

Mon cahier



de

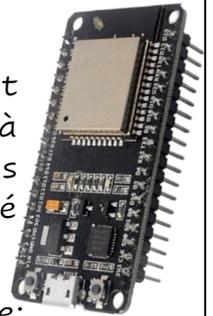
ROBOTIQUE



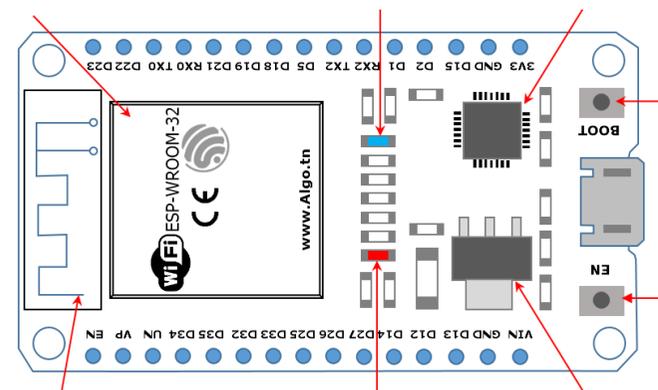
La **robotique** est l'ensemble des techniques permettant la conception et la réalisation de machines automatiques ou des robots. Un **robot** est un système alimenté en énergie qui évolue dans un environnement statique ou dynamique, il est formé d'un microcontrôleur ainsi que d'un ou plusieurs capteurs et actionneurs. (wikipédia)

La carte ESP32 :

L'ESP32 développée par la société Espressif, est une carte de développement à faible coût dédié à l'internet des objets (IoT) et les applications embarquées. C'est un (SoC) system on a chip doté de communications sans fil Wifi et Bluetooth.



- Identifier les différentes composantes de la carte:



Savoir plus

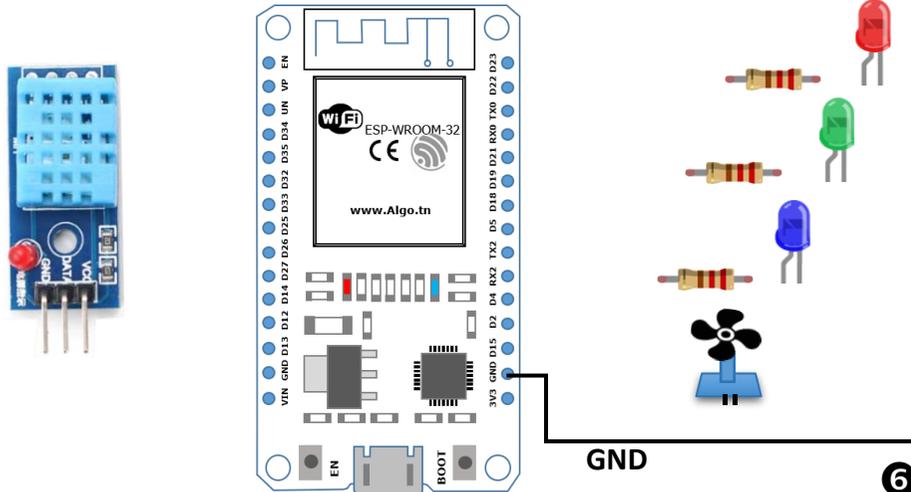
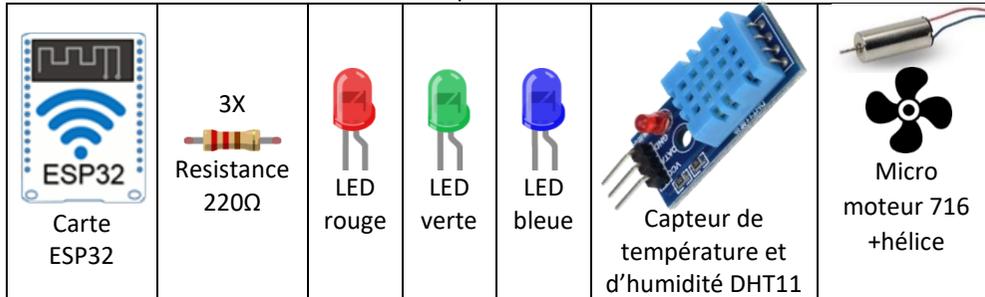
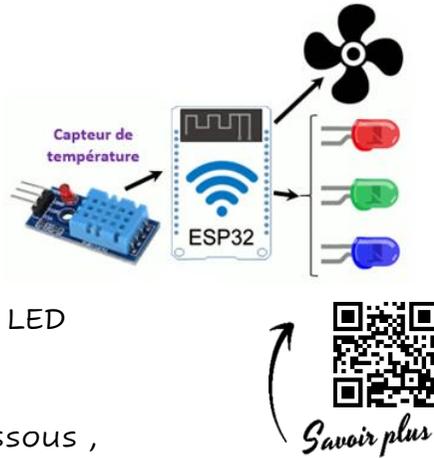
Mini Projet 5

On veut allumer des lampes LED et un ventilateur suivant une température capturée.

- Si température <14 allumer la LED bleue
- Si la température >=14 et <26 allumer la LED verte
- Si la température >26 allumer la LED rouge et actionner le ventilateur.

Etape 1: Montage

En utilisant les composants ci-dessous, réaliser un montage permettant d'allumer les LEDs et d'actionner le moteur suivant la température capturée. La capture de la température doit se faire sur le pin35. Les LED rouge, verte et bleue à brancher respectivement sur les pin 19, 18,5 et le moteur sur le pin 15



6

Suite Mini projet 1-étape2 :

- Python utilise la notion d'indentation (espacement) pour spécifier un bloc d'instructions.
- 2. Encercler un bloc d'instructions dans le programme ci-contre.
- Python possède plusieurs bibliothèques qu'on importe au début du code, ces biblio permettent d'importer des codes sans les coder soi-même.
- 3. Donner le mot utilisé pour réaliser l'importation
- 4. Donner les bibliothèques utilisées :

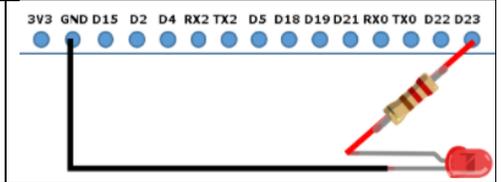
Etape 3 :

Modifier le programme précédent pour clignoter la LED avec un intervalle de 2s.

Mini Projet 1(Corrigé) : Lampe clignotante

Etape 1 :

Un montage permettant de faire clignoter la lampe led (à brancher sur le pin23 de la carte ESP32).



Etape 2 : Autre formulation de code MicroPython et solution :

```
import time
from machine import Pin # importer pin à partir du module machine
led=Pin(23, Pin.OUT) # Définir pin 23 comme output (sortie)
while True: # boucle infinie -> condition toujours vrai
    led.value(1) # ou led.on() Allumer la LED
    time.sleep(1) # attendre 1 seconde
    led.value(0) # ou led.off() Eteindre la LED
    time.sleep(1) # attendre 1 seconde
```

Mini Projet 2: Sirène d'alarme extérieure factice

Pour réaliser une Sirène d'alarme extérieure factice, on va utiliser 2 diodes leds et une carte ESP32. Faire le montage puis le script micro-python permettant de :

- 1-Allumer led1 pendant 0.5 seconde puis l'éteindre
- 2-puis allumer led2 pendant 0.5 seconde puis l'éteindre et on recommence

Utiliser le site wokwi.com pour faire le montage et écrire le script python :

<https://tinyurl.com/projet3-alarme>

3

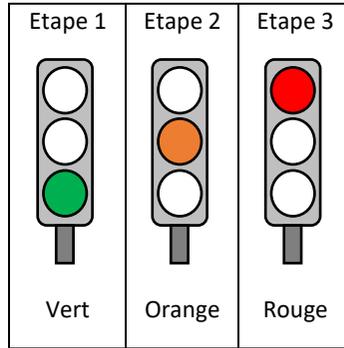


Mini Projet 3: Feu de circulation

Réaliser un feu de circulation en utilisant 3 diodes LED de couleurs (rouge, vert, orange) et 3 résistances de 220 Ω et la carte ESP32.

La séquence est la suivante :

1. Le feu vert s'allume pendant 3 secondes, puis s'éteint,
2. Le feu orange s'allume pendant 1 seconde puis s'éteint,
3. Le feu rouge s'allume pendant 3 secondes, puis s'éteint, et on recommence

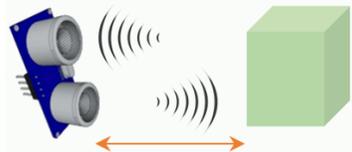
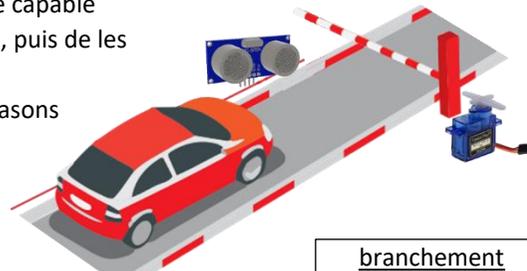


Mini Projet 4 : Barrière automatique

On va lever la barrière avec le servomoteur lorsque le capteur ultrason détecte un objet.

Un servomoteur est un système motorisé capable d'atteindre des positions prédéterminées, puis de les maintenir.

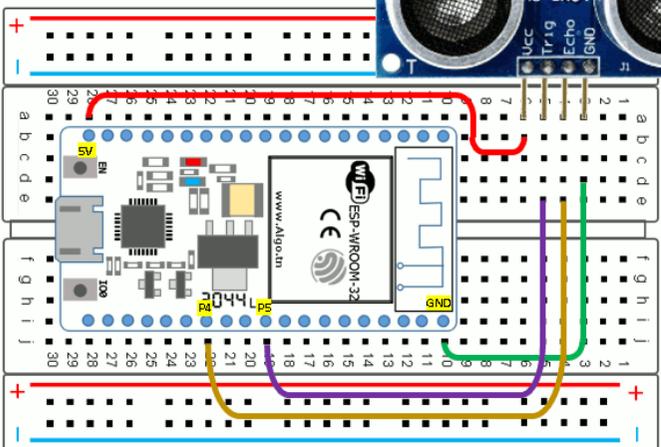
Capteur ultrason :HC-SR04 utilise les ultrasons pour déterminer la distance à laquelle se trouve un objet.



branchement servomoteur:
 Marron → GND
 Rouge → 5V
 Orange → pin23(PWM)



servomoteur 4

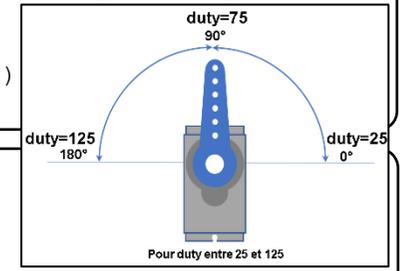


Utiliser les scripts suivants pour lever la barrière avec le servomoteur lorsque le capteur ultrason détecte un objet avec une distance <4 cm .

```

from hcsr04 import HCSR04
from time import sleep
sensor=HCSR04(trigger_pin=5,echo_pin=4,echo_timeout_us=10000)
while True:
    distance = sensor.distance_cm()
    print('Distance=', distance, 'cm')
    sleep(1)
    
```

Script 1



```

from machine import Pin,PWM
from time import sleep
servo = PWM(Pin(23),freq=50)
while True:
    x=int(input('valeur entre (25-125)='))
    servo.duty(x)
    sleep(1)
    
```

Script 2

```

from machine import Pin,PWM
from hcsr04 import HCSR04
from time import sleep
sensor=HCSR04(trigger_pin=5,echo_pin=4,echo_timeout_us=10000)
servo=PWM(Pin(23),freq=50)
led=.....
buzzer=.....
while True:
    distance = sensor.distance_cm()
    print('D=', distance, 'cm')
    if distance<4 :
    
```

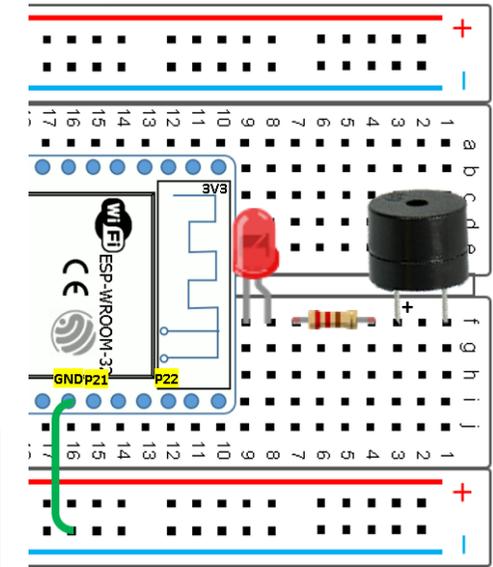
Un buzzer actif peut générer du son quand il reçoit une tension continue.



```

        servo.duty(75)
        sleep(1)
    else:
        servo.duty(125)
        sleep(1)
    
```

Ajouter une diode Led et un Buzzer respectivement sur les pins 22 et 21 qui s'activent pendant 1s lors de la détection d'un objet. (distance < 4 cm)



Rq: On doit aussi téléverser à la carte ESP32 la bibliothèque **hcsr04.py** téléchargée à partir de : <https://github.com/rsc1975/micropython-hcsr04> lien : tinyurl.com/capteur-hcsr04

5