

## Référentiel, Compétences

### Lycée :

- Mesurer une grandeur physique à l'aide d'un capteur.
- Utiliser un dispositif comportant un microcontrôleur.

### Lycée Professionnel :

- Variabilité de la mesure d'une grandeur physique.
- Vitesse de propagation.

### Compétences :

- **S'approprier** : Représenter une situation par un schéma.
- **Analyser Raisonner** : Proposer une stratégie de résolution.
- **Réaliser** : Mettre en œuvre des algorithmes.
- **Valider** : Proposer d'éventuelles améliorations d'une démarche.
- **Communiquer** : Expliquer une démarche.

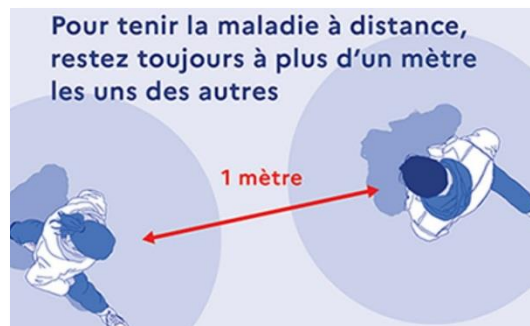
## Situation déclenchante

La consigne de l'OMS sur les gestes barrières semble issue d'un postulat historique, en place depuis 1930 par le médecin William F. Wells.

Il estime que la charge infectieuse des gouttelettes infectées atteint avec une grande difficulté la distance d'un mètre. Lors de la pandémie du Covid19, une entreprise française (société de Sophia-Antipolis) a mis au point un bracelet au poignet qui va vous avertir et clignoter si la distance barrière n'est pas respectée.

### Comment fonctionne ce dispositif ?

**Peut-on reproduire le même dispositif en utilisant la carte BBC micro:bit et la calculatrice TI-83 Premium CE Edition Python, en association avec quelques capteurs ?**



## Problématique

Quel est le principe d'un télémètre ?  
Comment mesure-t-on une distance avec la carte BBC micro:bit ?  
Quelle est la nature de l'onde utilisée dans un tel dispositif ?  
Comment générer une alerte sonore ou lumineuse en cas de proximité avec un individu ?

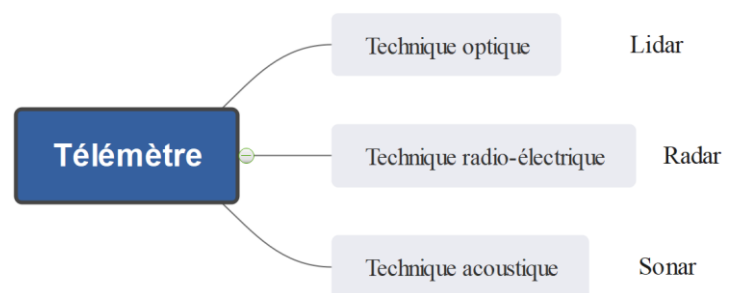
### Matériel nécessaire

- Une calculatrice TI-83 Premium CE Edition Python.
- Une carte BBC micro:bit.
- Un câble microUsb-miniUsb.
- 3 capteurs Grove « Ranger ».
- Une carte d'extension (Shield Grove).
- Un buzzer Grove.

### Principe d'un télémètre

Outil d'une grande simplicité mais très performant, le télémètre est l'accessoire de mesure indispensable pour chaque professionnel ayant besoin de mesurer une distance.

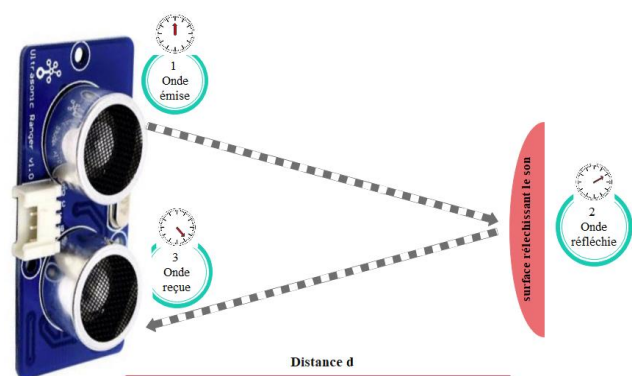
Il existe plusieurs types de télémètres selon la technique utilisée :



### Notion d'ultrasons et principe d'écholocalisation

Le capteur utilisé dans cette fiche est à base d'ultrasons. Il mesure la distance le séparant de l'obstacle. Il fonctionne par émission d'une salve d'ultrasons (paquet d'onde de courte durée, émis à intervalles égaux de temps) et détection de l'écho en retour.

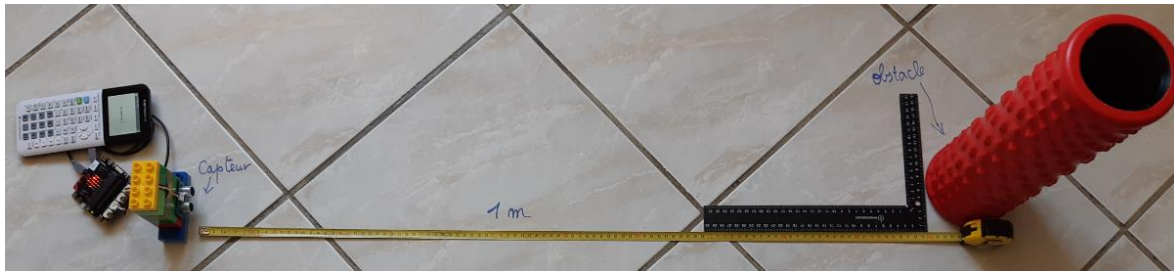
Le principe utilisé est celui de la propagation d'ondes ultrasonores d'une fréquence de 40 kHz dans l'air. La vitesse  $v$  du son dans l'air dépend de la température. Elle est de l'ordre de 340 m/s à 20°C. Une description plus détaillée est abordée dans la fiche « vitesse du son ».



## Mesurer une distance

### Mesure de distance

Réaliser le montage suivant (en connectant le ranger au pin0 de la carte Shield).



Le script permet d'afficher **la distance mesurée en cm** par le capteur à ultrasons.

La disponibilité de quelques commandes du module **grove** rend la réalisation de ce script très facile.

Chargement des modules nécessaires :

- **microbit** pour le fonctionnement de la carte.
- **ti\_system**
- **mb\_ti\_ptotlib** pour les représentations graphiques.
- **mb\_grove** pour la gestion des capteurs grove.
- Création d'une fonction **dist**.
  - Boucle **Tant que** (la touche **annul** n'est pas pressée)
  - Lecture de la valeur de la distance et affectation à la variable **d**.
  - Affichage de la valeur lue avec une décimale au centre de l'écran.

```
ÉDITEUR : TELEMET1
LIGNE DU SCRIPT 0001
from ti_system import *
import ti_plotlib as plt
from microbit import *
from mb_grove import *
def dist():
    while not escape():
        d=grove.read_ranger_cm(pin0)

        msg="DISTANCE=%.1f"%d+"cm"
        plt.cls()
        plt.text_at(7,msg,"center")
        sleep(2500)
```

Fns... | a A # | Outils | Exéc | Script

```
PYTHON SHELL

DISTANCE=99.0cm
```

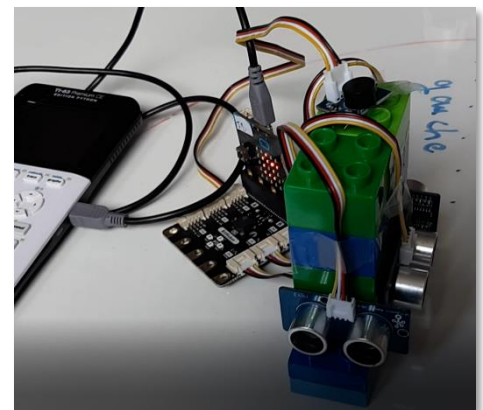
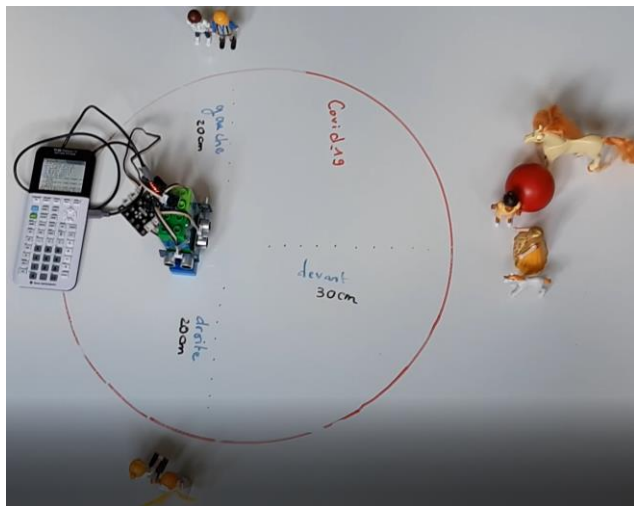
### Application

#### Distance barrière –Covid19

Le dispositif qui simulera un bouclier virtuel pour respecter la distance barrière d'un mètre, est constitué de 3 capteurs « ranger » et d'un buzzer afin d'avertir du non-respect de la distanciation, quand la distance mesurée est inférieure à une distance « seuil » fixée à l'avance.



# Mesurer une distance



## Algorithme

```

valeur seuil 1 ← valeur #Seuil à choisir
valeur seuil 2 ← valeur #Seuil à choisir
valeur seuil 3 ← valeur #Seuil à choisir
Tant que
  dd ← distance lue par le ranger sur le port 0 #distance à droite
  dg ← distance lue par le ranger sur le port 1 #distance à gauche
  df ← distance lue par le ranger sur le port 2 #distance en face

  Si dd < valeur seuil 1 ou dg < valeur seuil 2 ou df < valeur seuil 3 alors
    Envoi d'un signal sonore (pin8)
  Fin si
Fin Tant que
```



## Mesurer une distance



A. Yazi

Chargement des modules nécessaires :

- `microbit` pour le fonctionnement de la carte.
- `ti_plotlib` pour les représentations graphiques.
- `mb_pins` connexion aux ports.
- `mb_grove` pour gérer les capteurs grove.
- **Création de la fonction `alarme_covid`.**
- Boucle **Tant que** (la touche **annul** n'est pas pressée)
  - Lecture des 3 distances et leurs affectations à la variable `dd`, `dg` et `df`.
  - Affichage des valeurs lues avec une décimale.
  - Si l'une des 3 distances lues est inférieure à la distance seuil « de votre choix ».
  - Alors envoyer un signal sonore à partir du port 8.
  - `plt.cls()` : effacer l'écran.
  - `plt.text_at` : afficher un message et sa position.

```

ÉDITEUR : COVID19
LIGNE DU SCRIPT 0001
from ti_system import *
import ti_plotlib as plt
from microbit import *
from mb_grove import *
from mb_pins import *
#dd:droite dg:gauche df:en face
def alarme_covid(a,b,c):
  while not escape():
    dd=grove.read_ranger_cm(pin0
    )
    dg=grove.read_ranger_cm(pin1
    )
    df=grove.read_ranger_cm(pin2
    )
    if dd<a or dg<b or df<c :
      pin8.write_digital(1)
      msg1="DROITE=%.1f"%dd+"cm"
      msg2="GAUCHE=%.1f"%dg+"cm"
      msg3="FACE=%.1f"%df+"cm"
      plt.cls()
      plt.text_at(7,msg1,"center")

      plt.text_at(9,msg2,"center")

      plt.text_at(11,msg3,"center"
      )
      sleep(2500)

```

Fns... a A # |Outils| Exéc |Script

PYTHON SHELL

DROITE=155.4cm  
GAUCHE=167.1cm  
FACE=81.4cm

## Prolongement 1

## Effet de la température

Un télémètre à ultrasons n'est pas un outil précis en le comparant à un télémètre laser.

Cette mesure a été réalisée en supposant que la vitesse du son est de  $343 \text{ m} \cdot \text{s}^{-1}$ .

En réalité la vitesse d'une onde sonore est aussi influencée par la température du milieu traversé. La célérité a la propriété d'augmenter avec la température du milieu. Si  $T$  désigne la température du milieu, la célérité des ondes sonores dans ce milieu est approximativement égale à vingt fois la racine carrée de la température.

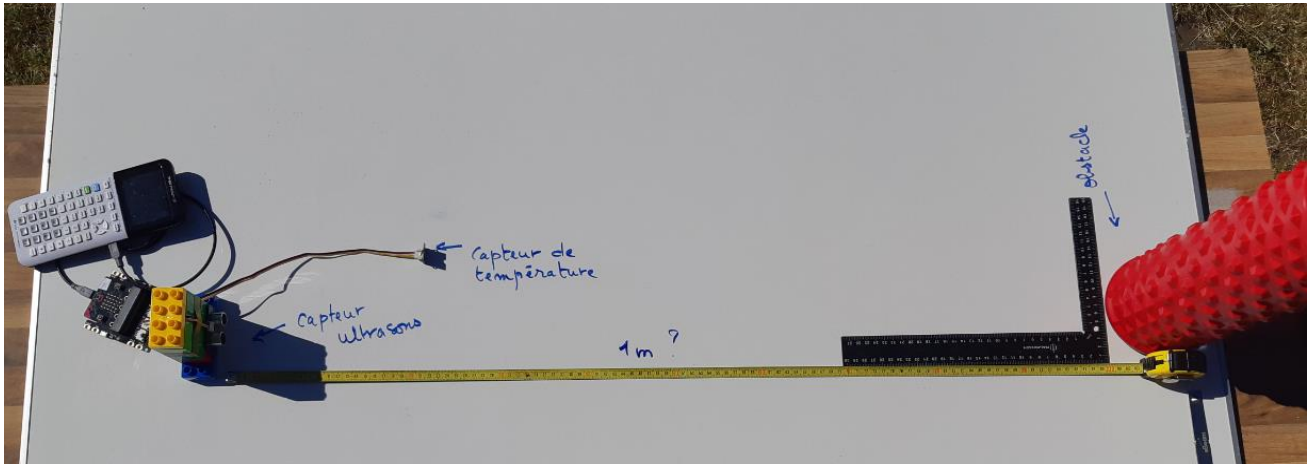
$$v = 20 \cdot \sqrt{T + 273,15} \quad T \text{ en } (^\circ\text{C})$$





## Mesurer une distance

On refait la même expérience en utilisant cette approximation et en insérant un capteur de température au montage précédent (au pin1 par exemple) :



Cette expérience est réalisée au mois de juillet (une journée où il fait 29°C)

**Remarque** : L'objectif de cette expérience est de sensibiliser les élèves sur le fait que la mesure d'une grandeur physique peut être influencée par d'autres paramètres (ici la température).

La température du milieu est mesurée grâce au capteur ci-contre



Chargement des modules nécessaires :

- `microbit` pour le fonctionnement de la carte.
- `ti_system`.
- `mb_ti_plotlib` pour les représentations graphiques.
- `mb_grove` pour la gestion des capteurs grove.

• **Création d'une fonction temp.**

- Lecture de la valeur de la température et affectation à la variable T.
- Renvoi la valeur lue.

• **Création d'une fonction dist.**

- Boucle **Tant que** (la touche **annul** n'est pas pressée)
  - Lecture de la valeur de la distance et affectation à la variable d.
  - Correction de la distance d en tenant compte du paramètre température T (argument de la fonction et fourni par la fonction temp) et affectation à la variable dT.
  - Affichage des valeurs des 3 variables avec une décimale.

```

ÉDITEUR : TELEMET2
LIGNE DU SCRIPT 0001
from math import *
from ti_system import *
import ti_plotlib as plt
from microbit import *
from mb_grove import *
from mb_pins import *
#Mesure de temperature
def temp():
    T=grove.read_temperature(pin1)
    return T
#Mesure de la distance
def dist(T):
    while not escape():
        d=grove.read_ranger_cm(pin0)
        dT=d*20/342*sqrt(T+273.15)
        msg1="DISTANCE=%.1f"%d
        msg2="TEMPERAT=%.1f"%T
        msg3="DISTANCE corrig=%.1f"%dT
        plt.cls()
        plt.text_at(7,msg1,"center")
        plt.text_at(9,msg2,"center")
        plt.text_at(11,msg3,"center")
    sleep(2500)
    
```



## Mesurer une distance



A. Yazici

La distance mesurée en prenant compte du paramètre température apporte la correction attendue.

$$v = 20 \cdot \sqrt{T + 273,15}$$

Aide pour le calcul de la distance corrigée :

$$d_{\text{corrigé}} = d_{\text{mesurée}} \times \frac{20 \times \sqrt{T + 273.15}}{342}$$

```
PYTHON SHELL

DISTANCE=98.4
TEMPERAT=28.5
DISTANCE corrig=99.9
```

## Prolongement 2

### Radars de recul

Le dispositif peut être adapté à un radar de recul, en remplaçant le buzzer avec un haut-parleur et en sollicitant le module music afin de générer différents signaux sonores en fonction du rapprochement de l'obstacle.



Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus !

