



Circuit Playground Express

Systèmes embarqués



Objectif :

*Circuit Playground Express
Micro Python*

Découvrir quelques aspects de la mise en œuvre de systèmes embarqués autour d'une carte peu onéreuse aux multiples possibilités la Playground Express d'Adafruit associée au langage micro python.

La carte est pilotée grâce à un microcontrôleur 32 bits ATMEL ATSAMD21, doté d'un cœur ARM® Cortex®-M0+ processor, opérant à une fréquence de 48 MHz.



SMART ARM-Based Microcontroller

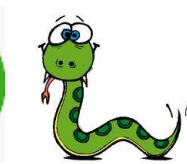
(Atmel est une branche de Microchip depuis son rachat en 2012 mais a conservé ses designs de circuits compacts et performants.)



Sommaire

- 1 **TP_2 : Les rubans de leds Neopixels** 2
 - 1.1 *Objectif* :.....2
 - 1.2 *Les différentes Neopixels*2
 - 1.3 *Neopixels Quesaco ?*2
 - 1.4 *Exemple de trame de commande*.....3
- 2 **Les Neopixels et la carte Playground express** 4
 - 2.1 *Le matériel*4
 - 2.2 *La gestion de la brillance*.....5
 - 2.3 *Combien de leds sur une seule sortie ?*.....5
- 3 **Mise en œuvre avec la carte Playground Express**..... 5
 - 3.1 *Un premier programme à tester*5
 - 3.2 *Expérimentations*.....6





1 TP_2 : Les rubans de leds Neopixels

Circuit Playground Express
Micro Python



1.1 Objectif :

Les Neopixels sont des leds RGB que l'on peut qualifier d'intelligentes. En effet comme vous allez pouvoir le découvrir dans cette activité elles sont faciles à commander et permettent des effets lumineux variés et agiles.

Quel est le principe de cette mise en série d'un grand nombre de leds dans des applications toujours plus lumineuses !

Comment les piloter et obtenir les effets lumineux désirés ?

Il est temps de les découvrir.



1.2 Les différentes Neopixels

Les rubans de Neopixels peuvent avoir plusieurs formes. En voici quelques-unes :

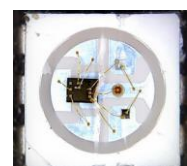


<https://wiki.mchobby.be/index.php?title=NeoPixel-UserGuide>

1.3 Neopixels Quesaco ?

Une Neopixel est une led tri couleur pilotée par son propre circuit interne. Le circuit assure deux fonctions essentielles :

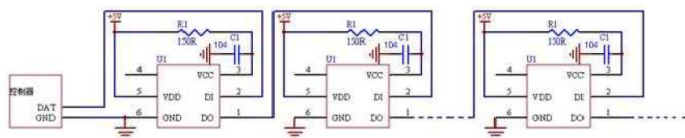
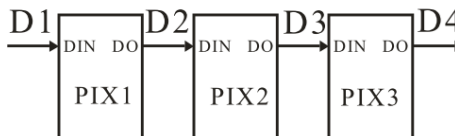
- Analyser la trame de commande et en extraire les trois premières valeurs RGB
- Piloter l'allumage de chaque LED en fonction des valeurs RGB récupérées.



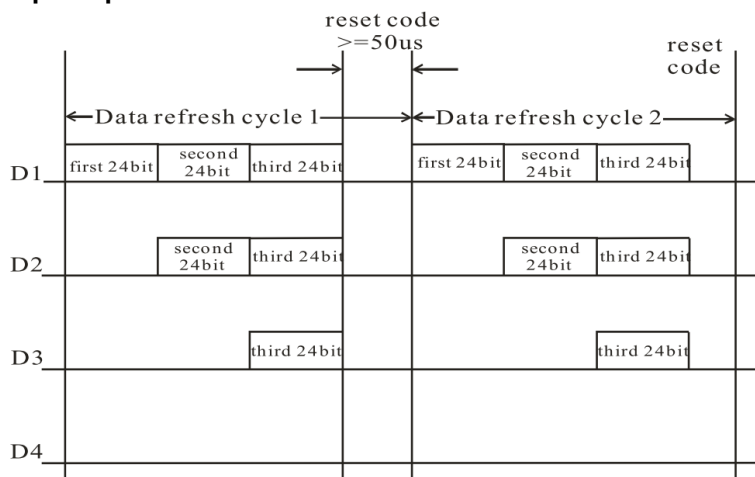


Le principe du câblage est indiqué ci-dessous, chaque led reçoit la trame sur son entrée IN et la restitue, amputée des trois premiers octets reçus, sur sa sortie OUT. Ainsi chacune des leds ignore le nombre de ses successeurs. Le processus est remis à zéro après un reset supérieur à 50 µs.

Le câblage en série des Neopixels



Le principe de la trame de commande

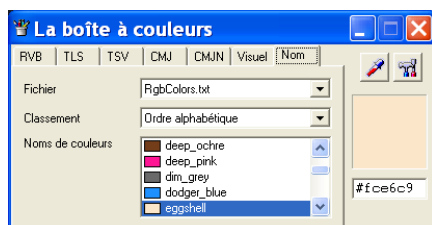


1.4 Exemple de trame de commande

Chaque composante de couleurs pour une Neopixel est codée sur 8 bits.

Q1. Combien de couleurs sont possibles avec une Neopixel ?

Voilà trois exemples de couleurs obtenues avec l'utilitaire La boîte à couleurs¹



¹ <https://www.netfox2.net/modules/wfdownloads/singlefile.php?cid=104&lid=751>



Q2. Encoder les trames ci-dessous avec les couleurs correspondantes.



G	7	G	6	G	5	G	4	G	3	G	2	G	1	G	0

Couleur Deep_pink

R	7	R	6	R	5	R	4	R	3	R	2	R	1	R	0

B	7	B	6	B	5	B	4	B	3	B	2	B	1	B	0

G	7	G	6	G	5	G	4	G	3	G	2	G	1	G	0

Couleur eggshell

R	7	R	6	R	5	R	4	R	3	R	2	R	1	R	0

B	7	B	6	B	5	B	4	B	3	B	2	B	1	B	0

G	7	G	6	G	5	G	4	G	3	G	2	G	1	G	0

Couleur eggshell

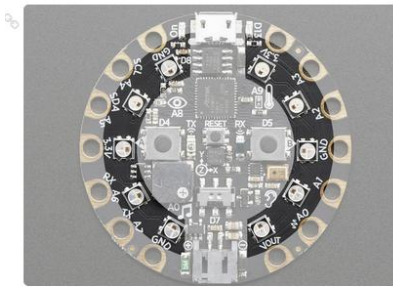
R	7	R	6	R	5	R	4	R	3	R	2	R	1	R	0

B	7	B	6	B	5	B	4	B	3	B	2	B	1	B	0

2 Les Neopixels et la carte Playground express

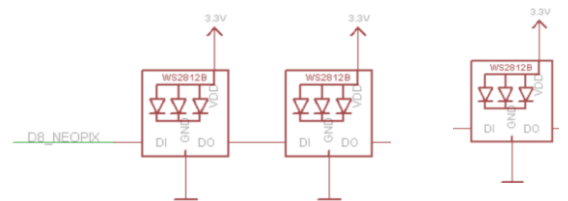
2.1 Le matériel

Une vue de l'anneau circulaire de la carte CPX



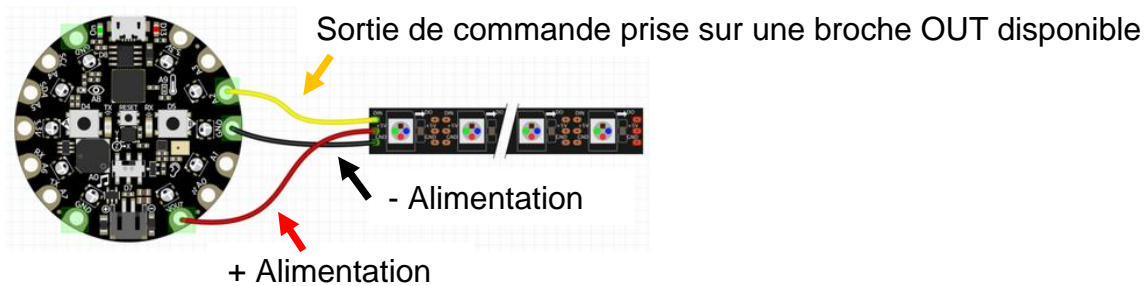
10 x Color NeoPixel LED
The ten LEDs surrounding the outer edge of the boards are all full color, RGB LEDs, each one can be set to any color in the rainbow. Great for beautiful lighting effects! The NeoPixels will also help you know when the bootloader is running (they will turn green) or if it failed to initialize USB when connected to a computer (they will turn red).

Le schéma correspondant



Q3. Combien de broches d'entrée - sortie sont-elles mobilisées pour la commande d'un ruban de leds Neopixels ?

On peut ajouter d'autres groupes de Neopixels en utilisant des broches supplémentaires. Voilà un exemple de câblage :





2.2 La gestion de la brillance²

Les fonctions implémentées dans l'interface de commande sous micro python permettent de gérer la brillance à condition de commander un bandeau de moins de 300 Neopixels !!

You can drive 300 NeoPixel LEDs with brightness control (set `brightness=1.0` in object creation) and 1000 LEDs without. That's because to adjust the brightness we have to dynamically recreate the data-stream each write.

La limitation à 300 leds se justifie par le fait que la brillance est créée en complétant dynamiquement le flux de commande pour obtenir l'effet de brillance désiré. Ce pilotage est automatique.

2.3 Combien de leds sur une seule sortie ?

L'alimentation 3.3V d'une carte CPX peut piloter, driver, 500 mA donc environ 50 leds en usage moyen. Au-delà il est nécessaire de recourir à un composant d'interface de type transistor.

3 Mise en œuvre avec la carte Playground Express

3.1 Un premier programme à tester

Voilà un script microPython qui utilise les Neopixels. Nous pouvons observer que les Neopixels sont vus dans un tableau de 10 valeurs. Chacune est un triplet, tuple en python, composé des valeurs RVB ou RGB désirée.

Les bibliothèques utiles

```
import time
import digitalio
import neopixel
from board import NEOPIXEL, BUTTON_A, BUTTON_B
```

L'instanciation, la création, du tableau permettant l'accès aux Neopixels

```
# Déclaration matérielle
#   Instanciation d'un objet NeoPixel
#   Attribution à la sortie NEOPIXEL de la carte
# L'instance s'appelle np et contient 10 leds de np[0] à np[9]
# Pour commander une led il suffit de lui envoyer un triplé de valeurs RGB
# np[x] = (valeur_rouge, valeur_vert, valeur_bleu)
# Exemple np[2] = (0, 0, 0xFF)
# Paramètres :
#   auto_write=False Les valeurs des neopixels ne sont modifiées
#                       uniquement que toutes ensembles par
#                       np.write()
#   auto_write=True  Changement de valeurs des neopixels pris en compte
#                       immédiatement
#   brightness=0.2   Réglage de la brillance

# Initialisation
np[0] = (0, 0, 0)
np[1] = (0, 0, 0)
np[2] = (0, 0, 0)
np[3] = (0, 0, 0)
np[4] = (0, 0, 0)
np[5] = (0, 0, 0)
np[6] = (0, 0, 0)
np[7] = (0, 0, 0)
np[8] = (0, 0, 0)
np[9] = (0, 0, 0)


# Répéter en permanence
while True:
    # Mise à jour des valeurs
    np[0] = (255, 0, 0)
    np[1] = (0, 255, 0)
    np[2] = (0, 0, 0xFF)
    # Envoi sur les neopixels
    np.write()
    # Temporisation
    time.sleep(2)

    np[0] = (0, 0, 0)
    np[1] = (0, 0, 0)
    np[2] = (0, 0, 0)
    np.write()
    time.sleep(2)

np = neopixel.NeoPixel(NEOPIXEL, 10, brightness=0.2, auto_write=False)
```

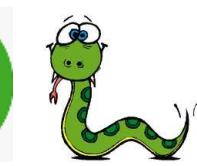
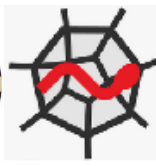
Un exemple de commande

Le code :  CPX_TP2_depart neopixel.py

 Q4. Avant de l'essayer, que fait ce programme ?



² <https://learn.adafruit.com/circuitpython-essentials/circuitpython-neopixel>



3.2 Expérimentations



Script1. Charger et tester le programme  CPX_TP2_depart neopixel.py

Script2. Charger et tester le programme  CPX_TP2_depart chenillard.py

Script3. Modifier le script précédent et soyez créatif !!

