

Activité 2 - 3

Traiter l'information numérique : python

Volet connecté

Problématique : Lors de l'installation du produit chez le client le technicien de maintenance doit vérifier le fonctionnement du produit avant de le livrer au client. Le déficit d'un seul des éléments doit amener le technicien soit à le remplacer, soit à le dépanner.

1. Présentation

1.1. Généralités

Les objectifs :

- Tester les capteurs et les actionneurs du produit.
- Tester le fonctionnement automatique.
- Tester le fonctionnement à distance.
- Tester le réglage du capteur de luminosité.



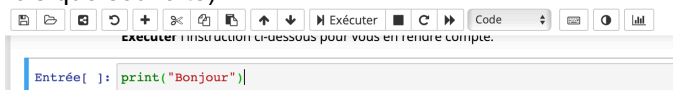
Problématique de l'activité :

Comment vérifier un bon fonctionnement des capteurs et des actionneurs ?

2. Découverte et prise en main.

Si vous n'avez aucune notion de python ou si vous souhaitez découvrir ou redécouvrir à votre rythme ce codage :

- Ouvrir le site : <https://basthon.fr>
- Choisir « Notebook »
- Charger le premier fichier ressource : intro1.ipynb
- Déplacer vous sur les cases de codage et faire Exécuter (Vous pourrez refaire exécuter autant de fois que souhaité).



- Continuer le travail demandé.
- A la fin du premier, charger et refaire ce travail avec les autres fichiers.

3. Programmation de découverte python avec microbit.

Exercice 1 : Clignotement de la led du volet

On désire réaliser le fonctionnement suivant :

Lorsque l'on appuie sur un bouton poussoir (Button.A), une led (P2) doit clignoter. Lorsque l'on relâche le bouton poussoir, la lampe doit s'éteindre. Le programme fonctionne en boucle.

Ici le symbole # correspond à un commentaire

Ci-dessous le mot clé « def » permet de créer une fonction

On définit la fonction on_forever()

def on_forever():

Dans python pour séparer les structures il faut indenter (tabulation sur le clavier)

La structure alternative à une action se réalise si le bouton A est pressé :

if input.button_is_pressed(Button.A) :

#Si le bouton est pressé alors la led connectée à P2 s'allume (1023 sur 10 bits)

pins.analog_write_pin(AnalogPin.P2, 1023)

#Une pause de 500 ms est réalisée

pause(500)

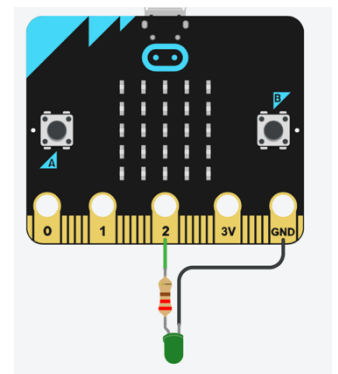
La mise à zéro de la sortie se fait en affectant P2 à 0

pins.analog_write_pin(AnalogPin.P2, 0)

pause(500)

On lance en boucle la fonction « on_forever » par la fonction basic.forever()

Voir le code ci-après :



basic.forever(**on_forever**)
Voici le programme sans les
commentaires :

```
def on_forever():
    if input.button_is_pressed(Button.A) :
        pins.analog_write_pin(AnalogPin.P2, 1023)
        pause(500)
        pins.analog_write_pin(AnalogPin.P2, 0)
        pause(500)

basic.forever(on_forever)
```

Q2	Réaliser le programme sur internet en utilisant le site Tinkercad.
----	--

3.1 Programme 2 : Allumage de la Led si action sur un bouton.

Q8	Proposer un montage et un programme permettant d'allumer une Led si un bouton est pressé. Après validation, téléverser votre programme et vérifier le fonctionnement.
----	--

2.1 Programme 3 : Mesure de la luminosité et allumage de la Led.

Il s'agit ici de remplacer l'action du bouton poussoir par la mesure de la luminosité comme sur le produit. L'instruction associée à la mesure analogique de cette grandeur est analogRead().

Q9	Proposer un programme permettant d'allumer la Led si la valeur de la luminosité est inférieure ou égale à 50 lux. Téléverser votre programme et vérifier le bon fonctionnement.
----	--

2.2 Programme 4 : Clignotement de la Led si baisse de la luminosité avec appel de fonction.

Dans cette partie vous allez réaliser un appel de fonction avec passage de paramètres.

Q10	En utilisant le programme1, construire une fonction et son appel de fonction permettant le clignotement si la luminosité est inférieure ou égale à 50 lux. Vous utiliserez l'instruction for() de python. Téléverser votre programme et vérifier le bon fonctionnement.
-----	--