

# Manipulation de la matrice de LED

Le coeur de la librairie de la matrice se situe dans un fichier appelé `rgbmatrix.so` (non disponible actuellement étant donné que la compilation a l'air d'être très dépendante du raspberry pi)

La matrice peut être commandé selon deux modes :

- Un mode instantané ou chaque appel de commande affiche un résultat sur la matrice de manière immédiate. On dispose dans ce mode des méthodes `Fill(couleur)` pour remplir la matrice en entière (couleur rgb en hexadécimal), et `SetPixel (couleur)` pour définir la couleur d'un pixel.
- Un mode de dessin ou l'on commence par préparer une Image python pour ensuite l'envoyer à la matrice.

Pour ce faire on crée l'image souhaitée avec python( je vous renvoie aux deux pdf à coté pour la librairie Image) puis on appelle la méthode `SetImage(id de l'image, coordonnées)` pour charger l'image dans la matrice

Voici les deux exemples de la librairie pour s'inspirer un peu :

```
1 : import time
from rgbmatrix import Adafruit_RGBmatrix
# Rows and chain length are both required parameters:
matrix = Adafruit_RGBmatrix(32, 1)

# Flash screen red, green, blue (packed color values)
matrix.Fill(0xFF0000)
time.sleep(1.0)
matrix.Fill(0x00FF00)
time.sleep(1.0)
matrix.Fill(0x0000FF)
time.sleep(1.0)
# Show RGB test pattern (separate R, G, B color values)
for b in range(16):
    for g in range(8):
        for r in range(8):
            matrix.SetPixel(
                (b / 4) * 8 + g,
                (b & 3) * 8 + r,
                (r * 0b001001001) / 2,
                (g * 0b001001001) / 2,
                b * 0b00010001)
```

```
time.sleep(10.0)
matrix.Clear()
```

2 :

```
import Image
import ImageDraw
import time
from rgbmatrix import Adafruit_RGBmatrix
```

```
# Rows and chain length are both required parameters:
matrix = Adafruit_RGBmatrix(32, 1)
```

```
# Bitmap example w/graphics prims
image = Image.new("1", (32, 32)) # Can be larger than matrix if wanted!!
draw = ImageDraw.Draw(image)    # Declare Draw instance before prims
```

```
# Draw some shapes into image (no immediate effect on matrix)...
draw.rectangle((0, 0, 31, 31), fill=0, outline=1)
draw.line((0, 0, 31, 31), fill=1)
draw.line((0, 31, 31, 0), fill=1)
```

```
# Then scroll image across matrix...
```

```
for n in range(-32, 33): # Start off top-left, move off bottom-right
    matrix.Clear()
    # IMPORTANT: *MUST* pass image ID, *NOT* image object!
    matrix.SetImage(image.im.id, n, n)
    time.sleep(0.05)
```

```
# 8-bit paletted GIF scrolling example
image = Image.open("cloud.gif")
image.load()          # Must do this before SetImage() calls
matrix.Fill(0x6F85FF) # Fill screen to sky color
```

```
for n in range(32, -image.size[0], -1): # Scroll R to L
    matrix.SetImage(image.im.id, n, 0)
    time.sleep(0.025)
```

```
# 24-bit RGB scrolling example.
# The adafruit.png image has a couple columns of black pixels at
# the right edge, so erasing after the scrolled image isn't necessary.
matrix.Clear()
image = Image.open("adafruit.png")
image.load()
```

```
    for n in range(32, -image.size[0], -1):  
        matrix.SetImage(image.im.id, ,1)  
        time.sleep(0.025)  
  
matrix.Clear()
```