



### CE QUE NOUS ALLONS FAIRE:

En plus des pattes numériques, l'Arduino possède 6 pattes qui peuvent être utilisées pour des entrées analogiques. Ces entrées prennent un voltage (entre 0 et 5V) et le convertissent en un nombre compris entre 0 (0V) et 1024 (5V) (résolution de 10 bits). Un composant très utile qui utilise ces entrées est le potentiomètre (aussi appelé résistance variable). Quand il est connecté avec 5V entre ses pattes extérieures, la patte centrale ressort des valeurs comprises entre 0 et 5V en fonction de l'angle selon lequel son axe est tourné (ex : 2,5V au milieu). Nous pouvons ensuite utiliser cette valeur comme variable de notre programme.

### LE CIRCUIT:

#### Composants :



**CIRC-08**  
Feuille de Connexions  
x1



Connecteur 2 pattes  
x4



Potentiomètre  
10k ohms  
x1

Fil

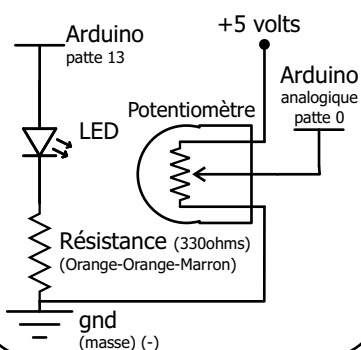


LED Jaune  
x1



Résistance 330 Ohms  
Orange-Orange-Marron  
x1

#### Schéma



#### Internet

.:Télécharger:.

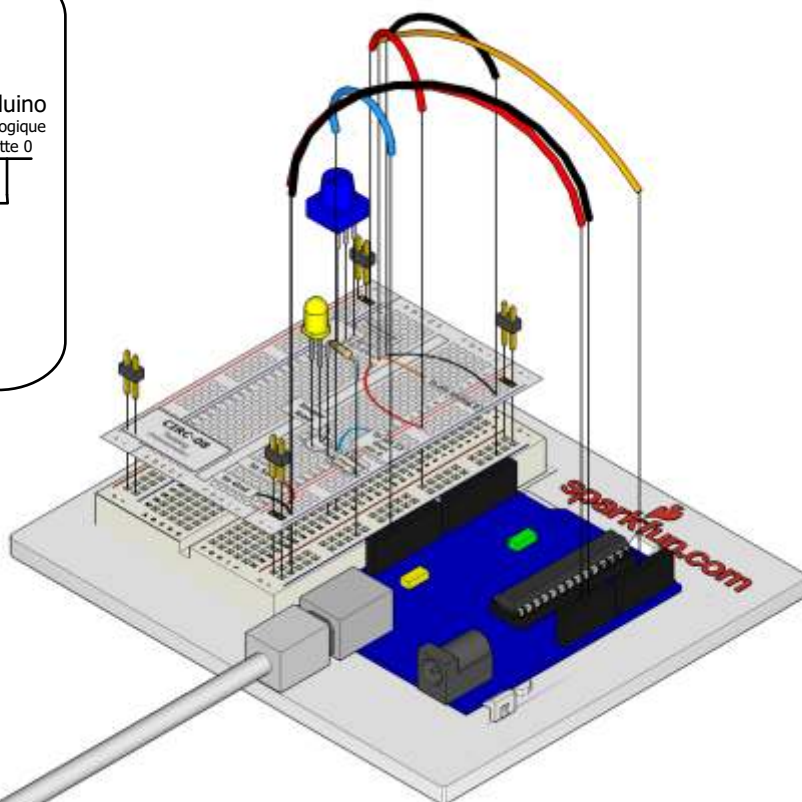
Feuille de Connexions

<http://ardx.org/BBS08S>

.:Voir:.

Vidéo de Montage

<http://ardx.org/VIDE08>



**CODE** (Il n'est pas nécessaire de taper quoi que ce soit)**File > Examples > 3.Analog > AnalogInput**(exemple du site [arduino.cc](http://arduino.cc), allez y faire un tour pour trouver d'autres idées)

```

/* Entrée analogique
 * Explication de l'entrée analogique en lisant un capteur analogique
sur
 * Le pin analogique 0 et en allumant une LED connectée au pin digital 13.
 * Le temps pendant lequel la LED sera allumée ou éteinte dépend de la valeur obtenue par
 * analogRead().
 * Créé David Cuartielles
 * Modifié le 16 Juin 2009
 * Par Tom Igoe
 * http://arduino.cc/en/Tutorial/AnalogInput
 */

int sensorPin = 0;    // sélectionne le pin d'entrée pour le potentiomètre
int ledPin = 13;      // sélectionne le pin pour la LED
int sensorValue = 0;  // variable pour stocker la valeur venant du capteur

void setup() {
  pinMode(ledPin, OUTPUT); //déclare ledPin comme une sortie:
}

void loop() {
  sensorValue = analogRead(sensorPin); // lit la valeur du capteur:
  digitalWrite(ledPin, HIGH);          // met ledPin à l'état haut
  delay(sensorValue);                  // arrête le programme pendant <sensorValue> millisecondes:
  digitalWrite(ledPin, LOW);           // met ledPin à l'état bas:
  delay(sensorValue);                  // arrête le programme pendant <sensorValue> millisecondes:
}

```

**CELA FONCTIONNE PAS ?** (3 choses à essayer)**Fonctionne par intermittence**

On dirait un mauvais branchement du potentiomètre. Généralement vous pouvez le régler en l'enfonçant un peu plus.

**Ne fonctionne pas**

Vérifiez que vous n'avez pas connecté par erreur la sortie du potentiomètre sur le pin digital 2 au lieu du pin analogique 2. (La rangée de pins en dessous des pins d'alimentation)

**Toujours à l'envers**

Vous pouvez essayer de faire fonctionner le circuit à l'envers. Parfois ça fonctionne.

**AMÉLIORER LE MONTAGE****Commutation selon un seuil:**

Parfois vous voulez changer la valeur d'une sortie quand une valeur excède un certain seuil. Pour faire cela avec un potentiomètre, changez le code de la fonction loop() par.

```

void loop() {
  int threshold = 512;
  if(analogRead(sensorPin) > threshold){
    digitalWrite(ledPin, HIGH);}
  else{ digitalWrite(ledPin, LOW);}
}

```

La LED s'allume quand la valeur est au dessus de 512 (environ la moitié), vous pouvez ajuster la sensibilité en changeant la valeur du seuil.

**Luminosité:**

Contrôlons la luminosité de la LED directement avec le potentiomètre. Tout d'abord, il faut changer le pin auquel la LED est connecté. Déplacez le fil du pin 13 au pin 9 et changez cette ligne de code.

```
int ledPin = 13; ----> int ledPin = 9;
```

Puis changez le code dans loop.

```

void loop() {
  int value = analogRead(potPin) / 4;
  analogWrite(ledPin, value);
}

```

Chargez le code et voyez comme la luminosité de la LED varie en fonction du potentiomètre. (Note: la raison pour laquelle on divise la valeur par 4 est que la fonction analogRead() retourne une valeur entre 0 et 1023 (10 bits), et analogWrite() prend une valeur entre 0 et 255 (8 bits) )

**Contrôler un servo::**

C'est vraiment un bel exemple et il permet d'associer plusieurs circuits. Câblez le servo comme pour le circuit CIRC-04 et ouvrez le programme d'exemple Knob (**File > Examples >**

**Servo > Knob** ), puis changez une ligne de code.

```
int potpin = 0; ----> int potpin = 2;
```

Chargez sur votre Arduino et observez le servo tourner en fonction du potentiomètre.

**PLUS, PLUS, PLUS :**

Plus de détails, où acheter des composants, où poser plus de questions :

**<http://ardx.org/CIRC08>**