

# Correction

# TP 0

- 1- Classer les composants matériels présentés dans l'annexe du TP0 dans le tableau au-dessous.  
 2- Chercher le rôle des composants non reconnus et ajouter une petite description sur vos portfolios.

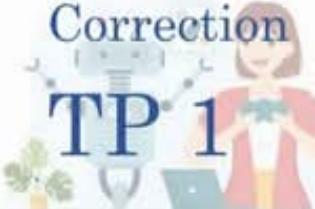


Capteurs	Unités de traitements	Actionneurs
 <b>CAPTEURS DE GAZ</b>  <b>DÉTECTEUR D'OBSTACLE</b>	 <b>Carte microbit</b>  <b>Carte Arduino</b>	 <b>Des leds</b>
 <b>CAPTEUR D'ÉLEVATION</b>  <b>Bouton poussoir</b>	 <b>Carte Raspberry</b>  <b>Carte Esp32</b>	 <b>Buzzer</b>  <b>Afficheur 7 segments</b>
 <b>CAPTEUR DE FLAMME</b>  <b>CAPTEUR DE MOUVEMENT PIR</b>		 <b>Afficheur LCD</b>  <b>Servomoteur</b>
 <b>CAPTEUR D'HUMIDITÉ DES SOLS</b>  <b>capteur de température</b>  <b>DHT11</b>  <b>DÉTECTEUR D'OBSTACLE</b>		 <b>Ventilateur</b>

Enseignante : Mme Marwa ZOUAOUI

# Correction

## TP 1



- Ouvrir le logiciel Thonny et saisir ces programmes puis les exécuter avec une carte ESP32 : écrire vos constatations.
- Téléverser le programme « Prog 3 » dans la carte.



Prog 1	Prog 2	Prog 3
<pre> 1 from machine import Pin 2 print('hello esp') 3 P2 = Pin(2,Pin.OUT) 4 P2.on()  → Résultat: .....  Une led intégrée dans la carte .....  s'allume en bleu.....    Supprimer la première instruction, que constatez-vous ? <b>La led ne s'allume plus</b>  Remplacer l'instruction <b>P2.on()</b> par <b>P2.off()</b>. Que se passe-t-il ? <b>La led s'éteint</b>  Remplacer <b>P2.on()</b> par <b>P2.value(1)</b> puis par <b>P2.value(0)</b>.  Maintenant, essayer <b>P2.high()</b> et <b>P2.low()</b>.  → <b>En déduire :</b> ➢ Pour allumer une led, on utilise : <b>P2.on()</b> ou <b>P2.value(1)</b> ou <b>P2.high()</b> .....  ➢ Pour allumer une led, on utilise : <b>P2.off()</b> ou <b>P2.value(0)</b> ou <b>P2.low()</b>.....</pre>	<pre> 1 import time 2 from machine import Pin 3 print('hello esp') 4 P2 = Pin(2,Pin.OUT) 5 P2.on() 6 time.sleep(5) 7 P2.off()  → Résultat: .....  Une led intégrée dans la carte s'allume .....  pendant 5 secondes puis s'éteint.    Remplacer <b>time.sleep(5)</b> par <b>time.sleep(2)</b>, que constatez-vous ? <b>La led s'éteint après 2 secondes</b>  Alors, quel est le rôle de <b>time.sleep(5)</b> ?  <b>Elle fait une pause (attendre) pendant 5 secondes avant d'exécuter l'instruction suivante.</b></pre>	<pre> 1 import time 2 from machine import Pin 3 print('hello esp') 4 P2 = Pin(2,Pin.OUT) 5 while True : 6     P2.on() 7     time.sleep(1) 8     P2.off() 9     time.sleep(1)  </pre> <p>→ <b>Résultat: .....  Une led intégrée dans la carte clignote indéfiniment.</b></p> <p>Supprimer l'instruction <b>while True :</b>, que constatez-vous ? <b>Une led intégrée dans la carte clignote une seule fois.</b></p> <p>En déduire le rôle de cette instruction : <b>Elle permet de répéter un bloc d'instructions indéfiniment.</b></p>

Enseignante : Mme Marwa ZOUAOUI

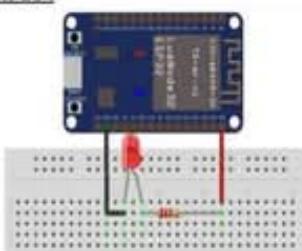


# Correction TP 2

- 1- Brancher une led externe à la carte ESP32 selon les schémas proposés puis écrire un programme avec Thonny pour la faire clignoter (voir « prog 3 » du TP1).  
2- On veut ajouter un capteur : le bouton poussoir pour contrôler la led.

## Clignoter une led externe

Branchements de la led dans le Pin 23 :



TP2 clignoter une led externe.py \*

```

1 import time
2 from machine import Pin
3 led = Pin(23,Pin.OUT)
4 while True:
5     led.on()
6     time.sleep(1)
7     led.off()
8     time.sleep(1)

```

## Contrôler une led avec un bouton poussoir

Terminer le branchement sachant que :



Créer un code pour allumer la led si le bouton est enfoncé.

TP2 allumer une led avec un bouton.py \*

```

1 from machine import Pin
2 led = Pin(23, Pin.OUT)
3 button = Pin(17, mode=Pin.IN, pull=Pin.PULL_UP)
4 while True:
5     if button.value()==1:
6         led.on()
7     else:
8         led.off()

```

Enseignante : Mme Marwa ZOUAOUI