

Activité Objets Connectés

Radar de recul

Objectif : Réaliser un dispositif d'aide au stationnement basé sur une carte micro:bit

1 Conventions

Consignes sur le travail à effectuer

Données

Ressources

2 Prérequis

- Programmation de base en python : variables, opérateurs, fonctions, boucles,
- Utilisation de la carte micro-bit : gestion des Inputs/Outputs
- Utilisation du logiciel mu-editor : gestion librairie, console REPL

3 Nouvelles notions à acquérir

- Utilisation des ultrasons
- Acquisition de données (temps, température)
- Traitement de données (calcul de vitesse, de distance, actions conditionnelles, ..)

4 Descriptif du contexte

Le radar de recul : Le véhicule est doté de capteurs qui émettent des ultrasons. Lorsque ceux-ci rencontrent un obstacle, tel qu'un piéton, un mur ou encore un trottoir, ils reviennent aux capteurs. Un calculateur évalue les données et émet un signal d'alerte à intervalles plus ou moins longs indiquant la distance de l'obstacle.



5 Utilisation de la carte micro-bit

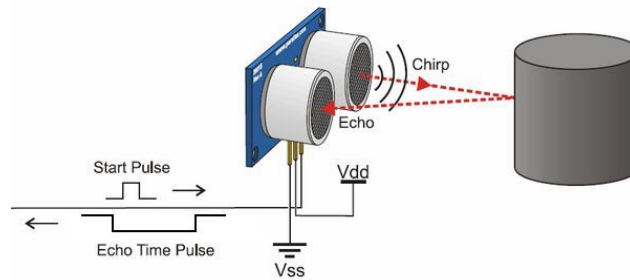
Pour réaliser ce dispositif, on utilise

- Une carte micro:bit
- Un shield Grove (interface pour connecter les capteurs et actionneurs)
- Un capteur à ultrasons (Module Grove - Ultrasonic Ranger)
- Un capteur de température (Module Grove - Temperature Sensor)
- Un "haut parleur" (Module Grove – Speaker)

Le programme nécessite la librairie micro-grove.py mise à votre disposition par votre professeur

6 Mesure d'une distance

1. Principe du capteur à ultrasons



Un capteur à ultrasons émet à intervalles réguliers de courtes impulsions sonores à haute fréquence. Ces impulsions se propagent dans l'air à la vitesse du son. Lorsqu'elles rencontrent un objet, elles se réfléchissent et reviennent sous forme d'écho au capteur.

Au début de l'émission une broche du capteur est mise à l'état bas (0v) et repasse à l'état haut (3,3v) lorsque le capteur reçoit l'écho. En mesurant le temps pendant lequel la broche a été à l'état bas, on obtient le **temps que les ultrasons ont mis pour faire un aller retour**, et on peut en déduire la distance par calcul.

On sait que :

La vitesse de propagation des ultrasons = 340 m/s

La distance parcourue par les ultrasons = vitesse de propagation x temps

En notant :

D : la distance entre le capteur à ultrasons et l'objet en mètres

V : la vitesse de propagation en mètres/seconde

t : le temps en secondes

Etablir l'expression de **D** en fonction de **V** et **t**

$D =$

Le capteur fournit le temps en microsecondes (10^{-6} s).

Calculer la nouvelle expression de D

$D =$

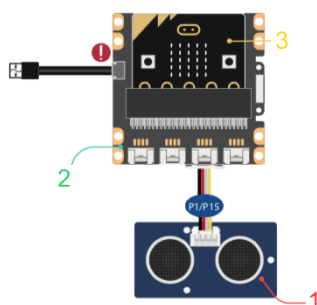
On souhaite obtenir la distance en centimètre (10^{-2} m).

Calculer la nouvelle expression de D

$D =$

2. Utilisation de la carte microbit

a) Effectuer les branchements comme indiqué ci-dessous



1 : capteur ultrasons sur la broche pin1

2 : shield Grove

3 : carte micro:bit

Avec le logiciel mu-editor, saisir le programme suivant :

```
from microbit import *
from micro_grove import *
from math import *

while True:
```

Dans la librairie micro_grove.py , on dispose de la fonction **mesure_temps_A_R(pin)** qui retourne le temps en microsecondes mis par les ultrasons pour faire un aller-retour.

Compléter le programme pour que :

La mesure du temps soit mise dans une variable **tps**

Le calcul de la distance en cm (voir §6.1) avec $V=340$ soit mis dans une variable **d**

La valeur de **d** s'affiche dans une console REPL

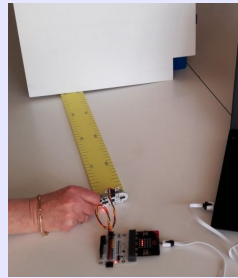
Faire constater le bon fonctionnement

Rappels :

La librairie micro-grove.py doit être copiée sur la carte micro:bit après avoir flashé le programme dans la carte micro:bit, en utilisant le menu "Fichiers" de mu-editor

Pour afficher le résultat de la fonction print(), cliquer sur REPL dans mu-editor et faire CTRL+D

Contrôler la cohérence de vos résultats en mesurant une distance connue



3. Fiabilisation du dispositif

On a supposé que la vitesse de propagation du son était constante. En réalité, celle-ci dépend de la température T en $^{\circ}\text{C}$ et est obtenue par la relation

$$V_{\text{air}} = 20,05 * \sqrt{T + 273,15}$$

Le radar de recul étant utilisé à l'extérieur, il doit être fiable pour des températures allant de -30°C à 45°C .

Pour estimer l'erreur comise quand on prend $V=\text{constante}=340$ m/s, on va comparer les distances obtenues par calcul à -30°C et 45°C pour **tps = 3000 microsecondes**

En utilisant l'expression ci-dessus :

Calculer V_{air} que l'on notera V_{-30} pour $T = -30^{\circ}\text{C} \rightarrow V_{-30} = \dots\dots\dots$

Calculer V_{air} que l'on notera V_{45} pour $T = 45^{\circ}\text{C} \rightarrow V_{45} = \dots\dots\dots$

En utilisant l'expression obtenue au §6.1,

Calculer la distance que l'on notera D_{340} avec $V=340$ m/s $\rightarrow D_{340} = \dots\dots\dots$

Calculer la distance que l'on notera D_{-30} avec $V= V_{-30} \rightarrow D_{-30} = \dots\dots\dots$

Calculer la distance que l'on notera D_{45} avec $V= V_{45} \rightarrow D_{45} = \dots\dots\dots$

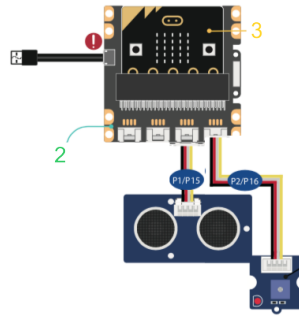
Conclusion 1 :

Conclusion 2 : Dans le programme, pour calculer la distance, il ne faut pas utiliser 340 m/s mais V_{air} . Ceci nécessite donc d'ajouter un capteur de température pour calculer V_{air} .

7 Mesure de température

1. Branchements

Ajouter le capteur de température sur la broche pin2 comme ci-dessous



2. Programmation

Dans la librairie `micro_grove.py`, on dispose de la fonction `mesure_temperatures(pin)` qui retourne la température du capteur.

Compléter le programme pour que :

- La mesure la température soit mise dans une variable **T**
- Le calcul de V_{air} soit mis dans une variable **V**
- Le calcul de la distance en cm prenant en compte **V** soit mis dans une variable **d**
- Les valeurs de **d** et **T** s'affichent dans une console REPL

Vérifier la cohérence de vos résultats

Vérifier que la mesure de distance est plus précise

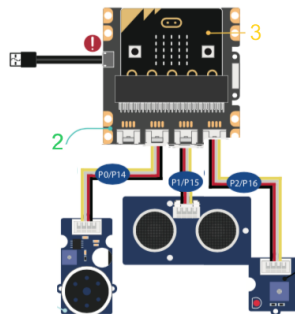
Pour vérifier que la compensation en température fonctionne bien, il faudrait effectuer des mesures à des températures très différentes, ce qui est difficile à réaliser dans une salle de classe.

On peut néanmoins constater que pour tps constant, si on augmente la température du capteur (en le plaçant dans sa main fermée) la vitesse de propagation affichée augmente

8 Signal d'alerte

1. Branchement

Ajouter le "haut parleur" sur la broche pin0 comme ci-dessous



2. Programmation

Dans la librairie `micro_grove.py`, on dispose de la fonction `bip(pin)` qui émet un bip sonore sur le haut parleur

Compléter le programme pour émettre un bip :

- toutes les secondes si la distance est comprise entre 100cm et 160cm (exclu)
- toutes les 1/2 seconde si la distance est comprise entre 50cm et 100cm (exclu)
- toutes les 100ms si la distance est comprise entre 30cm et 50cm (exclu)
- toutes les 10ms si la distance est inférieure à 30cm (exclu)

Faire constater le bon fonctionnement

```
# sleep for the given number of milliseconds.
sleep(ms)
```