très importantes :

## **setup() et loop().**

### **Fonction setup() :**

Cette fonction est exécutée une seule fois au démarrage de l'Arduino. Elle sert à préparer tout ce dont l'Arduino aura besoin pour fonctionner.

- On peut y indiquer que l'Arduino doit allumer l'écran, configurer la vitesse d'un moteur ou même définir quelles broches seront utilisées pour commander des éléments comme des LED.

### **Fonction loop() :**

Après les préparations, l'Arduino passe dans la fonction loop(), qui tourne en boucle, encore et encore, tant que l'Arduino est sous tension. C'est ici que les actions répétées ou continues se déroulent.

## **Programme minimum**

```
#include <Arduino.h>

// Déclaration des variables (optionnelle)

void setup() {

// Initialisation et configuration des entrées/sorties

}

void loop() {

// la fonction loop() s'exécute en boucle aussi longtemps que l'Arduino est sous tension

}
```

## Introduction du Moniteur Série

Qu'est-ce que le moniteur série ?

- Le moniteur série est une fenêtre spéciale dans l'environnement Arduino (IDE) qui permet de communiquer avec la carte Arduino.
- Il affiche les messages envoyés par l'Arduino à l'ordinateur, un peu comme un écran qui montre ce que pense ou fait la carte.

### Pourquoi l'utiliser ?

- C'est très pratique pour vérifier ce qui se passe dans le programme, surtout quand tu veux savoir si une action est bien faite, comme lire une valeur de capteur ou simplement savoir si la carte fonctionne.
- Cela te permet de déboguer (trouver des erreurs) et d'observer ce que fait la carte sans avoir à brancher des LEDs ou des capteurs.

### Ajout d'un message dans le moniteur série :

```
void setup(){
  Serial.begin(9600);
  Serial.println("Bonjour, bienvenue à l'atelier Arduino !");
}

void loop(){
  // Rien ici pour l'instant
}
```

- **Serial.begin(9600)** : Cela démarre la communication série entre l'Arduino et l'ordinateur à une vitesse de 9600 bauds (c'est l'unité de vitesse en communication).
- **Serial.println("Bonjour, bienvenue à l'atelier Arduino !")** : Cette ligne envoie le texte "Bonjour, bienvenue à l'atelier Arduino !" au moniteur série. C'est comme si l'Arduino te disait bonjour !