



CE QUE NOUS ALLONS FAIRE:

Alors que l'obtention de valeur d'un potentiomètre peut être sympa dans le cas d'un circuit contrôlé par l'homme, que faire pour un circuit contrôlé par notre environnement ? On va utiliser exactement le même principe mais à la place d'un potentiomètre on va utiliser une photorésistance (résistance basée sur la lumière). L'Arduino ne peut pas directement mesurer la résistance (elle mesure la tension) donc on utilise un pont diviseur de tension (<http://ardx.org/VODI>). La tension exacte de la patte de mesure est calculable, mais pour notre montage (on utilise la lumière relative) nous pouvons tester les valeurs et trouver celles qui marchent. Une faible valeur provient quand le capteur est très éclairé alors qu'une valeur importante arrive quand le capteur est dans le noir.

LE CIRCUIT:

Composants :



CIRC-09
Feuille de Connexions
x1



Connecteur 2 pattes
x4



Photo-Résistance
x1



Fil



Résistance 10k Ohms
Marron-Noir-Orange
x1

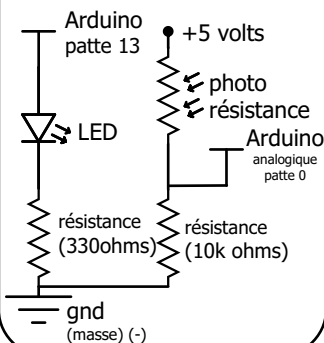


Résistance 330 Ohms
Orange-Orange-Marron
x1



LED Jaune
x1

Schéma



Internet

.:Télécharger:.

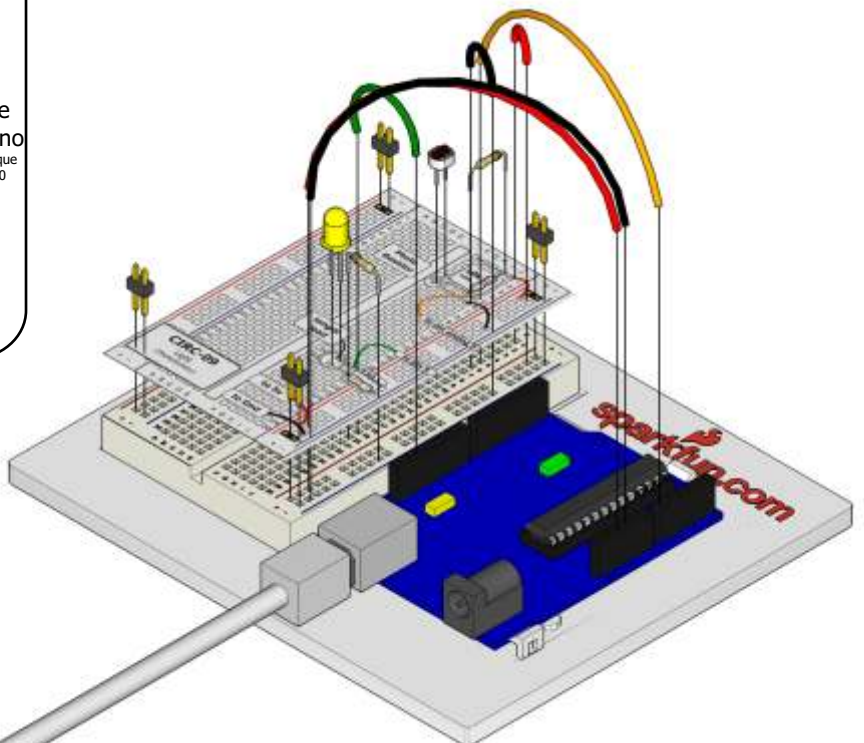
Feuille de Connexions

<http://ardx.org/BBLS09S>

.:Voir:.

Vidéo de Montage

<http://ardx.org/VIDE09>



CODE (Il n'est pas nécessaire de taper quoi que ce soit)**Téléchargez le code (<http://ardx.org/CODE09>)**

(et ensuite copiez le texte dans un nouveau sketch Arduino)

```

/*
*Un petit programme qui va changer
*l'intensité d'une LED en fonction de la lumière
*captée par la photorésistance.
*/

//patte de la photorésistance
int lightPin = 0; //la patte analogique ou
//la photorésistance est
//connectée
//la photorésistance n'est
//calibrée pour aucune unité
//ce n'est donc qu'un simple capteur
// de valeurs (relatives à la lumière).
)
//patte de la LED
int ledPin = 9;
//La patte à laquelle la LED est connectée
//nous allons contrôler la luminosité donc
//nous utilisons l'une des PWM(Pulse
//width Modulation)
< br>void setup()
{
  pinMode(ledPin, OUTPUT); //placé la LED en
                          //sortie
}
/*
* loop() -cette fonction commencera après le setup
*se termine et recommence
*/
void loop()
{
  int lightLevel = analogRead(lightPin); //lit le
                                          //niveau de
  luminosité
  lightLevel = map(lightLevel, 0, 900, 0, 255);
                //ajuste la valeur de 0 à 900 pour avoir de
  0 à 255
  lightLevel = constrain(lightLevel, 0, 255);
                //Vérifie que le niveau est entre 0 et 255
  analogWrite(ledPin, lightLevel); //Ecrit la valeur
}

```

CELA FONCTIONNE PAS ? (3 choses à essayer)**La LED reste noire**

C'est une erreur que nous continuons à faire encore et encore, si seulement ils pouvaient faire une LED qui fonctionne dans les deux sens. Débranchez la et faite lui faire un demi-tour.

La LED ne répond pas aux changements de luminosité

Puisque l'espacement des pattes de la photorésistance n'est pas standard, il est facile de les placer au mauvais endroit. Vérifiez que les branchements sont bons.

Ça ne fonctionne toujours pas ?

Vous devez être dans une pièce où il y fait trop sombre ou trop lumineux. Essayez d'allumer ou d'éteindre la lumière pour voir s'il y a des changements. Si vous avez une lampe de poche à portée de main essayez avec.

AMÉLIORER LE MONTAGE**Inverser la réponse :**

Peut être voulez-vous un fonctionnement inverse. Ne vous inquiétez pas nous pouvons facilement inverser le fonctionnement, changez juste :

```

analogWrite(ledPin, lightLevel); ---->
analogWrite(ledPin, 255 - lightLevel);

```

Téléchargez le et regardez le résultat :

Lampe de Nuit :

Plutôt que de faire varier la puissance de la LED en fonction de la luminosité, modifions le projet pour avoir un allumage ou une extinction de la LED en fonction d'un seuil de lumière.

Changez le code de loop() par :

```

void loop(){
  int threshold = 300;
  if(analogRead(lightPin) > threshold){
    digitalWrite(ledPin, HIGH);
  }else{
    digitalWrite(ledPin, LOW);
  }
}

```

Servo contrôlé par la lumière :

Utilisons nos nouvelles connaissances sur les capteurs de lumière pour contrôler un servo(et en même temps commençons à détourner le code Arduino). Branchez un servo sur la patte 9 (comme pour le CIRC-04) Ensuite ouvrez l'exemple KNOB (le même que pour le CIRC-08) **File > Examples > Servo > Knob**. Envoyez le code à la carte et regardez si il marche sans modification.

Utiliser toute la plage de déplacement du servo :

Vous avez du remarquer que le servo ne fonctionne que dans une zone réduite de sa plage de mouvements. C'est dû au fait qu'avec le système de pont diviseur de tension que nous utilisons, la tension sur la patte analog 0 n'est pas comprise entre 0 et 5 volts mais entre deux valeurs plus faibles (ces valeurs peuvent changer en fonction de votre installation). Pour modifier cela jouez avec les variables de la ligne val = map (val, 0,1023, 0, 179); Pour des indices sur comment faire allez sur : <http://arduino.cc/en/Reference/Map> .

PLUS, PLUS, PLUS :

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

<http://ardx.org/CIRC09>