En une page... les cartes Arduino

Résumé. Au laboratoire, pour contrôler des expériences ou des dispositifs complexes, il serait parfois plus qu'utile de faire appel à un programme informatique.

Une carte Arduino Uno version 3 [1] (figure 1 [2]) est une carte électronique animée par un microcontrôleur ATmega328P 8 bits capable d'échanger de l'information avec l'extérieur par le biais d'entrées/sorties, dites I/O pour input/output. Au regard d'un ordinateur conventionnel, cette carte possède des ressources extrêmement limitées (tableau 1), mais plus que suffisantes pour interagir avec un circuit électrique, des capteurs ou envoyer des consignes à des composants. Toutefois, les variables globales seront souvent préférées aux variables locales, en particulier pour les tableaux, de manière à éviter la duplication de grandes quantités de données. Il pourra alors être intéressant de s'intéresser à la notion de pointeur qui permet justement de pointer une case dans ce tableau.

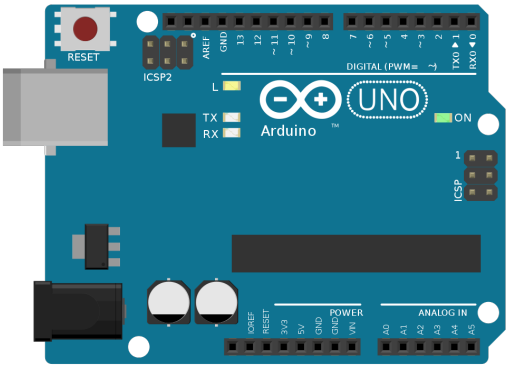

FIGURE 1 – Représentation d'une carte Arduino Uno

Caractéristique	Valeur
Fréquence processeur	16 MHz
Mémoire RAM	2 ko
Mémoire flash	32 ko
I/O analogiques	6
I/O numériques	19
I/O interrupt	2
Courant continu par I/O	20 mA

TABLEAU 1 – Caractéristiques de la carte

Ce type de carte se programme en utilisant un langage proche du langage C, dont l'éditeur de développement est librement téléchargeable [3]. Pour l'essentiel, nous pouvons considérer que les deux langages sont identiques [4], à l'exception notable des tailles des entiers et des nombres à virgule flottante. Deux fonctions de type void, qui ne renvoient rien, seront considérées comme étant indispensables, `setup` et `loop` qui sont exécutée respectivement à l'initialisation et en boucle jusqu'à la coupure de l'alimentation de la carte. Donnons l'exemple qui consiste à faire clignoter une DEL :

```
// Utilise la DEL sur la broche 13
int led = 13;

void setup() {
  // La broche DEL est une sortie
  pinMode(led, OUTPUT);
}

void loop() {
  // Allume la DEL avec une tension haute
  digitalWrite(led, HIGH);
  // Attend une seconde en millisecondes
  delay(1000);
  // Éteint la DEL avec une tension de zéro
  digitalWrite(led, LOW);
  delay(1000);
}
```

Le logiciel ainsi créé sera compilé puis téléversé sur la carte par la liaison série via le câble USB. Elle peut également servir à transférer de l'information de la carte vers le moniteur série de l'ordinateur. Pour informer l'ordinateur de l'état de la DEL à chaque commutation, il serait possible d'ajouter dans la fonction `setup`

```
Serial.begin(9600);
```

puis respectivement, avant et après la fonction `delay`

```
Serial.println("DEL allumée");
Serial.println("DEL éteinte");
```

Le langage étant un dérivatif du langage C, on fera attention à bien mettre un `;` à chaque fin d'instruction.

Parfois, un appui sur un bouton doit déclencher une instruction sur le champ. La notion d'interruption et la fonction `attachInterrupt` seront alors à considérer. Celle-ci permet de mettre en pause le code en cours d'exécution et de traiter immédiatement la fonction demandée par l'appui sur le bouton [5].

Pour étendre les possibilités de la carte, il sera possible d'utiliser des bibliothèques qu'il est possible de gérer directement dans l'éditeur.

1. Fiche technique d'une carte Arduino Uno : `https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf`
2. By PhilippHenkel - Own work, CC BY-SA 4.0, `https://commons.wikimedia.org/w/index.php?curid=74273108`
3. Éditeur Arduino : `https://www.arduino.cc/en/software`
4. Documentation du langage : `https://docs.arduino.cc/programming/`
5. Il faudra impérativement connecter le bouton ou le capteur aux broches 2 ou 3 de la carte Arduino Uno, ce sont les seules à supporter physiquement ce type d'action.

🅭🅯🄏🄎