

L'objectif de ce TP est de faire votre première programmation de la carte Bluefruit.

Nous utiliserons le logiciel **Thonny** pour taper notre code.

Branchez votre carte via le port USB de l'ordinateur.

Pour configurer Thonny :

Dans Outils puis options, choisissez interpréteur. Sélectionnez l'interpréteur MicroPython générique puis sélectionnez le Circuit Playground.

Après avoir tapé votre code sur Thonny, vous allez le sauvegarder en le nommant obligatoirement **code.py** sur la mémoire flash de la carte CIRCUITPY.

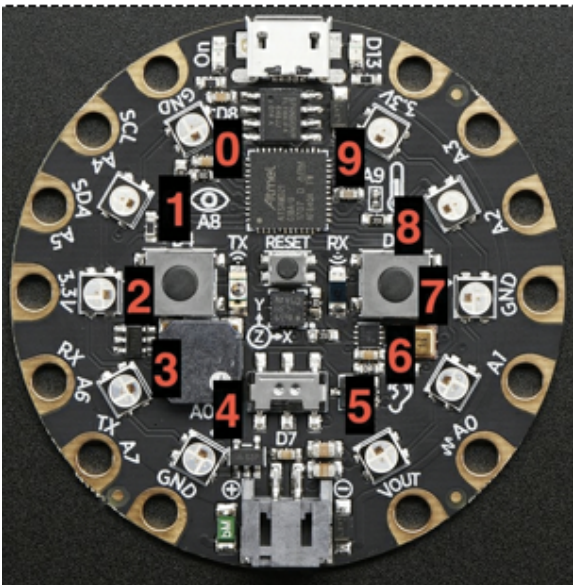
Ne débranchez pas la carte pour tester le code ;), il faut l'alimenter...

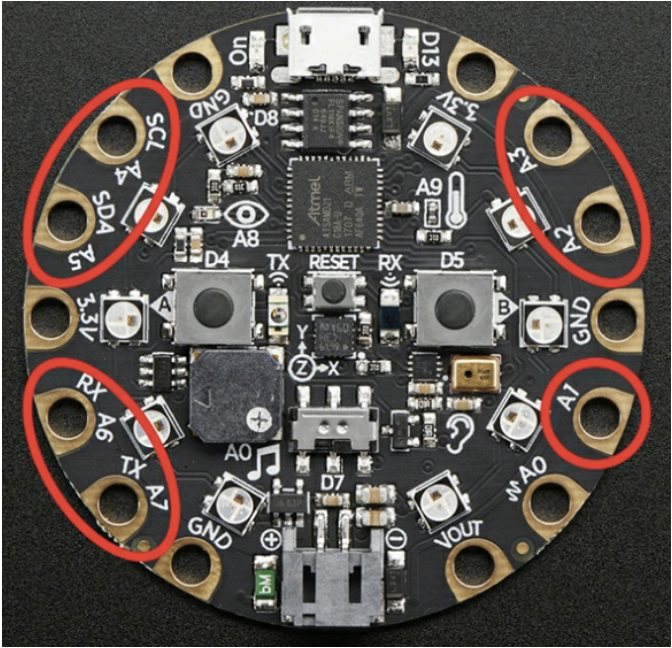
Voici quelques exemples de codes pour cette carte que vous allez tester.

Pensez à importer la bibliothèque au début de votre code:

```
from adafruit_circuitplayground import cp
```

Action	CP library usage
Accéléromètre	<pre>while True: x, y, z = cp.acceleration print((x, y, z))</pre>
Bouton A / B	<pre>while True: if cp.button_a: print("Bouton A pressé!") if cp.button_b: print("Bouton B pressé!")</pre>
Slide Switch	<pre>print("Slide switch:", cp.switch) (True à gauche, False à droite)</pre>
LED rouge D13	<pre>import time while True: cp.red_led = True time.sleep(1) cp.red_led = False time.sleep(1) (True allume la LED D13, False l'éteint)</pre>
Capteur de température	<pre>temperature_c = cp.temperature temperature_f = temperature_c * 1.8 + 32 print("Temperature celsius:", temperature_c) print("Temperature fahrenheit:", temperature_f) ou plus élaboré: from adafruit_circuitplayground import cp import time while True: temperature_c = cp.temperature temperature_f = temperature_c * 1.8 + 32 print("Temperature celsius:", temperature_c) print("Temperature fahrenheit:", temperature_f) time.sleep(1)</pre>
Capteur de lumière	<pre>import time while True: print("luminosité:", cp.light) time.sleep(1)</pre>
Jouer un fichier WAV (son)	<pre>cp.play_file("son.wav")</pre>

	Le fichier WAV doit être stocké sur la mémoire flash !
Jouer un son sur le haut-parleur avec une durée fixée	Paramètres: <ul style="list-style-type: none"> ○ fréquence (entier) - Fréquence du son en Hz ○ durée (décimal) - Durée du son en secondes cp.play_tone(440, 1.0) # 440 hz (La) pour 1 seconde
Jouer un son sur le hp jusqu'à ce que vous demandiez de l'arrêter (voir en-dessous)	Paramètre: <ul style="list-style-type: none"> ○ fréquence (entier) - Fréquence du son en Hz cp.start_tone(262)
Arrêter un son commencer avec start_tone	cp.stop_tone()
Détecter quand la plaque est tapée 1 fois	cp.detect_taps = 1 if cp.tapped: print("touché 1 fois détecté!")
Détecter quand la plaque est tapée 2 fois	while True: if cp.tapped: print("Touché 2 fois détecté!")
Détecter quand la plaque est remuée	Paramètre: <ul style="list-style-type: none"> ○ Seuil de la secousse (entier) – le seuil doit être franchi pour retourner la valeur True (par défaut: 30). Moins= plus sensible. Garder supérieur à 10. while True: if cp.shake(50): print("Grosse secousse détectée!")
Paramétrer NeoPixel LEDs	Paramétrer la couleurs des NeoPixels en RGB de 0 à 255. while True: cp.pixels[9] = (30, 0, 0) Sélection de plusieurs LEDs: cp.pixels[0:5] = [(30, 0, 0)]*5 #selectionne les leds 0, 1, 2, 3 et 4. Ne pas oublier le *5.
Paramétrer NeoPixel LEDs (en continue)	Les Leds sont numérotées dans le sens anti-horaire de <code>cpx.pixels[0]</code> à <code>cpx.pixels[9]</code> .  <p>Toutes les Leds peuvent être paramétrées à la même couleur avec <code>cp.pixels.fill</code>:</p> while True: cp.pixels.fill((30, 0, 0)) Pour éteindre toutes les Leds les régler à (0, 0, 0): cpx.pixels.fill((0, 0, 0))
Paramétrer la luminosité des NeoPixels	Régler la luminosité de toutes les Leds (de 0.0 à 1.0):

<p>Paramétrer les Pads A1 à A7 pour le toucher</p>	<pre>cp.pixels.brightness = 0.3</pre> <p>Les Touch Pads sont tout autour de la carte.</p>  <pre>while True: if cp.touch_A1: print('Touched pad A1')</pre> <p>Changer A1 en A2, A3, A4, A5, A6, A7 pour les autres Pads.</p>
<p>Paramétrer la sensibilité des pads.</p>	<p>Paramètres:</p> <ul style="list-style-type: none"> o Sensibilité (entier) – Une valeur élevée rend les Pads moins sensibles. <pre>cp.adjust_touch_threshold(200) while True: if cp.touch_A1: print('Touched pad A1 hard')</pre>

Un exemple de programme mêlant plusieurs éléments :

```
from adafruit_circuitplayground import cp
cp.pixels.brightness = 0.3
cp.pixels.fill((0, 0, 0)) # Turn off the NeoPixels if they're on!

while True:
    if cp.button_a:
        cp.pixels[0:5] = [(255, 0, 0)] * 5
    else:
        cp.pixels[0:5] = [(0, 0, 0)] * 5

    if cp.button_b:
        cp.pixels[5:10] = [(0, 255, 0)] * 5
    else:
        cp.pixels[5:10] = [(0, 0, 0)] * 5
```

Essayer de faire de même avec d'autres capteurs...