

2.3 – Programmer en Python

ACTIVITÉ 1

D'après toi, que fait le programme ci-dessous ?

```
from microbit import *
e0 = button_a.is_pressed()
if e0:
    display.show("1")
else:
    display.show("0")
```

Ouvrir l'éditeur Mu-éditeur, recopier ce programme et le téléverser sur la carte (bouton Flasher ).

2. **Observe** le fonctionnement de la carte et **propose** une ou plusieurs interprétations (pour confirmer tes observations, tu peux manipuler les boutons présents sur la carte : A / B / Reset).

Présenter tes réponses au professeur.

3. **Indique** le nombre de fois qu'est exécutée l'instruction conditionnelle.
4. **D'après toi**, combien de fois doit être exécutée l'instruction conditionnelle pour que la carte reste à l'écoute des entrées et réagisse chaque fois que l'état du bouton A change ?
5. **Améliore** le programme en tenant compte de tes observations et de tes réponses aux questions précédentes.

Présente la carte programmée au professeur.

PAGE À COLLER SUR LE CAHIER

PAGE À COLLER SUR LE CAHIER

PAGE À COLLER SUR LE CAHIER

Voici l'algorithme le plus courant d'un système informatique embarqué :

```
initialiser les actionneurs à leurs positions de départ
TANT QUE vrai :
    | lire les informations des capteurs
    | traiter ces informations
    | calculer les informations pour les actionneurs
    | écrire ces informations sur les actionneurs
```

REMARQUE

Quand on programme un calcul en mathématique, on veut qu'il s'arrête. Pour un système embarqué, **c'est le contraire**.

Exemple

C'est avec ce genre d'algorithme que le calculateur de bord d'une voiture autonome peut suivre la bonne trajectoire sur l'autoroute en **lisant** la position des lignes blanches et en **actionnant** le volant.

Exemple

Voici un programme écrit en Python qui permet de contrôler l'affichage des LED de la carte micro :bit en fonction de l'appui sur le bouton A :

```
from microbit import *
while True:
    if button_a.is_pressed():
        display.show(Image.HAPPY)
    else:
        display.clear()
```

Documentation du système de pilotage de Tesla

Huit caméras panoramiques offrent une visibilité à 360° autour de la voiture avec une portée de 250 m. À cela viennent s'ajouter douze capteurs à ultrasons nouvelle génération, permettant de détecter les objets solides ou les liquides avec une portée et une précision quasiment deux fois supérieure au système précédent. Un radar avant à haute capacité de traitement complète les données environnementales grâce à une longueur d'onde redondante capable de voir à travers de fortes pluies, le brouillard, la poussière et même la voiture qui vous précède.

Afin de traiter toutes ces données, un nouvel ordinateur de bord doté d'une puissance de calcul 40 fois supérieure à la génération précédente s'appuie sur le nouveau réseau neuronal Tesla pour les logiciels de visualisation, de sonar et de radar. Ce système offre ainsi une perception de l'environnement infiniment supérieure à celle d'un conducteur, surveillant simultanément toutes les directions, sur des longueurs d'ondes imperceptibles par l'homme.

Autopilot permet à votre véhicule de maintenir une trajectoire, d'accélérer et de freiner automatiquement dans sa voie. Les fonctionnalités actuelles de l'Autopilot exigent une surveillance active de la part du conducteur et ne rendent pas le véhicule autonome.

source : https://www.tesla.com/fr_FR/autopilot



ACTIVITÉ 2

Énumère (1) les capteurs disponibles sur cette voiture (à partir du texte et aussi de vos déductions) et (2) les actionneurs indispensables pour le pilotage de cette voiture.

Identifie quelques algorithmes de contrôle utilisés.