

I. Situation

Il s'agit d'afficher des chiffres (0 à 9) via un afficheur BCD 7 segments en fonction de l'appui sur des boutons, en utilisant une pyboard (c'est un micro contrôleur avec des fonctionnalités similaires au Raspberry pi). Pour cela, la pyboard est montée sur une carte comportant différents composants (voir figure 1).

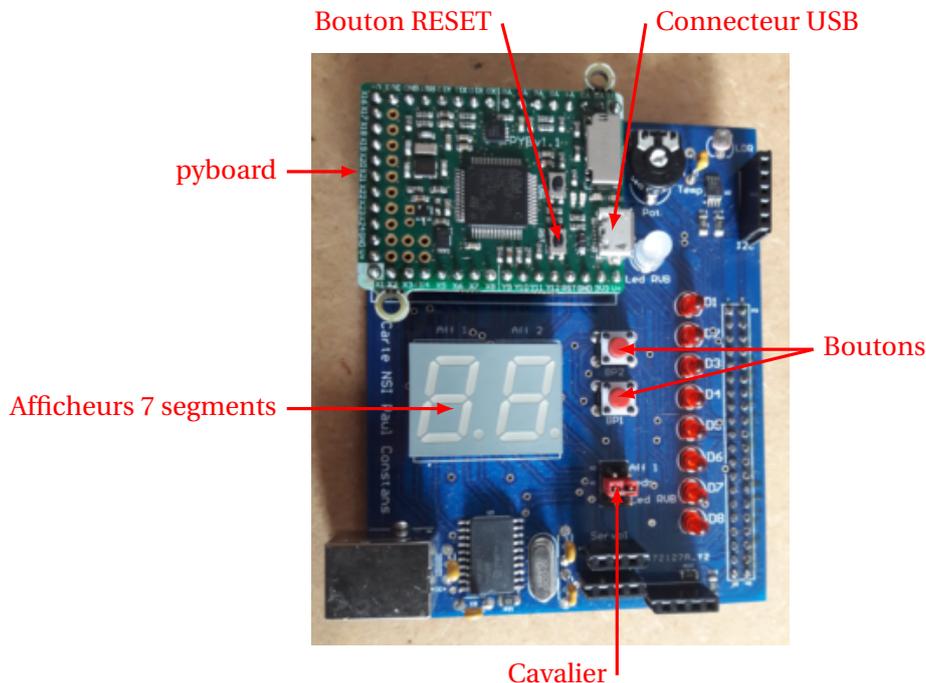


FIGURE 1 – pyboard sur la carte NSI

II. Le matériel

Vérifier tout d'abord que le cavalier est en position Aff 1 pour pouvoir utiliser les afficheurs 7 segments (sur la photo, il est en position Leds pour utiliser les 8 leds à droite de la carte).

Connecter ensuite la pyboard à un ordinateur à l'aide du câble USB. Normalement, un lecteur PYBFLASH se monte automatiquement dans un sous répertoire à votre nom du répertoire /media/. Si ce n'est pas le cas, appeler un enseignant.

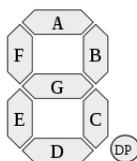


FIGURE 2 – afficheur BCD 7 segments. Les segments sont connectés aux ports X5, X6, ..., X12

III. La programmation

1) En console

Une fois la pyboard en marche, vous pouvez y accéder en tapant dans une console : `screen /dev/ttyACM0`

Vous vous trouvez alors dans une console python dans laquelle vous pouvez taper des instructions.

Pour gérer l'afficheur, il va falloir allumer ou éteindre les différents segments (voir figure 2). Par exemple, taper le code suivant dans la console python ouverte :

```
segA = pyb.Pin('X5', pyb.Pin.OUT_PP)
segA.high()
segA.low()
```

La première ligne déclare la broche X5 comme sortie (OUT) de type PP (Push Pull) et mémorise la référence dans la variable `segA`. Ensuite, `segA.high()` indique de mettre le port en position haute (3,3V), ce qui allume le segment correspondant, et `segA.low()` indique de le mettre en position basse (0V) ce qui l'éteint.

Pour les boutons, il faut configurer la broche comme entrée (IN). On peut lire alors sa valeur grâce à la méthode `value`. Les boutons sont connectés aux ports X20 et X21.

Exemple : taper le code suivant, en testant la valeur renvoyée par `bp1.value()` selon que le bouton 1 est appuyé ou non :

```
bp1 = pyb.Pin('X20', pyb.Pin.IN)
bp1.value()
```

2) Dans un script

Ouvrir le fichier `main.py` sur le lecteur PYBFLASH, et effacer son contenu s'il y en a un. Vous pouvez écrire un script dans ce fichier qui sera exécuté à chaque démarrage de la pyboard. Pour éviter de débrancher la pyboard à chaque fois, on peut la redémarrer en appuyant sur le bouton RESET (voir figure 1).

Par exemple, vous pouvez essayer le programme ci-dessous :

```
bp1 = pyb.Pin('X20', pyb.Pin.IN)
seg = pyb.Pin('X11', pyb.Pin.OUT_PP)
while True:
    if bp1.value() == 0:
        seg.high()
    else:
        seg.low()
    pyb.delay(100) # délai de 100 ms pour laisser à la pyboard le temps de gérer
```

Attention : Quand vous sauvegardez votre programme sur la pyboard, une led rouge s'allume. Attendez qu'elle s'éteigne avant de redémarrer la pyboard pour exécuter votre programme.

Remarque : Pour simplifier les tests, vous pouvez récupérer le fichier

<https://raw.githubusercontent.com/micropython/micropython/master/tools/pyboard.py>

et le sauvegarder dans votre dossier de travail. Ensuite, lancer Geany et, dans le menu Construire, choisir Définir les commandes de construction. Cliquer sur l'étiquette de la deuxième commande d'exécution (dernière ligne), et saisir MicroPython puis, dans la commande, taper la commande ci-dessous et valider.

```
python3 pyboard.py --device '/dev/ttyACM0' "%f"
```

Commandes d'exécution			
1.	Execute	python3 "%f"	<input type="button" value="✕"/>
2.	MicroPython	python3 pyboard.py --device '/dev/ttyACM0' "%f"	<input type="button" value="✕"/>

%d, %e, %f, %p et %l sont remplacés dans les dossiers et les commandes, voir le manuel pour plus de détails.

Vous pourrez alors éditer vos script dans ce répertoire, et les tester en cliquant sur MicroPython dans le menu Construire.

IV. Votre mission

Vous devez gérer l'afficheur 7 segments pour afficher un compteur : à chaque fois qu'on appuie sur le bouton Bp2, le compteur augmente de 1, et à chaque fois qu'on appuie sur le bouton Bp1, il diminue de 1.

Pensez à factoriser votre code, en évitant autant que possible les copier-coller. Pour cela, l'utilisation de boucles est obligatoire, de même que celle de fonctions. On commencera par se demander quels sont les segments à allumer pour afficher chacun des 10 chiffres.

Si vous avez fini, vous pouvez vous amuser à reprendre la même idée, mais en utilisant le deuxième afficheur pour pouvoir compter au delà de 10, ou encore en affichant l'écriture en binaire du compteur sur les 8 leds (ce qui permet de compter jusqu'à ... ?).