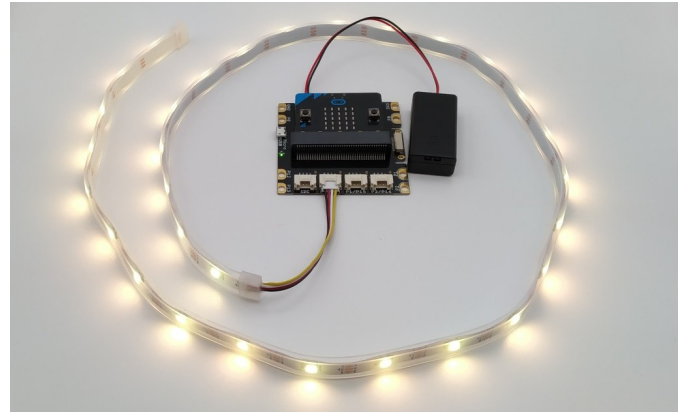
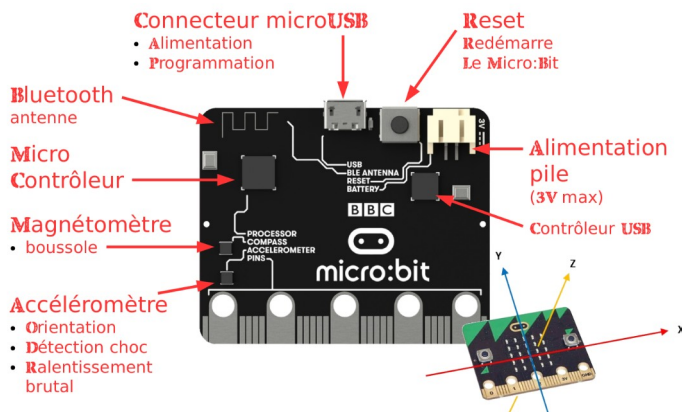


LIGHT – HAND

Arnaud Battarel

Description du projet

Light-Hand est un prototype d'installation domotique permettant de piloter la lumière avec la main. On utilise les vecteurs de l'accéléromètre de la carte Micro:bit pour piloter les couleurs et l'intensité des leds.



Liste du matériel utilisé

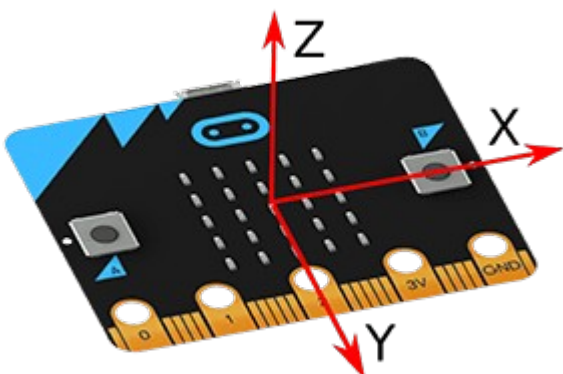
- 1 carte Micro:bit V1.5
- 1 shield Grove pour Micro:bit
- 1 module Grove ruban 30 leds

Fonctionnement des capteurs et actionneurs utilisés

- **Accéléromètre**

Récupération des données de l'accéléromètre

La carte dispose de trois accéléromètre suivant les axes Ox, Oy et Oz.



Trois méthodes permettent de récupérer les valeurs de ces accéléromètres en milli-g :

```
accelerometer.get_x()  
accelerometer.get_y()  
accelerometer.get_z()
```

Carte immobile

Lorsque la carte est immobile, ces valeurs permettent de connaître son orientation :

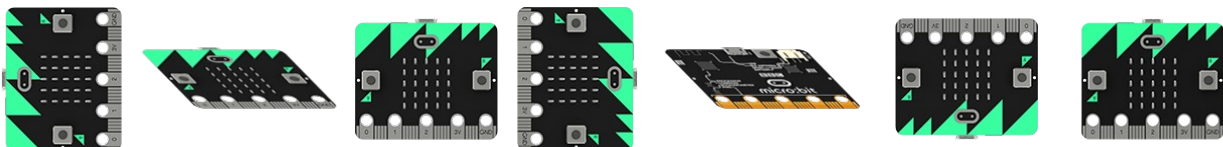
- `accelerometer.get_x()` : renvoie un nombre (entre -2048 et 2048 si la carte est immobile) correspondant à l'inclinaison de l'axe des x, la valeur 0 correspond à un axe des x horizontal.
- `accelerometer.get_y()` : renvoie un nombre (entre -2048 et 2048 si la carte est immobile) correspondant à l'inclinaison de l'axe des y, la valeur 0 correspond à un axe des y horizontal.
- `accelerometer.get_z()` : renvoie un nombre (entre -2048 et 2048 si la carte est immobile) correspondant à l'inclinaison de l'axe des z, la valeur 0 correspond à un axe des z horizontal.

Carte en mouvement

Les valeurs récupérées permettent également de savoir si la carte est en mouvement.

Utilisation de positions et mouvements prédéfinies

Pour simplifier la programmation, le langage reconnaît quelques positions et mouvements particuliers : "up", "down", "left", "right", "face up", "face down", "freefall", "3g", "6g", "8g", "shake".



"left" "face up" "up" "right" "face down" "down" "shake"

Trois méthodes utilisent ces positions particulières :

`microbit.accelerometer.is_gesture(position/mouvement)` : renvoie True si la position/mouvement de la carte correspond à la position/mouvement passée en paramètre.

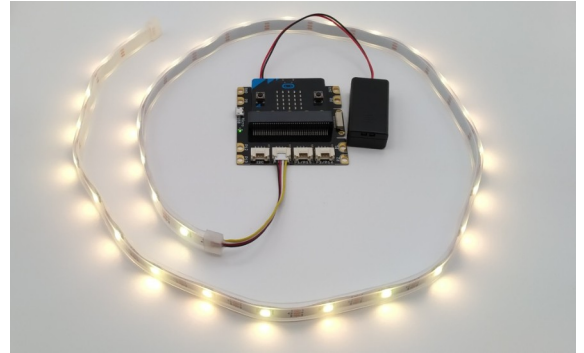
`microbit.accelerometer.was_gesture(position/mouvement)` : renvoie True si la carte a été dans la position/mouvement passée en paramètre depuis le dernier appel.

`microbit.accelerometer.get_gestures()` : renvoie un tuple de l'historique des positions/mouvements de la carte depuis le dernier appel.

• Ruban leds

→ description fonctionnelle

Le ruban leds Neopixel dispose de 30 leds adressables. La couleur de chaque led peut être contrôlée individuellement par trois valeurs correspondants aux 3 couleurs primaires : rouge, vert et bleu.



→ Pilotage du ruban de leds :

En début de programme on crée une occurrence de l'objet Neopixel et on lui spécifie ses paramètres soit le port sur lequel il est connecté (pin) et le nombre de leds du ruban.

```
>>> np=neopixel.NeoPixel(pin1, 30)
```

Les principales méthodes de la librairie Neopixel sont :

```
>>> np.clear() # pour éteindre toutes les leds
```

```
>>> np.clear(n) # pour éteindre la led n
```

```
>>> np.show() # pour allumer les leds,
```

```
>>> np.show(n) # pour allumer la led n
```

Pour assigner une couleur à une led n on attribuer le tuple (rouge, vert, bleu) à notre objet np (qui se comporte ici comme une liste de tuples)

```
>>> np[n]=(50,0,255)
```

Pour le projet, on récupère les trois valeurs x, y et z de l'accéléromètre (comprises entre -2048 et +2048) et on transforme ces valeurs pour qu'elles soient comprises entre 0 et 255.

Code Python du projet

| | |
|--|--|
| <pre>from microbit import * import neopixel nombre_de_leds = 30 np = neopixel.NeoPixel(pin1, nombre_de_leds) np.clear() while True : xdeg = accelerometer.get_x() ydeg = accelerometer.get_y() zdeg = accelerometer.get_z() rouge = int((xdeg+2048)/4096*255) vert = int((ydeg+2048)/4096*255) bleu = int((zdeg+2048)/4096*255) couleur = (rouge,vert,bleu) for x in range(nombre_de_leds): np[x] = couleur np.show() sleep(100)</pre> | <pre># import de la librairie microbit # import de la librairie neopixel # on stocke le nombre de leds dans une variable # création d'une occurrence de neopixel sur le pin 1 # réinitialisation du ruban leds # boucle infinie # on récupère les 3 valeurs de l'accéléromètre # on calcule les valeurs rouge, vert et bleu en fonction # des valeurs de l'accéléromètre. On transforme une # valeur comprise entre -2048 et +2048 en une valeur # comprise entre 0 et 255 # on attribue aux 30 leds les mêmes valeurs RGB # On allume les leds # on patiente 0,1 seconde</pre> |
|--|--|

Extensions ou développements possibles

On peut attribuer aux gestes reconnus par la Microbit d'autres fonctions :

- éteindre la lumière en tournant la carte vers le bas
- déclencher un clignotement avec un "shake"
- ...

On peut également imaginer ajouter des capteurs et actionneurs pour piloter la lumière :

- capteur de son ou de lumière
- capteur de distance
- potentiomètre

Ce dispositif pourrait être intégré dans une télécommande type TV, box ou domotique.