



CE QUE NOUS ALLONS FAIRE:

Un potentiomètre tactile (soft pot) est comme un potentiomètre standard comme vu dans le CIRC-08 excepté le fait qu'il est très fin, flexible et qu'il n'a pas de bouton rotatif. Les potentiomètres sont aussi connus comme « résistances variables ». Dans le cas d'un potentiomètre tactile, la résistance est déterminée par l'endroit où une pression est appliquée. La pression peut être appliquée par un doigt, un stylet ou un morceau de plastique (wiper). En appuyant à différents endroits de la bande, la résistance varie de 100 à 10k Ohms ce qui permet de déterminer la position sur la bande. Vous pouvez l'utiliser pour suivre un mouvement. Dans ce circuit nous allons l'utiliser pour contrôler la couleur d'une LED RGB.

LE CIRCUIT:

Composants :



CIRC-14
Feuille de Connexions
x1



Connecteur 2 pattes
x4



LED RGB 5mm
x1



Fil

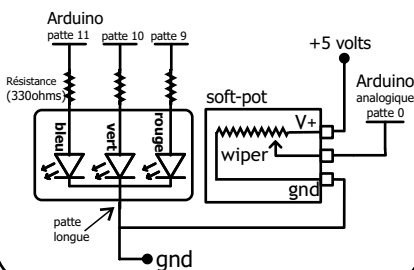


Potentiomètre tactile
x1



Résistance 330 Ohms
Orange-Orange-Marron
x3

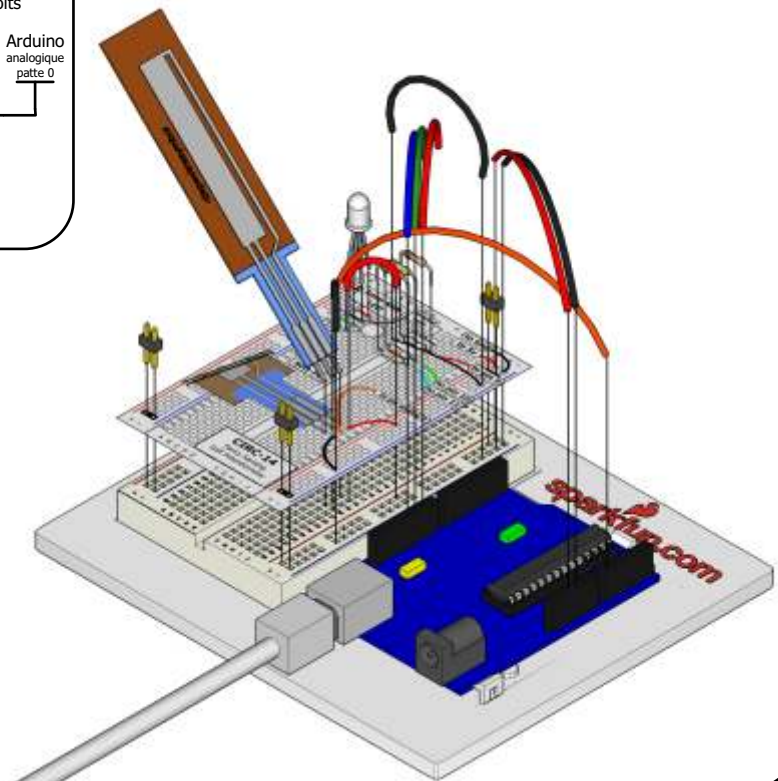
Schéma



Internet

::Télécharger::

Feuille de Connexions
<http://ardx.org/BBL14S>



CODE (Il n'est pas nécessaire de taper quoi que ce soit)**Téléchargez le code (<http://ardx.org/CODE14S>)**

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

```
// Code d'Exemple CIRC-14
// Balaye les couleurs d'une LED RGB
// entre Rouge Vert et Bleu
// en fonction de la valeur du
// potentiomètre tactile
// pattes de la LED connectées aux pattes
// de PWM
const int RED_LED_PIN = 9;
//patte Rouge de la LED
const int GREEN_LED_PIN = 10;
//patte Verte de la LED
const int BLUE_LED_PIN = 11;
//patte Bleue de la LED

void setup() {
  //pas besoin de code ici
}

void loop() {
  int sensorValue = analogRead(0);
  //lit le potentiomètre tactile
  int redValue = constrain(
    map(sensorValue, 0, 512, 255,
        0,0,255);
    //calcule la valeur de rouge (255-0
    //dans la plage 0-512)
    int greenValue = constrain(
      map(sensorValue, 0, 512, 0, 255),0,255)-
      constrain(
        map(sensorValue, 512, 1023, 0,
            255),0,255);
    // calcule la valeur de vert (0-255
    //sur la plage 0-512 et 255-0 sur la plage
    //512-1023)
    int blueValue = constrain(
      map(sensorValue, 512, 1023, 0,
          255),0,255);
    //calcule la valeur de bleu 0-255 sur
    //la plage 512-1023
    //affiche la couleurs voulue
    analogWrite(RED_LED_PIN, redValue);
    analogWrite(GREEN_LED_PIN, greenValue);
    analogWrite(BLUE_LED_PIN, blueValue);
  }
```

CELA FONCTIONNE PAS ? (3 choses à essayer)**La LED ne s'allume pas ou affiche des couleurs incorrectes**

Les pattes des LED sont très proches, il est facile de les placer au mauvais endroit. Essayez en vérifiant une fois de plus que les pattes sont à l'endroit où elles devraient 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);
```

Résultats bizarres

La cause la plus courante est le fait que vous appuyez sur le potentiomètre en plusieurs points. C'est normal et cela peut aussi être utilisé pour obtenir de meilleurs résultats.

AMÉLIORER LE MONTAGE**Couleur HSB (Teinte, Saturation & Luminosité)**

Notre LED RGB affiche des couleurs en utilisant les codes couleur RGB. Cependant, ce n'est pas toujours la manière la plus facile de travailler avec des couleurs (comme le fait remarquer le code pas facile à lire ci-dessus). Un système bien plus intuitif est l'HSB.

Détails complets :

<http://ardx.org/HSB>

Pour passer de RGB à HSB, tout ce qui est nécessaire est un calcul un petit peu compliqué. Pour un exemple de programme, visitez :

<http://ardx.org/CODE14MB>

basé sur le code original www.kasperkamperman.com :

<http://ardx.org/KASP>

Maintenant, quand vous utilisez le potentiomètre souple, vous remarquerez un fondu plus propre et plus complet du rouge au violet.

Faux Boutons

À cause de la manière dont le potentiomètre souple fonctionne, il est aussi possible de l'utiliser pour faire des boutons customisés. Pour cela définissez une plage de valeurs correspondant à un bouton.

Utilisez le fragment de code ci-après et la fenêtre de débogage pour déterminer les valeurs voulues.

```
if(analogRead(0) > minValue &&
  analogRead(0) < maxValue){
  boutonAction()
}
```

Ensuite, recouvrez le potentiomètre souple avec un dessin de bouton imprimé ou dessiné

PLUS, PLUS, PLUS :

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

<http://sparkfun.com/>