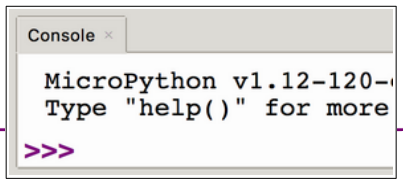
Objectif : déterminer la vitesse propagation du son dans l'air.

Rappel : Il faut établir la connexion avec la carte microcontrôleur ESP32 – Soprolab -
MENU : => Exécuter
=> sélectionner l'interpéteur … → microPython ESP32
=> Port → Silicon Lab CP210x ...



1ere étape : Contrôler une broche (Pin)

Pour commencer, vous allez allumer une led pendant 0.5s

```
# Charger la configuration des broches en mémoire
from machine import Pin
# Charger la fonction d'attente en mémoire
from time import sleep

# Déclarer une broche en sortie
led = Pin ( 12, Pin.OUT )

# Mettre l'état de la broche à on ( )
led.on ( )
# Attendre 0.5s
sleep ( 0.5 )
# Mettre la broche à off ( )
led.off ( )
```

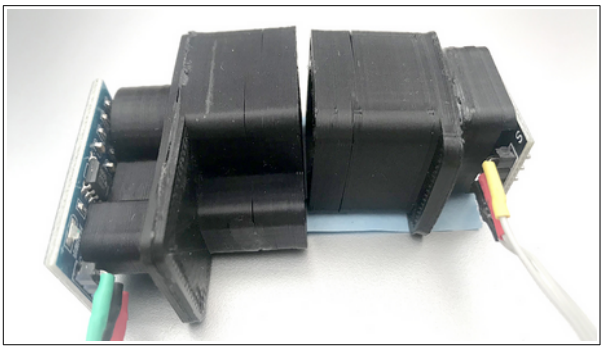
Note : la LED jaune est connectée à la broche 13 et la rouge à la broche 14.

Pour faire d'autres essais, vous pourrez par la suite changer la broche n°12 par les broches n°13 et n°14 pour sélectionner les deux autres LED jaune et rouge.

Rappel, vous devez enregistrer votre script Python dans vos documents sur votre espace personnel avant de l'exécuter . N'oubliez pas d'utiliser un dossier spécifique pour pouvoir réutiliser vos scripts ensuite.

2eme étape : Utiliser un module Buzzer / micro

Après avoir branché le module buzzer / micro, vous allez pouvoir interagir via les broches 4 et 0.
Dans un premier temps, vous allez maintenir le buzzer collé au micro comme indiqué sur l'image ci-contre ->
Par la suite, il faudra éviter de toucher le micro pour éviter les perturbations (vibrations, bruit, ...)



On va ensuite vérifier le bon fonctionnement du dispositif :

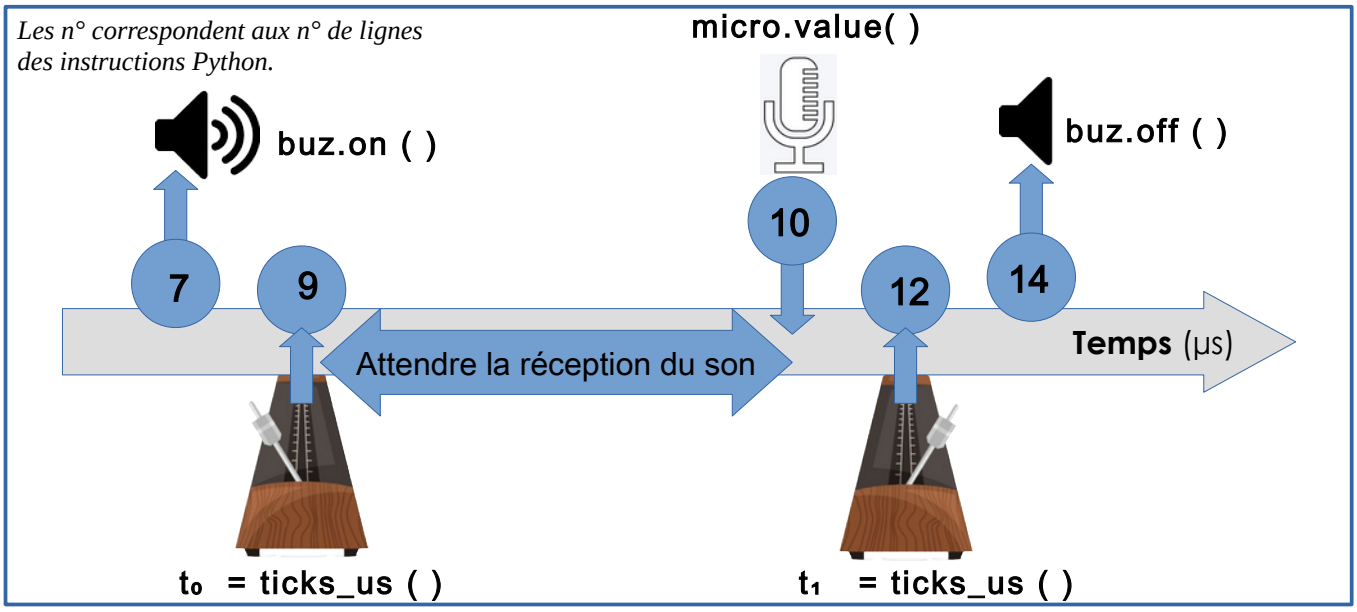
```
# Charger la configuration des broches en mémoire
from machine import Pin
# Charger la fonction d'attente en mémoire
from time import sleep

# Déclarer la broche du buzzer
buz = Pin ( 4, Pin.OUT )
# Déclarer la broche du micro
micro = Pin ( 0, Pin.IN )

# Mettre la broche du buzzer à on ( )
buz.on ( )
# Attendre 0.5s
sleep ( 0.5 )
# Mettre la broche du buzzer à off ( )
buz.off ( )
```

=> Ce code doit produire un « beep » pendant 0,5 s.
A chaque « beep » , une deuxième led verte verte doit s'allumer sur le micro.

Cette même led doit être éteinte en absence de bruit ou de vibrations.
Si la deuxième led verte reste allumée, demander à ce que la sensibilité du micro soit réglée.

Utiliser un métronome en microsecondes !

Nous allons utiliser **un métronome en microsecondes** intégré au microcontrôleur.
Pour accéder au compteur de microsecondes, on utilise la fonction **ticks_us ( )**.



3eme étape : Mesurer le temps de réaction en microsecondes !

Etant donné que le buzzer est collé au microphone, on va donc **mesurer le temps de réaction** lorsque la distance peut être considérée comme nulle.

```
# Charger la config. des broches en mémoire      1- from machine import Pin
# Charger la fonction de métronome en µs         2- from time import ticks_us
                                                 3-
# Déclarer la broche du buzzer                   4- buz = Pin ( 4, Pin.OUT )
# Déclarer la broche du micro                    5- micro = Pin ( 0, Pin.IN )
                                                 6-
# Mettre la broche du buzzer à on ( )            7- buz.on ( )
                                                 8-
# Mettre la valeur du métronome dans t0          9- t0 = ticks_us ( )
# Attendre que le micro capte un son             10- while micro.value( )==1 :
                                                 11-    pass
# Mettre la valeur du métronome dans t1          12- t1 = ticks_us (  )
                                                 13-
# Mettre la broche du buzzer à off ( )           14- buz.off ( )
# Calculer l'intervalle de temps : delta_t = t1 – t0   15- delta_t = t1 - t0
# Afficher le temps mesuré                       16- print("Temps :", delta_t,"µs")
```

Faites plusieurs mesures pour déterminer **le temps moyen de réaction**.

=> Si le code vous en dit : vous pouvez programmer les 8 essais avec une boucle "pour" `for i in range(8) :` avec une pause d'une seconde entre chaque mesure ...

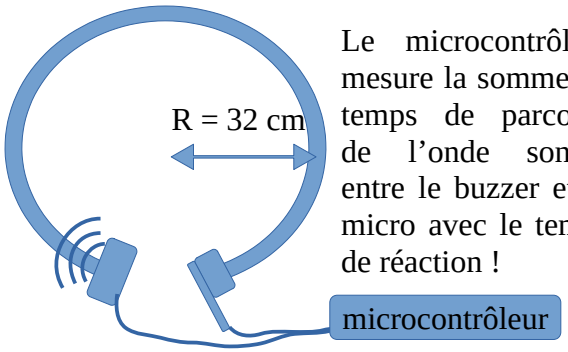
N° de mesure	1	2	3	4	5	6	7	8	Moyenne
Temps (µs)									

Vous pouvez retirer les mesures anormalement élevées ou basses pour ne garder que les mesures significatives à partir desquelles vous pouvez calculer le temps moyen de réaction.

4eme étape : En déduire le temps de propagation du son dans l'air !

1 - Placer le buzzer à une extrémité du tuyau et le microphone à l'autre extrémité.
Attention : il est inutile de forcer vous risqueriez de détériorer le matériel !
Les modules sont prévus pour être juste emboîtés sur le tuyau afin d'être maintenus en place.

2 - Refaire plusieurs mesures pour déterminer un temps moyen.



Le microcontrôleur mesure la somme du temps de parcours de l'onde sonore entre le buzzer et le micro avec le temps de réaction !

N° de mesure	1	2	3	4	5	6	7	8	Moyenne
Temps (µs)									

A faire : => A partir de la valeur obtenue, et du temps de réaction, vous pouvez en déduire le temps de parcours de l'onde sonore dans le tuyau.

3 – **Si le code vous en dit ;-)** vous pouvez modifier la ligne 15 de votre programme (calcul de delta_t) pour que le microcontrôleur fasse le calcul lui même …

4 – Calculer la circonférence d'un cercle de rayon 32cm pour déterminer la longueur du tuyau => La distance parcourue est de d = m

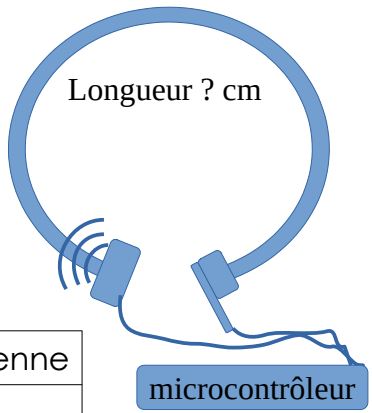
5 – En déduire la vitesse de propagation du son dans l'air contenu dans le tuyau

6 – **Si le code vous en dit ;-)** vous pouvez ajouter quelques lignes de code en Python à votre programme (calcul de la vitesse) pour que le microcontrôleur fasse le calcul lui même et affiche le résultat …

7 Comparer la valeur ainsi obtenue avec la valeur théorique de V = 340 $m.s^{-1}$ au regard de la marge d'erreur constatée lors des mesures.

5eme étape : Déterminer la longueur d'un autre tuyau ...

Vous avez calculé la vitesse de propagation du son dans l'air. Vous allez pouvoir utiliser le résultat de vos calculs, pour déterminer la longueur d'**un autre tuyau** …



A faire => Enregistrer votre script microPython sous un autre nom puis, modifier le de telle sorte que le microcontrôleur calcule puis affiche la longueur du tuyau ...

N°	1	2	3	4	5	6	7	8	Moyenne
Temps (µs)									

Déterminer la valeur moyenne de la longueur du tuyau et précisez l'encadrement.