

Correction du devoir de synthèse n°3

Exercice 1 : (...../3 points)

Compléter la définition ci-dessous par le mot convenable à partir de la liste suivante :

communique, Internet, objet connecté, objet

Un **objet connecté** est un objet qui **communique** avec d'autres objets via Internet.

Exercice 2 : (...../5 points)

Réécrire dans la colonne **Correction** les phrases données dans la colonne **Proposition** en apportant les modifications nécessaires afin qu'elles soient correctes.

Proposition	Correction
IOT est l'abréviation de Index Organized Table.	IOT est l'abréviation de Internet Of Things.
Un capteur réalise l'action spécifiée par l'interface programmable.	Un actionneur réalise l'action spécifiée par l'interface programmable.
La carte ESP32 contient une seule LED interne.	La carte ESP32 contient deux LEDs internes.
La LED interne rouge est connectée au GPIO 2.	La LED interne bleue est connectée au GPIO 2.
While(True) : s'agit d'une boucle finie.	While(True) : s'agit d'une boucle infinie.

Exercice 3 : (...../12 points)

Soit le programme suivant qui permet d'allumer une LED externe rouge :

```
1 from machine import Pin
2 from time import sleep
3 ledr=Pin(15,Pin.OUT)
4 while(True):
5     ledr.value(True)
6     sleep(2)
7     ledr.value(False)
8     sleep(2)
```

1. Modifier le programme précédent pour que la LED rouge s'allume puis s'éteint. Ensuite la LED jaune s'allume puis s'éteint.

NB : il faut tout d'abord créer un PIN de sortie sur la GPIO 23

```
from machine import Pin #importation de la classe Pin de la bibliothèque machine
from time import sleep #importation de la classe sleep de la bibliothèque time
ledr=Pin(15,Pin.OUT) #créer un Pin de sortie sur la GPIO 15
ledj=Pin(23,Pin.OUT) #créer un Pin de sortie sur la GPIO 23
while(True): #boucle infinie
    ledr.value(True) #allumer la LED rouge
    sleep(2) #attendre 2 secondes
    ledr.value(False) #éteindre la LED rouge
    sleep(2) #attendre 2 secondes

    ledj.value(True) #allumer la LED jaune
    sleep(2) #attendre 2 secondes
    ledj.value(False) #éteindre la LED jaune
    sleep(2) #attendre 2 secondes
```

2. Modifier le programme modifié au niveau de la question précédente pour que la LED rouge s'allume puis s'éteint. Ensuite la LED jaune s'allume puis s'éteint. Enfin, la LED verte s'allume puis s'éteint.

NB : il faut tout d'abord créer un PIN de sortie sur la GPIO 22

```
from machine import Pin #importation de la classe Pin de la bibliothèque machine
from time import sleep #importation de la classe sleep de la bibliothèque time
ledr=Pin(15,Pin.OUT) #créer un Pin de sortie sur la GPIO 15
ledj=Pin(23,Pin.OUT) #créer un Pin de sortie sur la GPIO 23
ledv=Pin(22,Pin.OUT) #créer un Pin de sortie sur la GPIO 22
while(True): #boucle infinie
    ledr.value(True) #allumer la LED rouge
    sleep(2) #attendre 2 secondes
    ledr.value(False) #éteindre la LED rouge
    sleep(2) #attendre 2 secondes

    ledj.value(True) #allumer la LED jaune
    sleep(2) #attendre 2 secondes
    ledj.value(False) #éteindre la LED jaune
    sleep(2) #attendre 2 secondes

    ledv.value(True) #allumer la LED verte
    sleep(2) #attendre 2 secondes
    ledv.value(False) #éteindre la LED verte
    sleep(2) #attendre 2 secondes
```

3. Modifier le programme modifié au niveau de la question précédente pour que la LED jaune clignote 5 fois avant que la LED verte s'allume.

```
from machine import Pin #importation de la classe Pin de la bibliothèque machine
from time import sleep #importation de la classe sleep de la bibliothèque time
ledr=Pin(15,Pin.OUT) #créer un Pin de sortie sur la GPIO 15
ledj=Pin(23,Pin.OUT) #créer un Pin de sortie sur la GPIO 23
ledv=Pin(22,Pin.OUT) #créer un Pin de sortie sur la GPIO 22
while(True): #boucle infinie
    ledr.value(True) #allumer la LED rouge
    sleep(2) #attendre 2 secondes
    ledr.value(False) #éteindre la LED rouge
    sleep(2) #attendre 2 secondes

    for i in range(5): #répéter le traitement 3 fois
        ledj.value(True) #allumer la LED jaune
        sleep(2) #attendre 2 secondes
        ledj.value(False) #éteindre la LED jaune
        sleep(2) #attendre 2 secondes

    ledv.value(True) #allumer la LED verte
    sleep(2) #attendre 2 secondes
    ledv.value(False) #éteindre la LED verte
    sleep(2) #attendre 2 secondes
```

BON TRAVAIL...