

Langage ARDUINO

I. Introduction

Le langage Arduino est un langage de programmation convivial et populaire adapté aux microcontrôleurs Arduino. Il simplifie la programmation, s'appuyant sur C et C++, et est couramment utilisé pour le développement de projets électroniques et robotiques.



II. Syntaxe d'un programme Arduino

Dans le contexte d'Arduino, le programme est appelé le "code" et ce code est divisé en deux parties distinctes :

```
// fonction d'initialisation de la carte
void setup()
{
    // contenu de l'initialisation
}

// fonction principale, elle se répète (s'exécute) à l'infini
void loop()
{
    // contenu de votre programme
}

Mr A. OUAANABI
```

❖ Fonction : void setup()

Il s'agit d'une fonction qui s'exécute une seule fois lors du démarrage de la carte Arduino. On peut également la désigner comme une "fonction d'initialisation" qui englobe les configurations, les conditions initiales et les déclarations des entrées/sorties.

❖ Fonction : Void loop()

Donc, dans la fonction loop(), on écrit ce que le programme doit faire. Cette fonction est spéciale car elle est toujours en train de travailler. Elle ne s'arrête jamais, elle continue de faire et refaire son travail encore et encore. C'est comme une boucle sans fin.

III. Les règles générales de langage ARDUINO (C++)

1. Instructions

Elles sont des lignes de code écrites dans le langage de programmation C++ spécifique à la plateforme Arduino. Ces instructions sont utilisées pour contrôler les microcontrôleurs Arduino et créer des programmes qui effectuent diverses tâches.

```
digitalWrite (13, HIGH); // Instruction 1
delay (1000); // Instruction 2
Mr A. OUAANABI
```

2. Les points virgules

Les points virgules terminent les instructions. Si par exemple je dis dans mon programme : « appelle la fonction » je dois mettre un point-virgule après l'appel de cette fonction :

Exemple : digitalWrite (13, HIGH) ; ←

3. Les commentaires

En langage Arduino, les commentaires sont utilisés pour expliquer le code et fournir des informations utiles aux programmeurs sans affecter l'exécution du programme.

- **Commentaires sur une seule ligne :** Utilisez « // » pour indiquer un commentaire sur une seule ligne.
- **Commentaires sur plusieurs lignes :** Pour créer des commentaires sur plusieurs lignes, vous pouvez utiliser « /* » pour ouvrir le commentaire et « */ » pour le fermer.

Exemple : digitalWrite (13, HIGH) ; // allumer la LED connectée à la broche 13

IV. Les variables

En langage Arduino C++, les variables sont des emplacements de mémoire utilisés pour stocker des données. Les variables sont essentielles pour stocker des valeurs temporaires, des résultats de calculs, des données provenant de capteurs, etc.

1. Type 1 : int

Les entiers (integers) sont utilisés pour stocker des nombres sur l'Arduino Uno et autres cartes ATmega. Un int représente une valeur de 16 bits, allant de -32 768 à 32 767.

Exemple : `int x=30 ; // on déclare X en type entier et initialisé par une valeur de 30`

2. Type 2 : float

Le type de données pour les nombres à virgule flottante est utilisé pour approximer des valeurs analogiques avec une résolution supérieure. Ils peuvent atteindre 3.4028235E+38 et descendre jusqu'à -3.4028235E+38, stockés sur 32 bits (4 octets).

Exemple : `float y=2.37 ; // on déclare y en type entier et initialisé par une valeur de 2.37`

3. Type 3 : bool

Un booléen (bool) peut prendre l'une des deux valeurs, vrai ou faux. (Chaque variable booléenne occupe un octet de mémoire.)

Exemple : `bool y=true ;`

4. Type 4 : char

Un type de données utilisé pour stocker une valeur de caractère. Les littéraux de caractère sont écrits entre guillemets simples, comme ceci : 'A' (pour plusieurs caractères - des chaînes de caractères - utilisez des guillemets doubles : "ABC").

Exemple : `char var = 'A' ;`

5. Autre types

Dans ce tableau, nous regroupons d'autres types qui proviennent des types mentionnés précédemment.

Type	Description	Syntaxe
byte	Un octet stocke un nombre non signé de 8 bits, de 0 à 255.	<code>byte var = 128;</code>
Short	Un short est un type de données de 16 bits. Sur tous les Arduino (à base d'ATmega et ARM), un short stocke une valeur de 16 bits (2 octets). Cela donne une plage de -32 768 à 32 767.	<code>short var = 1034;</code>
double	Nombre en virgule flottante à double précision. Sur l'Arduino Uno et d'autres cartes basées sur l'ATMEGA, cela occupe 4 octets.	<code>double var = -0.023;</code>
long	Les variables de type long sont des variables de taille étendue pour le stockage de nombres entiers, et elles stockent 32 bits (4 octets), avec une plage allant de -2 147 483 648 à 2 147 483 647.	<code>long var = 1239462223;</code>
string	En C++ sur Arduino, le type de données String est utilisé pour manipuler des chaînes de caractères de manière conviviale.	<code>String Texte = "Bonjour!";</code>