

# Les entrées et sorties analogiques

## I) Définitions

**Entrée** : Tous les événements qui peuvent être détectés (lus) par la carte Arduino et qui peuvent influencer le déroulement du programme

Physiquement il s'agit de broches dont la configuration permet de recevoir une tension provenant de : bouton poussoir, capteur divers...

**Sortie** : Il s'agit d'éléments extérieurs contrôlés par le programme. Concrètement se sera des broches sur lesquelles on aura connecté des : del, servomoteurs, buzzer, etc...

### Analogique :

Tout signal qui peut varier avec une infinité de valeurs (de manière analogue à une grandeur physique) :

- En entrée l' Arduino accepte des tensions analogiques de 0 à 5V. On dispose de capteurs compatibles ( mesure de : température , pression, luminosité, angle de rotation etc...)
- En sortie la carte Arduino ne dispose pas de vraie sortie analogique, mais seulement de sortie "PWM" ce qui est suffisant pour faire varier l'intensité lumineuse d'une del .

## II) Je teste les EXEMPLES DE PROGRAMMES fournis par Arduino

→ ils sont disponibles dans le menu **fichier, exemples, ...**

### Sortie analogique :

Chargez le fichier exemples/03.Analog/Fade ;

Ce fichier commande l'éclairage progressif d'une DEL connectée en D9.

Modifiez le fichier pour une DEL connectée en D6.

Connectez la DEL en D6, téléversez le programme sur la carte et vérifiez son bon fonctionnement.

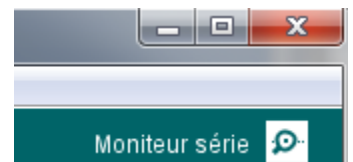
### Entrées analogiques :

Chargez le fichier exemples/01.Basics/AnalogReadSerial ;

Ce fichier lit la valeur analogique (tension) du composant connecté en A0 (typiquement un potentiomètre) et l'affiche sur le moniteur série.

Connectez le potentiomètre en A0, téléversez le programme sur la carte et vérifiez son bon fonctionnement, en activant le moniteur série :

*Pour activer le moniteur série, cliquez sur l'outil Loupe de la fenêtre Arduino :*



> Quelle est la commande Arduino permettant de lire la valeur d'une entrée analogique ?

> Quelle est la plage de valeurs possibles ?

## III) Analyse du programme d'exemple AnalogInOutSerial :

Imprimez le programme.

- a) Surlignez en bleu l'instruction permettant de lire l'entrée analogique (valeur du potentiomètre)
- b) Surlignez en rouge l'instruction permettant d'écrire la grandeur analogique en sortie (intensité d'éclairage de la LED)

- c) Surlignez en vert les instructions permettant d'afficher des informations sur le Moniteur série.  
Surlignez également en vert la ligne permettant d'initialiser la liaison série (et de fixer son débit).

## IV) EXERCICES

### EX 1 : VARIATEUR DE LUMIERE

On utilisera :

1. Une del rouge connectée en D5 et une del bleue connectée en D6
2. Un potentiomètre connecté en A0



Un potentiomètre est une résistance variable.



Il est constitué d'un élément résistif en forme d'arc de cercle qui est électroniquement divisé en deux parties par un curseur. Le déplacement de ce curseur sur l'élément résistif modifie la proportion de la résistance de chaque côté du curseur.

**Les broches extérieures du potentiomètre étant connectées à 5 V et GND la tension sur le curseur va évoluer entre 0 et 5V.**

C'est l'image de cette valeur qui sera lue sur la broche analogique de l'Arduino.

**La valeur lue sera proportionnelle à l'angle de rotation du potentiomètre (valeur lue de 0 à 1023 pour une variation du mini au maxi du potentiomètre)**

**Pour 0V, potentiomètre au minimum (en butée sens trigonométrique) la valeur lue est 0.**

**Pour 5V, potentiomètre au maximum (en butée sens horaire) la valeur lue est 1023.**

### EX 2 : SYSTEME D'ECLAIRAGE AUTOMATIQUE.

On utilisera :

- Une del connectée en D5
- Un capteur de luminosité connecté en A1
- Un potentiomètre connecté en A0

### Question

Écrire le code pour :

1. Créer une variable "luminosité" lui affecter la valeur issue du capteur de luminosité , noter les valeurs maximale et minimale (capteur en pleine lumière et dans l'obscurité)
2. Créer un programme qui permet d'allumer la del lorsque intensité lumineuse est proche de l'obscurité, tester le programme.
3. Modifier le programme afin d'utiliser le potentiomètre pour régler le seuil de déclenchement de l'éclairage, tester le programme.

- Utiliser 2 seuils différents (hystérésis) pour l'allumage et l'extinction : la del s'allumera lorsque la luminosité sera inférieure à la valeur (seuil -10) et s'éteindra pour une luminosité supérieure à (seuil +10).

## V) PROJET COMPLET : Le Banc

**Dans cette partie vous allez utiliser vos connaissances pour réaliser un lampadaire d'éclairage pour banc.**

**Ce lampadaire est destiné à être intégré à une maquette de banc**

**Vous disposez de:**

- un détecteur de luminosité connecté en A0
- un détecteur de mouvement connecté en D5
- un bouton poussoir connectée en D2
- deux potentiomètres connectés en A1 et A2
- une del pour simuler l'éclairage connectée en D3.



### **Description du fonctionnement final (cahier des charges) :**

Le lampadaire doit s'allumer lorsqu'il fait nuit (luminosité faible) et qu'il détecte un individu à proximité du banc ou sur le banc.

Il s'éteint lorsque la luminosité est suffisante ou après une temporisation lorsque qu'il ne détecte plus personne.

Un bouton permet de tester le lampadaire, une action sur le bouton allume le lampadaire pendant 10s.

Le seuil de détection de la luminosité est réglable.

La temporisation est réglable entre 20s et 2mn.

## ANNEXE : listing du programme d'exemple AnalogInOutSerial :

```
/*
  Analog input, analog output, serial output
*/

// These constants won't change. They're used to give names
// to the pins used:

const int analogInPin = A0; // Analog input pin of the potentiometer const
const int analogOutPin = 9; // Analog output pin of the LED

int sensorValue = 0; // value read from the pot
int outputValue = 0; // value output to the PWM (analog out)

void setup() {

  // initialize serial communications at 9600 bps:
  Serial.begin(9600);

}

void loop() {

  // read the analog in value:
  sensorValue = analogRead(analogInPin);

  // map it to the range of the analog out:
  outputValue = map(sensorValue, 0, 1023, 0, 255);

  // change the analog out value:
  analogWrite(analogOutPin, outputValue);

  // print the results to the serial monitor:
  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);

  // wait 2 milliseconds before the next loop
  // for the analog-to-digital converter to settle
  // after the last reading:
  delay(2);
}
```