



Département d'Informatique

Info 2 PC

Licence S2 - SI

2012-2013

TP3

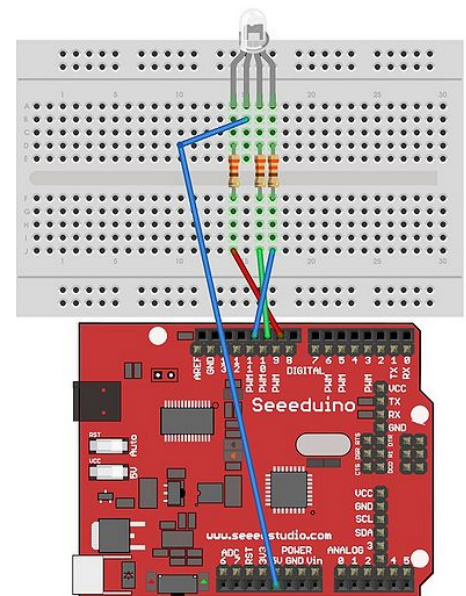
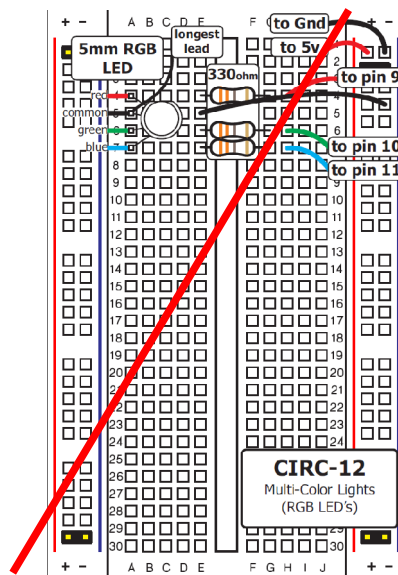
Jouons un peu plus avec Arduino

Rappel sur l'utilisation de l'environnement

Après avoir lancé un terminal (une console), il suffit de se mettre dans le répertoire `arduino-1.0.4` avec la commande `cd arduino-1.0.4` puis de taper `./arduino`

ATTENTION : RELIER LE 5V d'un ARDUINO À LA MASSE PEUT LE TUER (MÊME SI C'EST FAIT VIA LE BREADBOARD !!!)

Mais qu'elles sont LED !



ATTENTION !! La LED de notre kit est inversée par rapport au schéma à gauche et à celui de la page suivante (néanmoins les explications sont bonnes) ! La patte la plus longue est l'anode (**elle doit être connectée au 5V et non au GND**). Il faut suivre le schéma de droite !

CIRC-12

.:Lumière Colorée.: .:LEDs RGB:.



CE QUE NOUS ALLONS FAIRE:

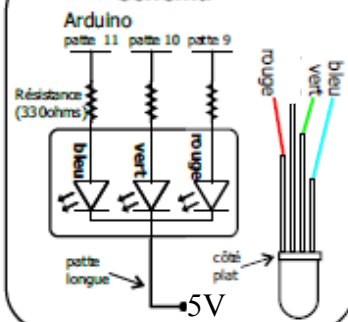
Vous avez commencé par le CIRC-01 ou vous vous êtes amusés à faire clignoter une LED rouge. Mais maintenant vous avez dépassé ça, N'est-ce pas ? Vous voulez de l'orange, du violet, de l'indigo ! Heureusement il y a un moyen simple pour obtenir plusieurs couleurs avec une seule LED. Pour cela, nous utilisons une LED RGB. Une LED RGB n'est pas une simple LED. En fait, il s'agit de 3 LEDs dans un petit boîtier : une Rouge, une Verte et une Bleue. Quand vous les allumez ensemble, vous obtenez différentes couleurs. La couleur que vous obtenez est le résultat de la somme des intensités individuelles de chaque LED. Nous contrôlons l'intensité avec les PWM (Pulse Width Modulation) que nous avons déjà utilisé pour contrôler la luminosité d'une LED ou la vitesse d'un moteur.

LE CIRCUIT:

Composants :

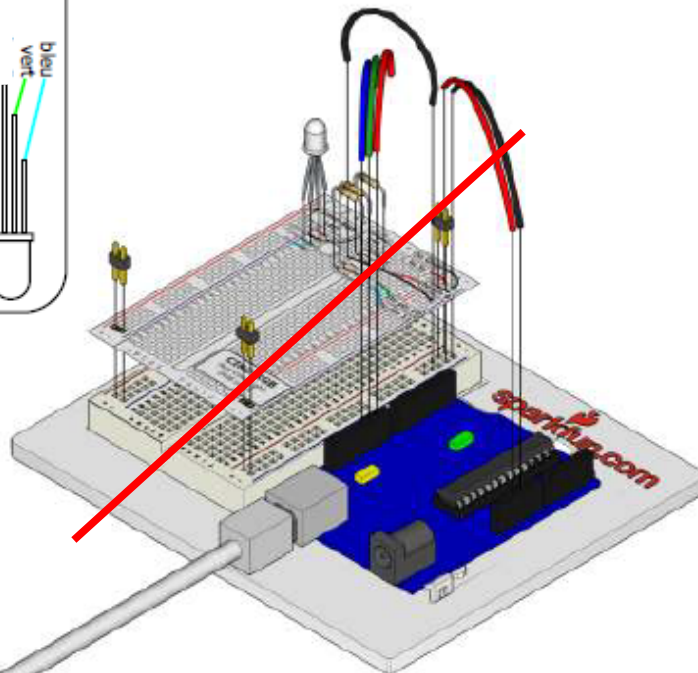
- CIRC-12 Feuille de Connexions x1
- Connecteur 2 pattes x4
- LED RGB 5mm x1
- Résistance 330 Ohms Orange-Orange-Marron x3
- Fil

Schéma



Internet

.:Télécharger.:
Feuille de Connexions
<http://arduino.org/BLS12>



Pour ne pas taper le code ci-dessous, on peut télécharger le code à l'adresse : <http://ardx.org/CODE12S> (par contre les commentaires seront en anglais).

CODE (Il n'est pas nécessaire de taper quoi que ce soit)

Téléchargez le code (<http://ardx.org/CODE12S>)

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

```
/*Faire défiler les couleurs sur la LED RGB*/
// Les pattes de la LED sont connectées sur
les pattes de PWM
const int RED_LED_PIN = 9;
const int GREEN_LED_PIN = 10;
const int BLUE_LED_PIN = 11;

// Utilisé pour stocker le niveau d'intensité
actuel
int redIntensity = 0;
int greenIntensity = 0;
int blueIntensity = 0;

// Temps d'affichage de chaque couleur
const int DISPLAY_TIME = 100; // millisecondes

void setup() {
  // Pas de setup nécessaire.
}

void loop() {
  // Cycle allant de rouge à vert
  for (greenIntensity = 0;
    greenIntensity <= 255;
    greenIntensity+=5) {
    redIntensity = 255-greenIntensity;

    analogWrite(GREEN_LED_PIN, greenIntensity);
    analogWrite(RED_LED_PIN, redIntensity);
    delay(DISPLAY_TIME);
  }

  // Cycle allant de vert à bleu
  for (blueIntensity = 0;
    blueIntensity <= 255;
    blueIntensity+=5) {
    greenIntensity = 255-blueIntensity;
    analogWrite(BLUE_LED_PIN, blueIntensity);
    analogWrite(GREEN_LED_PIN, greenIntensity);
    delay(DISPLAY_TIME);
  }

  // Cycle allant de bleu à rouge
  for (redIntensity = 0;
    redIntensity <= 255;
    redIntensity+=5) {
    blueIntensity = 255-redIntensity;
    analogWrite(RED_LED_PIN, redIntensity);
    analogWrite(BLUE_LED_PIN, blueIntensity);
    delay(DISPLAY_TIME);
  }

  ----- Plus de code en ligne -----
}
```

CIRC-12

CELA FONCTIONNE PAS ? (3 choses à essayer)

La LED ne s'allume pas ou pas de la bonne couleur

Avec 4 pattes très proches, la LED n'est pas facile à câbler. Vérifiez que chaque patte est là où elle devrait être.

Vous voyez rouge ?

La LED rouge dans la LED RGB peut être plus brillante que les deux autres. Pour avoir des couleurs plus équilibrées, utilisez une résistance plus importante. Ou modifiez le code :

```
analogWrite(RED_LED_PIN,
  redIntensity);
par
analogWrite(RED_LED_PIN,
  redIntensity/3);
```

Vous en attendez plus ?

Si vous cherchez à en faire plus pourquoi ne pas aller voir tous les montages et produits disponibles sur : <http://www.SparkFun.com>

AMÉLIORER LE MONTAGE

Utiliser les code couleur HTML

Si vous êtes familiers avec la création de pages web, vous devriez préférer le code hexadécimal comme lorsque vous utilisez HTML ou CSS. Un triplet hexadécimal décrit une couleur en utilisant une série de lettres et de nombres comme `#FF0000` pour rouge `#800080` pour violet. Vous pouvez en apprendre plus sur comment ça marche sur Wikipedia (<http://ardx.org/HEXCOL>) et trouver une liste de couleurs et leur triplet hexadécimal associé, ainsi, vous n'aurez pas à le faire vous-même.

Téléchargez le code:

<http://ardx.org/RGBMB>

Une plus grande diffusion

L'un des désavantages d'utiliser une LED RGB fabriquée à partir de trois LED séparées pour générer notre couleur est qu'il est parfois possible de voir la couleur de chaque LED. L'une des manières de contourner ceci est de trouver une façon de rendre la lumière plus diffuse (ou dispersée) afin de mieux mélanger les couleurs. La LED fournie avec votre kit diffuse mieux que les transparentes pour augmenter l'efficacité de ce mélange des couleurs. Si la lumière n'est toujours pas assez diffuse, vous pouvez essayer de placer la LED derrière du papier ou de l'acrylique ; ou à l'intérieur d'une balle de pingpong ou encore dans une boule de polystyrène.

PLUS, PLUS, PLUS :

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

<http://sparkfun.com/RGB>

31

Y a pas photo (résistance) !

Inutile de débrancher la LED RGB ! Par contre 9, 10 et 11 faisant du PWM, on pourra utiliser une des 3 pattes de la LED RGB.

CIRC-09

.:Lumière:. .:Photorésistance:.



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

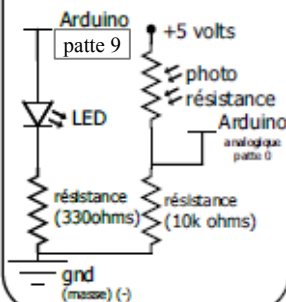
Photo-Résistance
x1

Résistance 10k Ohms
Marron-Noir-Orange
x1

Résistance 330 Ohms
Orange-Orange-Marron
x1

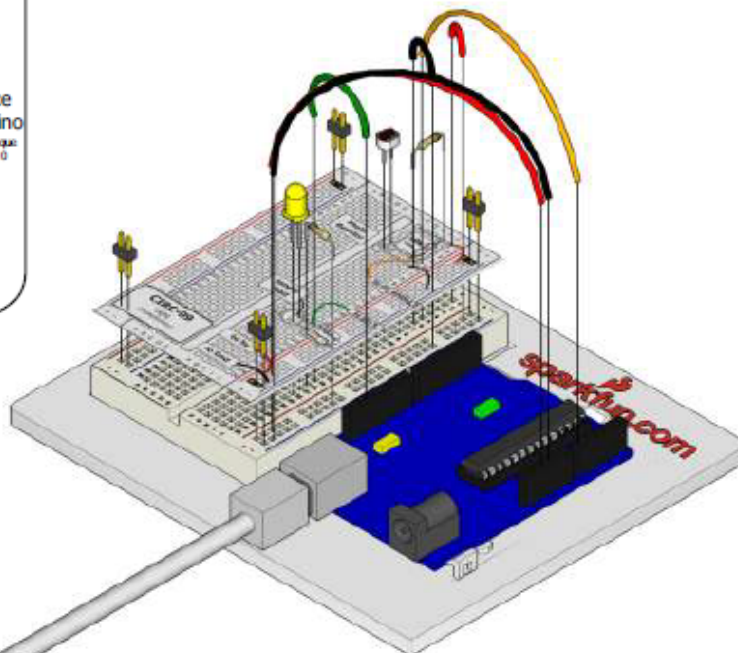
LED Jaune
x1

Schéma

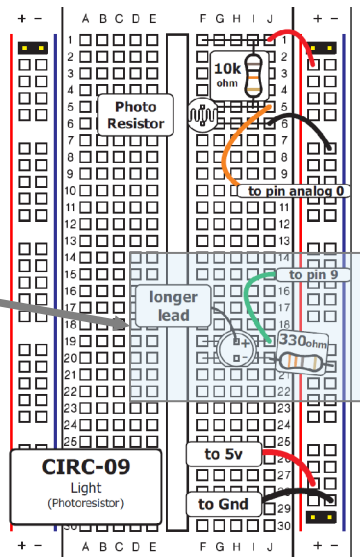


Internet

.:Télécharger:.
Feuille de Connexions
<http://ardx.org/B8LS09S>
.:Voir:.
Vidéo de Montage
<http://ardx.org/VIDE09>



Notre LED RGB
(donc inutile de
changer le
circuit)



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 dû 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>

Et si on se parlait petit Arduino ?

NE RIEN DEBRANCHER ! Pour ne pas taper le code ci-dessous, on peut ouvrir un exemple qui le reprend dans « Fichiers », « Exemples », « 3. Communication », « Dimmer » (par contre les commentaires seront en anglais). En fonction de la valeur tapée, la LED sur la broche défini s'ajustera :

```
const int ledPin = 9;          // the pin that the LED is attached to

void setup()
{
  // initialize the serial communication:
  Serial.begin(9600);
  // initialize the ledPin as an output:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  byte brightness;

  // check if data has been sent from the computer:
  if (Serial.available()) {
    // read the most recent byte (which will be from 0 to 255):
    brightness = Serial.parseInt();
    // set the brightness of the LED:
    analogWrite(ledPin, brightness);
    delay(2000);
  }
}
```

Essayons avec ce programme pour lequel on tape : « a » ou « e »

```
const int ledPin = 9;          // the pin that the LED is attached to

void setup()
{
  // initialize the serial communication:
  Serial.begin(9600);
  // initialize the ledPin as an output:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  char action;

  // check if data has been sent from the computer:
  if (Serial.available()) {
    // read the most recent byte (which will be from 0 to 255):
    action = (char)Serial.read();
    switch(action){
      case 'e': digitalWrite(ledPIN, HIGH); break; // OFF
      case 'a': digitalWrite(ledPIN, LOW); break;  // ON
    }
  }
}
```

Notre Arduino sait donc lire ! On lui fait écrire ?

```
const int ledPin = 13;          // the pin that the LED is attached to

void setup()
{
  // initialize the serial communication:
  Serial.begin(9600);
  // initialize the ledPin as an output:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  char action;

  // check if data has been sent from the computer:
  if (Serial.available()) {
    // read the most recent byte (which will be from 0 to 255):
    action = (char)Serial.read();
    switch(action){
      case 'e': digitalWrite(ledPIN, HIGH); Serial.println("OFF"); break; // OFF
      case 'a': digitalWrite(ledPIN, LOW); Serial.println("ON"); break;   // ON
    }
  }
}
```