

22 Utilisation du microcontrôleur Arduino UNO

A Présentation du microcontrôleur

Un microcontrôleur est un petit ordinateur intégré dans un système de petite dimension. Il dispose de plusieurs entrées et sorties permettant de lire et de générer des signaux électriques.

En Physique-Chimie, on l'utilise pour analyser et traiter les informations de différents capteurs.

➤ Ports numériques (en orange)

Les ports numériques peuvent être utilisés en sortie ou en entrée, ils n'admettent que deux valeurs : LOW (0 V) et HIGH (5 V).

Les ports 1 et 0 sont uniquement utilisés pour connecter deux cartes Arduino.

Les ports PWM (signe ~) peuvent simuler une sortie analogique en basculant rapidement entre leurs valeurs haute et basse, la tension moyenne en sortie est alors une tension moyenne située entre 0 V et 5 V.

➤ Entrées analogiques (en rouge)

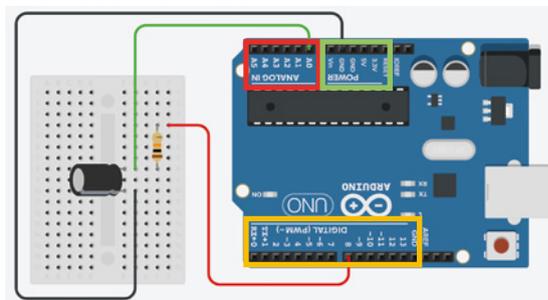
Les entrées analogiques fonctionnent à l'aide d'un convertisseur analogique-numérique (CAN) qui transforme une tension d'entrée en une valeur comprise entre 0 et 1023 (carte fonctionnant sur 10 bits).

Les entrées analogiques acceptent des tensions entre 0 V et 5 V, toute tension inférieure à 0 V se voit affecter la valeur 0 et toute tension supérieure à 5 V se voit affecter la valeur maximale 1023.

La valeur maximale de 5 V peut être modifiée pour être supérieure ou inférieure.

➤ Tensions de référence et d'alimentation (en vert)

Le port 3.3V délivre une tension de 3,3 V, le port 5V délivre une tension de 5 V. Les ports GND sont à la tension de référence du microcontrôleur, définie comme 0 V pour les mesures. Le port VIN est à la tension de l'alimentation externe s'il y en a une.



B Structure type d'un code Arduino

Le langage Arduino se base sur le langage informatique C++. La structure type d'un code Arduino est la suivante : déclaration des variables, initialisation des ports, écriture du programme principal.

La commande du microcontrôleur peut aussi se faire par langage Python grâce à la librairie **pyduino**.

```
int sortie=2; // on définit une variable qui contient le numéro du port utilisé
int duree = 0; // on définit une autre variable pour choisir la durée pendant laquelle le son est émis
int temps=1116;
```

```
void setup() {
  // on indique à Arduino que la borne 'sortie' enverra une tension:
  pinMode(sortie, OUTPUT);
}
```

```
void loop() {
  while(durée<500) { // le cycle peut s'effectuer jusqu'à ce que la variable durée atteigne la valeur 500
    digitalWrite(sortie, LOW);
    delayMicroseconds(temps);
    digitalWrite(sortie, HIGH);
    delayMicroseconds(temps);
    durée++;
  }
  duree = 0;
}
```