ARDUINO : LES BASES

1. Cartes Arduino

Arduino est une plateforme, de prototypage électronique qui est composée de composants et de logiciels facile à utiliser. Elle est entièrement libre de droit.

Les cartes Arduino sont faciles à programmer. On peut y connecter d'autres composants électroniques sans soudure, tels que des boutons, des potentiomètres, des LED, des capteurs, des moteurs, des écrans, et même des cartes additionnelles (appelées shields) pour étendre encore les fonctionnalités.

Les premières cartes Arduino ont été développées en 2005 par une équipe de jeunes passionnés de programmation qui avaient besoin de cartes simples et bon marché à des fins éducatives. Plusieurs cartes plus ou moins puissantes ont été conçues dans les années qui ont suivi, de différentes tailles et avec différentes options sur la carte (Wi-Fi, Bluetooth ...).

2. Qu'est-ce qu'un microcontrôleur ?

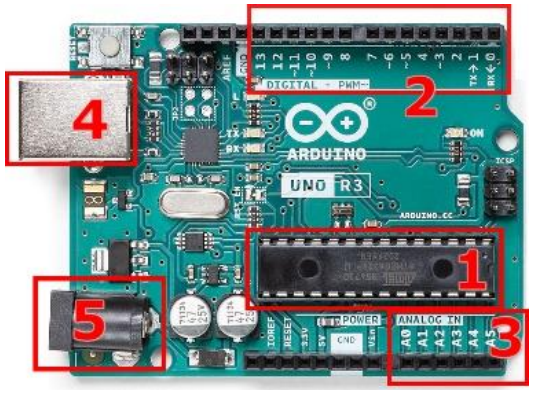
Les cartes Arduino font partie de la famille des microcontrôleurs.

Un microcontrôleur (µc, uc, ou encore MCU en anglais) est un circuit intégré compact, qui regroupe les éléments essentiels d'un ordinateur (microprocesseur associé à un ou plusieurs périphériques, interfaces d'entrées-sorties, mémoires).

ATmega328P (microcontrôleur de la carte Arduino Uno Rev 3)		
Tension d'alimentation	1,8 V à 5,5 V	
Mémoire Flash (programmable)	32 ko	
Mémoire vive statique (SRAM)	2 ko	
Mémoire morte (EEPROM)	1 ko	
Entrées/sorties (GPIO)	23	
Fréquence d'horloge	20 MHz	

On peut comparer un microcontrôleur à un ordinateur classique, mais sans système d'exploitation et avec une puissance de calcul considérablement plus faible. Les microcontrôleurs sont inévitables dans les domaines de l'informatique embarquée, de l'automatique et de l'informatique industrielle. Ils permettent de réduire le nombre de composant et de simplifier la création de cartes électroniques logiques.

3. Arduino UNO



	Composant	Fonction
1	Microcontrôleur	Faire des calculs, stocker le programme
2	Entrées/Sorties numériques	Acquérir / Commander
3	Entrées Analogiques	Acquérir
4	Connecteur USB	Alimenter la carte et transférer le programme
5	Connecteur alimentation	Alimenter la carte en 7-12V

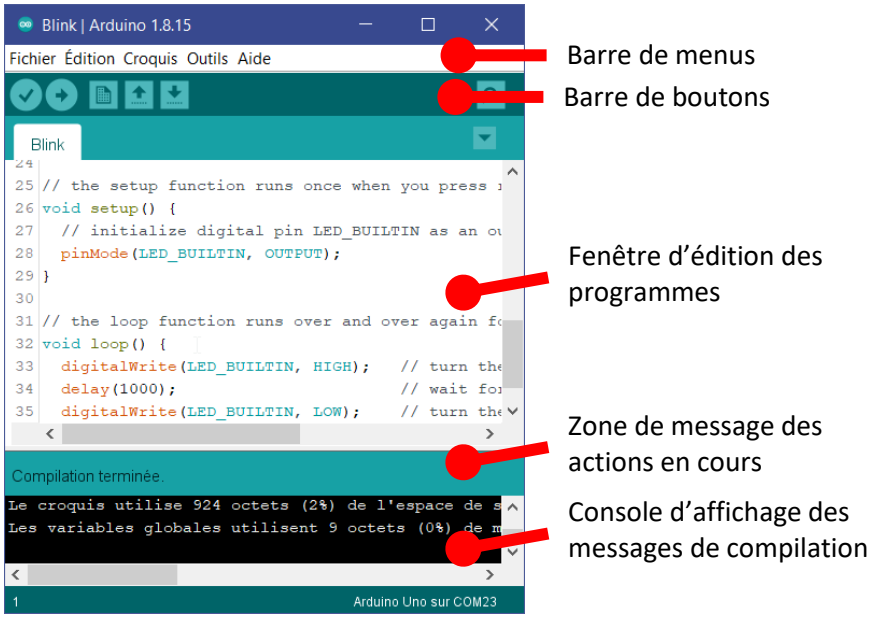
4. IDE

Un environnement de développement intégré (abrégé EDI en français ou IDE en anglais, pour integrated development environment) est une application logicielle qui aide les programmeurs à développer efficacement le code logiciel.

Le logiciel Arduino IDE fonctionne sur Mac, Windows et Linux. C'est grâce à ce logiciel que nous allons créer, tester et envoyer les programmes sur l'Arduino.

L'IDE est téléchargeable à l'adresse suivante : http://arduino.cc.

Une version en ligne de l'éditeur de code est disponible à cette adresse : https://create.arduino.cc/editor



Barre de boutons		
	Vérifier	Vérifier que le programme écrit ne présente pas de bugs afin de s'exécuter correctement.
	Téléverser	Transférer votre programme compilé dans la mémoire de votre carte Arduino
	Nouveau	Créer de nouveaux programmes
	Ouvrir	Accéder aux programmes d'exemples de l'IDE ou aux programmes présents sur votre machine
	Enregistrer	Sauvegarder le travail que vous avez réalisé afin d'y revenir quand vous le souhaitez

Pour mettre en œuvre une carte Arduino, il faut suivre les étapes suivantes :

1. Écrire un programme : le langage Arduino est proche du C et du C + + ;
2. Compiler le programme : il s'agit en fait de transformer le code « humain » saisi par l'utilisateur en langage « machine » c'est-à-dire une succession de 0 et de 1. L'ordinateur va alors vérifier la syntaxe du code ;
3. Téléverser le programme : c'est-à-dire transférer le programme sur la carte pour qu'il soit mis en œuvre.

5. Structure d'un programme

Le programme est lu par le microcontrôleur du haut vers le bas.

Un programme est divisé en trois parties : en-tête déclaratif, fonction setup et fonction loop

En-tête déclaratif

(Facultatif)

- Fichiers d'inclusion
- Déclaration des constantes
- Déclaration des variables globales

Fonction Setup

void setup ()

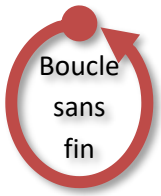
Fonction d'initialisation

- Déclaration des variables locales
- Configuration des broches
- Initialisation des variables
- Initialisation des fonctionnalités
- Initialisation des interruptions

Cette fonction est appelée une seule fois lorsque le programme commence. C'est pourquoi c'est dans cette fonction que l'on va écrire le code qui n'a besoin d'être exécuté qu'une seule fois.

Fonction Loop

void loop ()



Fonction principale

- Instructions exécutées en boucle

C'est donc dans cette fonction que l'on va écrire le cœur du programme. Elle tourne en boucle après l'exécution de la fonction setup() et ce jusqu'à la mise hors tension du microcontrôleur.

5.1. Les instructions

Les instructions sont des lignes de code qui disent au programme : « fais ceci, fais cela… » Ce sont donc les ordres qui seront exécutés par l'Arduino.

Il est très important de respecter exactement la syntaxe ; faute de quoi, le code ne pourra pas être exécuté.

Il est formellement interdit de mettre des accents en programmation, sauf dans les commentaires.

5.2. Les points-virgules ;

Les points-virgules terminent les instructions. Si par exemple on dit dans notre programme : « appelle la fonction avancerLeRobot », on doit mettre un point-virgule après l'appel de cette fonction.

Remarque : Lorsque le code ne fonctionne pas, c'est souvent parce qu'il manque un point-virgule.

5.3. Les accolades { }

Les accolades sont les « conteneurs » du code du programme. Elles sont propres aux fonctions, aux conditions et aux boucles. Les instructions du programme sont écrites à l'intérieur de ces accolades.

5.4. Les commentaires // ou /* */

Les commentaires sont des lignes de code qui seront ignorées par le programme. Elles ne servent à rien lors de l'exécution du programme. Ils permettent d'annoter et de commenter le programme.

6. Analyse d'un programme : blink

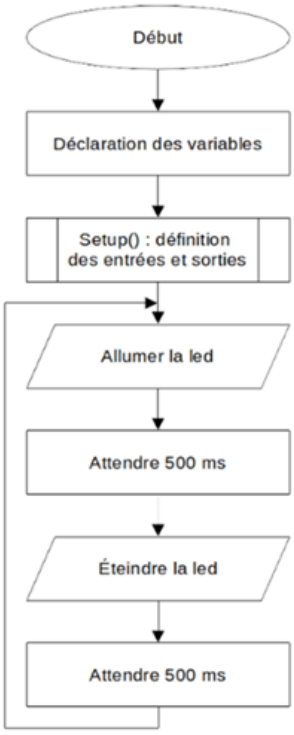
Les cartes Arduino possèdent une LED reliée à la broche numérique 13. Le programme suivant permet de la faire clignoter (blink en anglais) afin de vérifier le bon fonctionnement de la carte.

Clignotement_led §

```
1  /*
2    Code 1 - destiné à l'Arduino UNO
3    Objectif: faire clignoter la LED montée sur la broche 13
4  */
5
6  //***** EN TÊTE = déclaration des variables, des constantes,indication de l'utilisation de bibliothèques *****
7  // L'en-tête est lu une seule fois, au démarrage du programme
8
9  //***** FONCTION SETUP = Code d'initialisation *****
10 // La fonction setup() est exécutée en premier et une seule fois, au démarrage du programme
11
12 void setup()     // début de la fonction setup()
13 {
14     pinMode(13, OUTPUT);  // Initialise la broche 13 comme sortie
15 }               // fin de la fonction setup()
16
17 //***** FONCTION LOOP = Boucle sans fin = cœur du programme *****
18 // la fonction loop() s'exécute sans fin en boucle aussi longtemps que l'Arduino est sous tension
19
20 void loop()      // début de la fonction loop()
21 {
22     digitalWrite(13, HIGH);  // Met la broche 13 de la led au niveau haut = allume la LED
23     delay(500);                      // Pause de 500 ms
24     digitalWrite(13, LOW);   // Met la broche 13 de la led au niveau bas = éteint la LED
25     delay(500);                      // Pause 500 ms
26 }               // fin de la fonction loop()
```

Associer les fonctions de l'algorigramme aux lignes du programme.

Algorigramme **Lignes du programme associées**



Lignes : .....

Lignes : .....

Lignes : .....

Lignes : .....

Lignes : .....

Lignes : .....

7. Exercice : Lancer un SOS

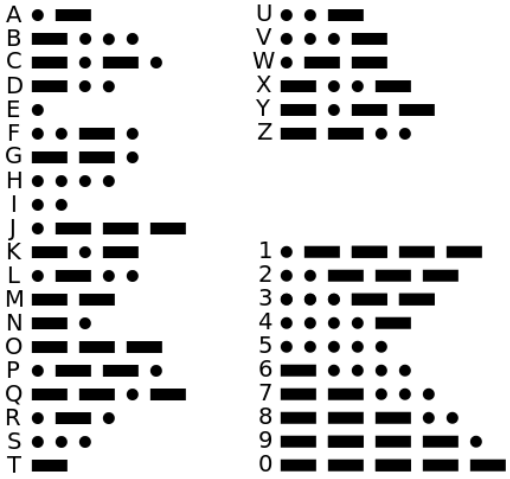
En vous inspirant de l'exercice précédent, on vous demande de coder l'Arduino pour que sa LED transmette un signal de SOS lumineux.

En code morse, les lettres sont codées à l'aide d'une succession d'impulsions courtes et longues.

Code morse international

- Un tiret est égal à trois points.
- L'espacement entre deux éléments d'une même lettre est égal à un point.
- L'espacement entre deux lettres est égal à trois points.
- L'espacement entre deux mots est égal à sept points.

Attention : il n'y a pas d'espace entre les lettres d'un SOS.



Voici à quoi ressemble un SOS en morse :