python:neopixel

Ce programme contrôle un bandeau de LED **NeoPixel** (probablement une bande de LEDs adressables individuellement) à l'aide de la bibliothèque `neopixel`. Il utilise une boucle infinie pour alterner les couleurs sur les pixels de la bande, créant un effet visuel animé. Voici une explication détaillée de chaque partie du programme :

1.

```
import time
from neopixel import Neopixel
```

- `time` : Ce module permet de gérer des temporisations (pauses) dans le programme.
- `Neopixel` : Il s'agit d'une bibliothèque qui permet de contrôler des bandes de LED NeoPixel. Dans ce cas, la classe `Neopixel` est utilisée pour définir et manipuler la bande de LEDs.

2.

```
pixels = Neopixel(17, 0, 6, "GRB")
```

- Cette ligne crée un objet `pixels` de type `Neopixel`.
  - 17 : Nombre de LEDs dans la bande (ici, 17 LEDs).
  - `0` et 6 : Indices des broches (GPIO) du microcontrôleur auxquelles les données et l'alimentation sont connectées (cela dépend de votre configuration matérielle).
  - `"GRB"` : Cela spécifie l'ordre des couleurs des LEDs. Ici, "GRB" signifie que chaque LED utilise les valeurs de couleur dans l'ordre Vert (Green), Rouge (Red), Bleu (Blue). Il existe aussi d'autres formats comme "RGB", "BRG", etc.

3.

```
colors = [
  (0xb6, 0xe4, 0x30),
  (0x42, 0xd1, 0xe0),
]
```

- Un tableau `colors` est créé avec deux couleurs définies sous forme de tuples de valeurs hexadécimales :
  - `(0xb6, 0xe4, 0x30)` : Cette couleur est un vert-jaune.
  - `(0x42, 0xd1, 0xe0)` : Cette couleur est un bleu clair.
- Les valeurs hexadécimales représentent les intensités de lumière pour chaque couleur : rouge (R), vert (G), et bleu (B).

4.

```
pixel_index = 0
color_index = 0
```

- `pixel_index` garde la trace de l'index du pixel actuel qui doit changer de couleur.
- `color_index` détermine quelle couleur de la liste `colors` doit être utilisée (0 ou 1).

5.

```
while True:
  pixels.set_pixel(pixel_index, colors[color_index])
  pixels.show()
  pixel_index += 1
  if pixel_index == 16:
    pixel_index = 0
    color_index = (color_index + 1) % 2
  time.sleep(0.1)
```

- `while True` crée une boucle infinie qui continuera à s'exécuter indéfiniment.
- À chaque itération de la boucle :
  - `pixels.set_pixel(pixel_index, colors[color_index])` : La couleur définie pour l'index `pixel_index` est appliquée à ce pixel.
  - `pixels.show()` : Cette fonction applique toutes les modifications aux pixels de la bande de LEDs.
  - `pixel_index += 1` : L'index du pixel est incrémenté, ce qui signifie que la couleur sera appliquée au pixel suivant.
  - Si `pixel_index` atteint 16 (le dernier pixel de l'index, car les indices vont de 0 à 16 pour 17 LEDs), on réinitialise `pixel_index` à 0 et on change la couleur en modifiant `color_index`. L'incrémentation `(color_index + 1) % 2` alterne entre 0 et 1, ce qui permet de passer de la première couleur à la deuxième et vice versa.
  - `time.sleep(0.1)` : Cette fonction met en pause le programme pendant 0.1 seconde entre chaque itération, ce qui ralentit l'effet pour qu'il soit visible à l'œil nu. Vous pouvez augmenter cette valeur pour un effet plus lent, ou la réduire pour un effet plus rapide.

6.

- La boucle alterne les couleurs de la bande LED, en assignant une couleur à chaque pixel successivement.
- Chaque pixel passe à la couleur suivante, puis le programme passe à la couleur suivante dans la liste `colors` après avoir atteint le dernier pixel.
- Le processus recommence indéfiniment, créant un effet de balayage de couleur qui alterne entre les deux couleurs définies.

Résumé :

Le programme contrôle une bande de 17 LEDs NeoPixel, où chaque LED passe alternativement entre deux couleurs définies dans la liste `colors`. La couleur des LEDs change progressivement en avançant de pixel en pixel, et chaque changement de couleur est visible à intervalle de 0.1 seconde. Lorsque le dernier pixel est atteint, le programme recommence avec le premier pixel et change la couleur.

Ce programme pourrait être utilisé, par exemple, pour créer des effets visuels ou des animations sur une bande de LEDs adressables.