

Table of Contents

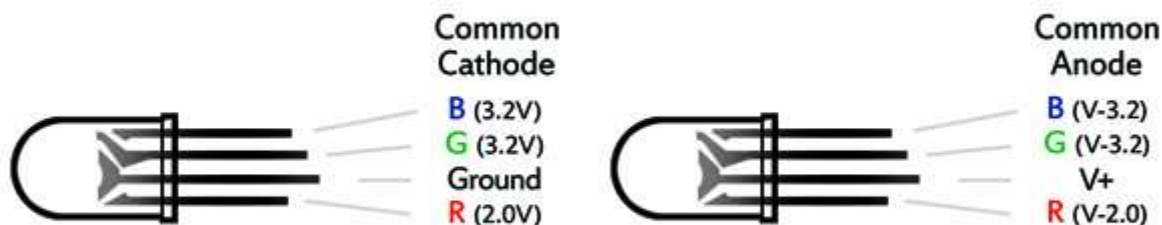
Contrôler une LED tricolore RGB depuis les GPIO	1
Les DEL RGB Common cathode (à cathode commune)	2
Les DEL RGB Common Anode (à anode commune)	4

Contrôler une LED tricolore RGB depuis les GPIO

Les DEL RGB sont des DEL contenant en fait 3 DEL : une rouge (R), une verte (G), et une bleue (B). Le format est celui d'une DEL classique, sauf qu'il y a 4 pattes. On peut allumer l'une des 3 couleurs, pour avoir du rouge, du vert ou du bleu, mais on peut également allumer des combinaisons de deux couleurs (Rouge + Vert = jaune, par ex), ou même les trois couleurs en même temps pour faire du blanc. Enfin, si vous contrôlez le niveau de chaque led avec de la PWM, vous pouvez faire varier l'intensité de chaque couleur. Imaginons que vous puissiez faire varier en pwm chaque patte de la led de 0 à 255 : en pratique, vous obtenez des nuances de couleurs comme dans un logiciel d'image, avec 256 nuances de chaque couleur de base, et une combinaison de $256 \times 256 \times 256 =$ environ 16 millions de couleurs! Aujourd'hui, nous ne nous occuperons pas de PWM, car le PI ne dispose que d'une broche PWM. On se contentera d'allumer chaque couleur, et de combiner les couleurs de façon simple.

En premier lieu, vous trouverez divers types de LED RGB. Il y en aura des "clear", dont le boîtier epoxy est transparent. La luminosité de celles-ci est maximale devant la LED, mais plus faible sur le côté. Elles sont souvent éblouissantes vues de face. Il y aura aussi les "diffused", dont le boîtier est translucide, vous ne verrez pas l'intérieur clairement. Celles-ci diffusent davantage la lumière, et sont mieux visibles dans tous les sens. L'autre point important, c'est la distinction entre les *common cathode* (cathode commune) et les *common anode* (anode commune). Les premières, les common cathode ont une patte pour le + de chaque couleur, et une patte commune pour le - de toutes les couleurs. Pour les *common anode*, c'est le contraire, la patte commune est l'anode, donc on a une patte commune pour le + (l'anode), et une patte pour le - de chaque couleur. Cela signifie que le câblage ne sera pas le même, et la logique de contrôle sera également différente. Si vous avez le choix, les *common cathode* sont plus simples, bien que les *common anode* ne soient pas particulièrement complexes.

Dans les deux cas, voici la disposition des pattes de la LED :



En pratique, voici ce que peut donner une LED RGB:



Variations de couleur d'une LED RGB contrôlée par un Raspberry Pi

Les DEL RGB Common cathode (à cathode commune)

Ici, c'est donc le - qui est commun. Le câblage est donc très simple : vous branchez la cathode (patte 2), qui est la plus grande patte, sur le -, vous reliez les pattes 1 et 4 à des GPIO, et la patte 3 à une résistance qui va dans un GPIO. Il suffit alors de définir le GPIO à high pour allumer une couleur, et à low pour l'éteindre. Idéalement, la résistance en série pour la patte 3 (rouge) devrait faire 65 Ohms.

Voici [un schéma de câblage pour Arduino](#), avant de passer à la version Raspberry Pi :) Le schéma de câblage sera donc le suivant, [en utilisant un MCP23017 pour profiter de ses GPIO](#) :

-très bientôt- Quand au code, il est relativement simple :

[colorChangeCA.py](#)

```
#!/usr/bin/python
from Adafruit_I2C import Adafruit_I2C
from Adafruit_MCP230xx import Adafruit_MCP230XX
import smbus
import time

def lightRGBLed(rState,gState,bState,rPin,gPin,bPin):
    mcp.output(rPin, rState)
    mcp.output(gPin, gState)
    mcp.output(bPin, bState)
```

```
def lightRGBLed2(color,rPin,gPin,bPin):
    if color == 'red':
        lightRGBLed(0,1,1,rPin,gPin,bPin)
    elif color == 'green':
        lightRGBLed(1,0,1,rPin,gPin,bPin)
    elif color == 'blue':
        lightRGBLed(1,1,0,rPin,gPin,bPin)
    elif color == 'yellow':
        lightRGBLed(0,0,1,rPin,gPin,bPin)
    elif color == 'magenta':
        lightRGBLed(0,1,0,rPin,gPin,bPin)
    elif color == 'cyan':
        lightRGBLed(1,0,0,rPin,gPin,bPin)
    elif color == 'white':
        lightRGBLed(0,0,0,rPin,gPin,bPin)

if __name__ == '__main__':
    mcp = Adafruit_MCP230XX(busnum = 1, address = 0x20, num_gpios = 16)
    r1=7
    g1=6
    b1=5
    mcp.config(b1, mcp.OUTPUT)
    mcp.config(g1, mcp.OUTPUT)
    mcp.config(r1, mcp.OUTPUT)

    while (True):
        print "off"
        lightRGBLed(1,1,1,r1,g1,b1)
        time.sleep(1)

        print "red"
        lightRGBLed(0,1,1,r1,g1,b1)
        time.sleep(1)

        print "green"
        lightRGBLed(1,0,1,r1,g1,b1)
        time.sleep(1)

        print "blue"
        lightRGBLed(1,1,0,r1,g1,b1)
        time.sleep(1)

        print "green+blue"
        lightRGBLed(1,0,0,r1,g1,b1)
        time.sleep(1)

        print "red+blue"
        lightRGBLed(0,1,0,r1,g1,b1)
        time.sleep(1)

        print "red+green"
```

```
lightRGBLed(0,0,1,r1,g1,b1)
time.sleep(1)

print "red+green+blue"
lightRGBLed(0,0,0,r1,g1,b1)
time.sleep(1)
```

Les DEL RGB Common Anode (à anode commune)

Ici, c'est donc le + qui est commun. Le câblage sera donc différent : on connecte toujours chaque couleur à un GPIO, et toujours une résistance en série pour le rouge. En revanche, la broche commune, l'anode, sera branchée sur le +3.3v. La différence c'est qu'ici, pour allumer une couleur, il faudra mettre le GPIO correspondant à 0. Si le GPIO est à 0, il est à 0V. Donc, le courant entre par l'anode à +3.3V, cela fait une tension de 3.3V au bornes de la DEL. En revanche, si on met le GPIO à 1, il sera à 3.3V. Ce qui fait qu'à chaque borne de la DEL, on a du +3.3V. du coup la tension aux bornes de celle ci est de 0V, et elle reste éteinte! Le code est donc inversé, puisqu'il faut passer le GPIO à 0 pour allumer, et à 1 pour éteindre.

From:
<http://nagashur.com/wiki/> - **nagashur**

Permanent link:
http://nagashur.com/wiki/doku.php?id=raspberry_pi:controle_led_rgb

Last update: **22/02/2015 05:16**

