Célérité d'une onde ultrasonore
Application à la mesure de distance

Mesure de la vitesse d'un signal sonore dans l'air

Le son est une onde sonore qui se propage dans un milieu matériel. Vous souhaitez estimer sa vitesse de déplacement dans l'air.

Document 1 : Utilisation du capteur ultrasons HC-SR04

Le capteur HC-SR04 est un émetteur-récepteur à ultrasons. On s'en sert communément pour évaluer la vitesse de propagation d'un signal sonore ou pour mesurer des distances.
Lorsque le signal lui est donné, l'émetteur envoie une salve ultrasonore qui se propage en ligne droite jusqu'à être réfléchie par un obstacle. Elle revient alors vers le récepteur : c'est le phénomène d'écho. Un traitement approprié, à l'aide d'un microcontrôleur par exemple, permet de déterminer la durée de cet aller-retour.

Le capteur HC-SR04 fonctionne avec une tension d'alimentation de 3,3 V et permet d'effectuer des mesures entre 2 cm et 350 cm.

Document 2 : Utilisation du microcontrôleur pilotant le capteur HC-SR04

Le capteur HC-SR04 est piloté par le microcontrôleur : c'est ce dernier qui lui donne des instructions et qui récupère les données mesurées.
Un exemple de programme Python (à remettre dans l'ordre et à compléter) est proposé en annexe. Téléversé vers le microcontrôleur à l'aide d'un logiciel adéquat (tel que Mu), il permet l'affichage de la durée du parcours de la salve ultrasonore (en microsecondes) dans la console REPL du logiciel.

Document 3 : Matériel à disposition

- 1 carte micro:bit et ses accessoires
- 1 capteur à ultrasons HC-SR04
- 1 grande règle graduée
- le programme présenté en annexe
- le logiciel Mu

1. À l'aide des informations et du matériel énuméré, établir un protocole expérimental permettant de déterminer la vitesse d'un signal sonore ou ultrasonore dans l'air.
2. En utilisant l'annexe (à découper), remettre dans l'ordre toutes les étapes du programme à téléverser vers le microcontrôleur.
3. Compléter les parties manquantes du programme (repérées par des pointillés).
4. Saisir le programme à l'aide du logiciel Mu puis le téléverser vers le microcontrôleur.

Mettre en œuvre le protocole.

5. Exploiter la mesure de durée effectuée par le microcontrôleur afin de déterminer la vitesse de propagation du signal sonore dans l'air.
6. Discuter de la valeur obtenue en la comparant avec la valeur théorique donnée par l'expression suivante :

$$c = 332 + 0,606 \cdot \vartheta\,(°C)$$

Prolongement : application à la réalisation d'un télémètre

On souhaite maintenant utiliser le capteur à ultrasons HC-SR04 comme télémètre, la vitesse de propagation du son dans l'air étant connue, dans le cadre de la réalisation d'un radar de recul.

1. Modifier le programme précédent afin qu'il retourne la distance (en cm) séparant le capteur de l'obstacle placé devant lui.
2. Téléverser le programme vers le microcontrôleur puis effectuer quelques mesures de distances.
3. Pour simuler l'avertisseur sonore présent sur tout radar de recul, il convient d'ajouter au programme les instructions nécessaires pour le déclenchement d'un signal sonore sous condition de distance.
   On utilisera le buzzer fourni et on pourra par exemple moduler la fréquence des « bips » en fonction de la distance entre l'émetteur/récepteur et l'obstacle.
   La mise en route du radar s'effectuera par appui sur le bouton A du microcontrôleur et l'arrêt par appui sur le bouton B. Une fois en marche, la mesure de distance s'effectuera en continu.

Annexe : Programme Python permettant la mesure et l'affichage de la durée du parcours d'une salve US après écho

```
    # ......
    while not button_a.was_pressed():
        display.show(Image.ARROW_W)
```

```
# Importation de tous les modules de la bibliothèque micro:bit
from microbit import *
# Importation du module time (fournit différentes fonctions liées au temps)
import time
```

```
    # Si le bouton A est pressé, lancement de la mesure et affichage de la durée de
    ↪ parcours
    duree = duree_parcours_us(pin0)
    display.show(Image.YES, clear=True)
    print("Duree : {} microsecondes".format(duree))
```

```
# Exécution du blocs d'instructions qui suit en boucle
while True:
```

```
# Définition de la fonction déclenchant l'émission d'une salve et la mesure de la durée
↪ du parcours
def duree_parcours_us(pin, timeout = 1000000):
    pin.write_digital(0)
    time.sleep_us(2)
    pin.write_digital(1)
    time.sleep_us(5)
    pin.write_digital(0)
    t_init = time.ticks_us()
    while (pin.read_digital() != 1):
        if(time.ticks_us() - t_init > timeout):
            return 0
    start = time.ticks_us()        # Date d'émission de la salve US
    while (pin.read_digital() == 1):
        if(time.ticks_us() - t_init > timeout):
            return 0
    end = time.ticks_us()        # Date de réception de la salve US
    return ......        # La fonction retourne la durée du parcours de la salve
```