



# 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
 CAPTEURS DE GAZ  DETECTEUR D'OBSTACLE  CAPTEUR ULTRASON  Bouton poussoir  CAPTEUR DE FLAMME  CAPTEUR DE MOUVEMENT PIR  CAPTEUR DE SON  CAPTEUR D'HUMIDITÉ DES SOLS  DS18B1C capteur de température  DETECTEUR D'OBSTACLE	 Carte microbit  Carte Arduino  Carte Raspberry  Carte Esp32	 Des leds  Buzzer  Afficheur 7 segments  Afficheur LCD  Servomoteur  Ventilateur

Enseignante : Mme Marwa ZOUAOUI

## Correction

## TP 1

- 1- Ouvrir le logiciel Thonny et saisir ces programmes puis les exécuter avec une carte ESP32 : écrire vos constatactions.
- 2- 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() </pre> <p>➡ <u>Résultat :</u> .....</p> <p>..Une led intégrée dans la carte.....</p> <p>..s'allume en bleu.....</p> <p>Supprimer la première instruction, que constatez-vous ? <b>La led ne s'allume plus</b></p> <p>Remplacer l'instruction <b>P2.on()</b> par <b>P2.off()</b>.</p> <p>Que se passe-t-il ? <b>La led s'éteint</b></p> <p>Remplacer <b>P2.on()</b> par <b>P2.value(1)</b> puis par <b>P2.value(0)</b>.</p> <p>Maintenant, essayer <b>P2.high()</b> et <b>P2.low()</b>.</p> <p>➡ <u>En déduire :</u></p> <ul style="list-style-type: none"> <li>➢ Pour allumer une led, on utilise : <b>P2.on()</b> .. ou <b>P2.value(1)</b> .. ou <b>P2.high()</b> ..</li> <li>➢ Pour allumer une led, on utilise : <b>P2.off()</b> .. ou <b>P2.value(0)</b> .. ou <b>P2.low()</b> ..</li> </ul>	<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() </pre> <p>➡ <u>Résultat :</u> .....</p> <p><b>Une led intégrée dans la carte s'allume pendant 5 secondes puis s'éteint.</b></p> <p>Remplacer <b>time.sleep(5)</b> par <b>time.sleep(2)</b>, que constatez-vous ?</p> <p><b>La led s'éteint après 2 secondes</b></p> <p>Alors, quel est le rôle de <b>time.sleep(5)</b> ?</p> <p><b>Elle fait une pause (attendre) pendant 5 secondes avant d'exécuter l'instruction suivante.</b></p>	<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>➡ <u>Résultat :</u> .....</p> <p><b>Une led intégrée dans la carte clignote indéfiniment.</b></p> <p>Supprimer l'instruction <b>while True :</b>, que constatez-vous ?</p> <p><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



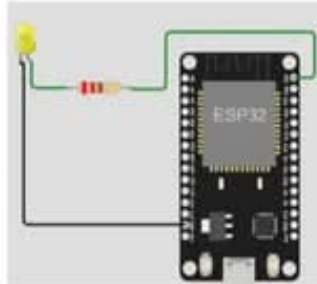
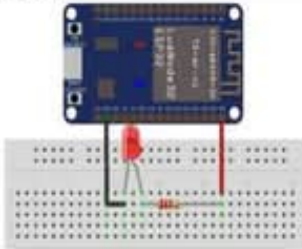
## Correction

## 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

Branchement de la led dans le  
Pin 23 :

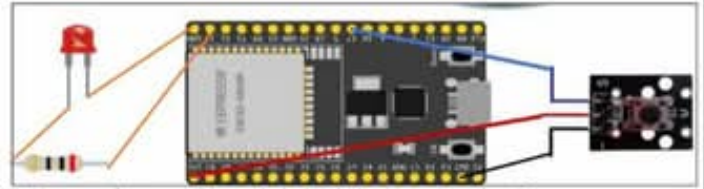


TP2 clignoter une led externe.py \* >

```
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