

Tutoriel Arduino 3 : Communication

Objectif

L'objectif de ce troisième tutoriel est d'établir une communication c'est à dire échanger des données ou bien des commandes entre une carte Arduino et un ordinateur ou encore entre deux cartes Arduino. Pour illustrer ce principe, nous enverrons la valeur lue sur la broche A0 vers l'ordinateur et inversement nous commanderons l'allumage d'une DEL connectée sur la broche 13 en envoyant un caractère depuis l'ordinateur.

Schéma électrique

Pour communiquer avec la carte Arduino nous allons utiliser l'interface série de la carte. Cette interface était historiquement connectée au port série de l'ordinateur mais avec l'avènement de l'interface USB ce genre de port a fini par disparaître et pour le coup l'interface série de la carte Arduino est accessible *via* son interface USB. Pour communiquer entre la carte et un ordinateur il suffit donc de la connecter à l'ordinateur avec un câble USB.

Pour ce tutoriel il vous faudra connecter une DEL sur la broche 13 (cf tutoriel 1) et un potentiomètre ou tout autre type de capteur sur la broche A0 (cf tutoriel 2).

Programme Arduino

Pour utiliser l'interface série, le langage Arduino définit les commandes suivantes :

Serial.begin(speed) : pour initialiser l'interface série. Le paramètre `speed` est utilisé pour configurer la vitesse de transmission des données (aussi appelé *baudrate*).

Serial.available() : pour savoir si des caractères sont en attente de lecture/réception sur l'interface série.

Serial.read() : pour lire/recevoir un caractère sur l'interface série.

Serial.write(c) : pour écrire/envoyer un caractère `c` sur l'interface série.

Serial.print(val) : pour écrire la valeur `val` sur l'interface série,

Serial.println(val) : pour écrire la valeur `val` sur l'interface série avec en plus un saut à la ligne suivante,

Commençons par simplement envoyer les valeurs lues sur la broche A0 connectée au capteur ou au potentiomètre. Nous allons utiliser la commande **Serial.println**(val) pour envoyer sur l'interface série la valeur lue sur la broche A0 par un appel à **analogRead**(). C'est aussi simple que cela. Toutefois, il est conseillé de rajouter un appel à **delay**(100) pour éviter de surcharger l'affichage sur la fenêtre de la ligne série.

Continuons en implémentant la commande de la broche 13 par l'interface série. En envoyant le caractère '1', l'utilisateur allume la DEL tandis qu'envoyer n'importe quel autre caractère éteint la DEL.

Dans la boucle **loop** il faut tout d'abord tester si l'interface série a des caractères en attente. Pour cela on utilise la commande **Serial.available**() . Cette commande retourne une valeur supérieure à 0 s'il y a effectivement des caractères en attente, 0 sinon. Si la valeur de retour est supérieure à 0, on appelle **Serial.read**() pour lire les caractères en attente, un caractère à la fois. En comparant ensuite la valeur du caractère à '1' nous savons

s'il faut mettre la broche 13 à **HIGH** ou à **LOW**. On utilisera bien sûr la commande `digitalWrite()`.

Voici le programme complet :

```
void setup()
{
  Serial.begin(9600);
  pinMode(13, OUTPUT);
}

void loop()
{
  if (Serial.available() > 0) {
    char c = Serial.read();
    if (c == '1') {
      digitalWrite(13, HIGH);
    } else {
      digitalWrite(13, LOW);
    }
  }
  int val = analogRead(A0);
  Serial.println(val);
  delay(100);
}
```