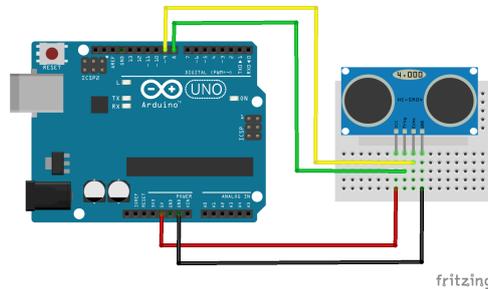


6 À vous de jouer

6.1 Mesure de la vitesse du son

6.1.1 Le capteur ultrason



6.1.2 La réalisation

Les objectifs

Réaliser le traitement d'un flux de données provenant d'un capteur ultrason afin de déterminer la vitesse du son. Le contrôle de l'acquisition est effectué avec une carte *Arduino UNO*, le traitement et la modélisation sont réalisés avec le langage de programmation Python. Les étapes essentielles de ce projet sont :

- Écrire le programme Arduino pour l'acquisition des données.
- Récupérer avec Python les mesures de temps pour des distances comprises entre 10 cm et 50 cm.
- Tracer avec Python le graphique de la distance en fonction du temps.
- Sauvegarder les données (temps, distance) dans un fichier CSV.

Document n° 1 : Le code Arduino et Python

```

1 // Déclaration des variables globales : broches
2
3 // Broche TRIGGER
4 // Broche ECHO
5
6 void setup() {
7
8   pinMode(trigg, OUTPUT); // Configuration des broches
9   digitalWrite(trigg, LOW); // La broche TRIGGER doit être à LOW au repos
10  pinMode(echo, INPUT); // La broche ECHO en entrée
11
12  // À compléter // Démarrage de la liaison série
13 }
14
15 void loop() {
16
17   digitalWrite(trigg, HIGH); // Lance une mesure de distance en envoyant
18   delayMicroseconds(10); // une impulsion HIGH de 10 microsecondes
19   digitalWrite(trigg, LOW);
20
21   temps = pulseIn(echo, HIGH); // Mesure le temps en microseconde entre
22   // l'envoi de l'ultrason et sa réception
23
24   // À compléter // Les résultats sur le port série
25   // On fait une pause
26 }

```

Un petit bout de programme Python pour tester la fonction `input` et donner des idées sur la manière de gérer le flux de données envoyé par la carte Arduino.

```

1 mesure = float(input("Entrez votre mesure : "))
2 while mesure != -1:
3     serial_port.flushInput()
4     print(serial_port.readline())
5     mesure = float(input("Entrez votre mesure : "))

```

Document n° 2 : La fonction `input()`

Il est souvent utile que l'utilisateur puisse entrer des données au clavier. La méthode la plus simple consiste à employer la fonction native de Python : `input()`. Cette fonction provoque une interruption dans le programme courant. L'utilisateur est invité à entrer des caractères au clavier et à valider en appuyant sur <Enter>. Lorsque cette touche est enfoncée, l'exécution du programme se poursuit, et la fonction fournit en retour une chaîne de caractères correspondant à ce que l'utilisateur a entré. Cette chaîne peut alors être référencée par un nom de variable quelconque.

On peut invoquer la fonction `input()` en laissant les parenthèses vides. On peut aussi y placer en argument un message destiné à l'utilisateur.

Attention le nom de cette fonction dépend de votre version de Python

- Python 2.7 : `raw_input()`
- Python 3.x : `input()`

Document n° 3 : Caractéristiques techniques du module HC-SR04

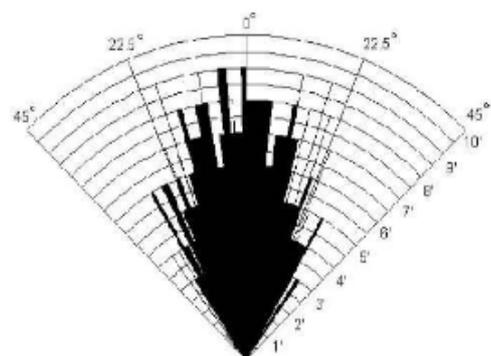
Le capteur HC-SR04 utilise les ultrasons pour déterminer la distance d'un objet. Il offre une excellente plage de détection sans contact, avec des mesures de haute précision et stables. Son fonctionnement n'est pas influencé par la lumière du soleil ou des matériaux sombres, bien que des matériaux comme les vêtements puissent être difficiles à détecter.

Caractéristiques

- Dimensions : 45 mm x 20 mm x 15 mm
- Plage de mesure : 2 cm à 400 cm
- Résolution de la mesure : 0.3 cm
- Angle de mesure efficace : 15°
- Largeur d'impulsion sur l'entrée de déclenchement : 10 μ s (Trigger Input Pulse width)

Broches de connection

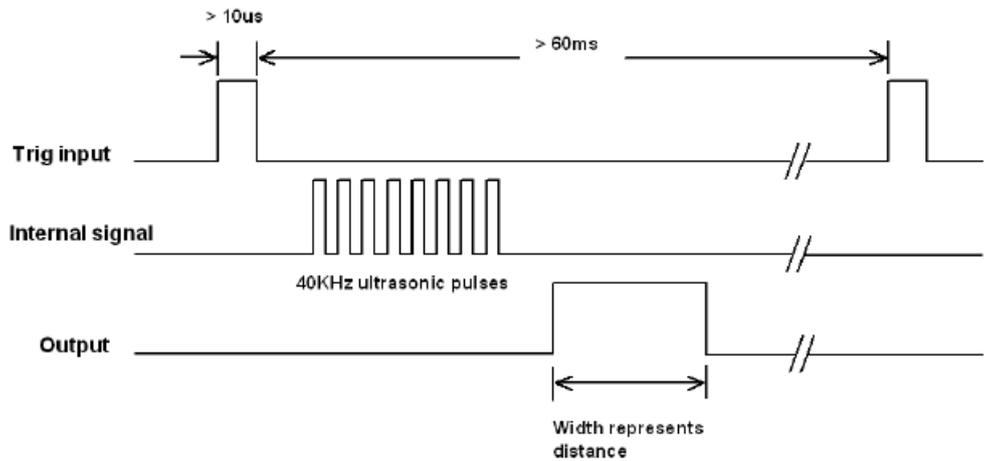
- Vcc = Alimentation +5 V DC
- Trig = Entrée de déclenchement de la mesure (Trigger input)
- Echo = Sortie de mesure donnée en écho (Echo output)
- GND = Masse de l'alimentation



*Practical test of performance,
Best in 30 degree angle*

Document n° 4 : Caractéristiques techniques du module HC-SR04

Pour déclencher une mesure, il faut une impulsion "high" (5V) d'au moins 10 μ s sur l'entrée "Trig". Le capteur émet alors une série de 8 impulsions ultrasoniques à 40 kHz, puis il attend le signal réfléchi. Lorsque celui-ci est détecté, il envoie un signal "high" sur la sortie "Echo", dont la durée est proportionnelle à la distance mesurée.

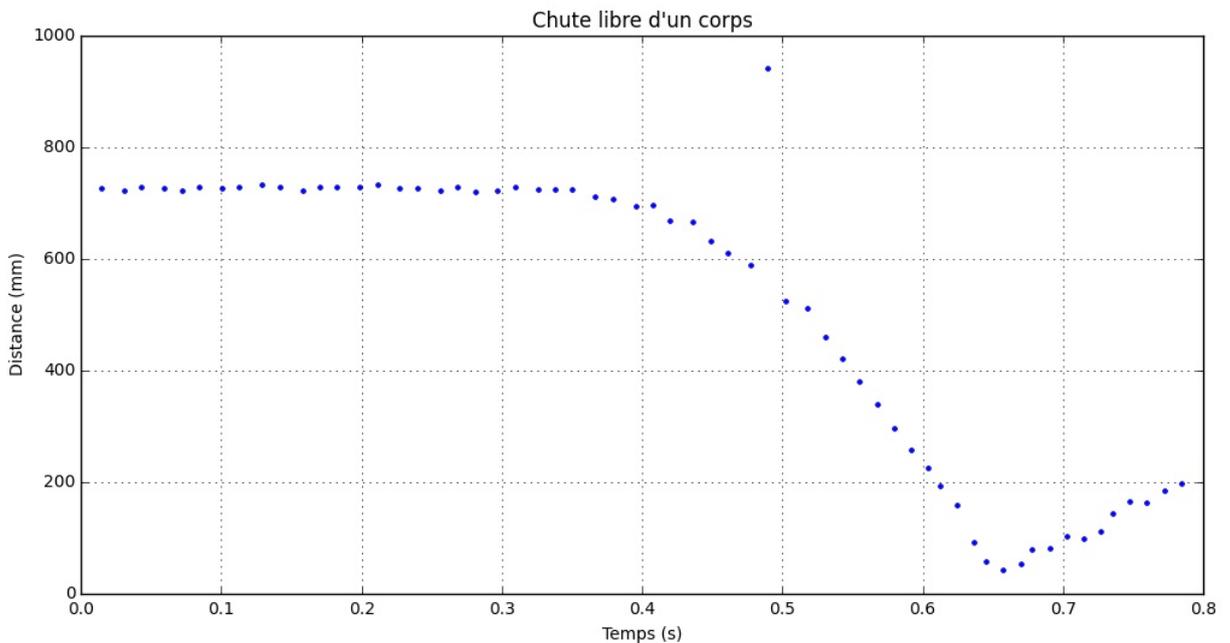


6.2 Des idées pour la suite...

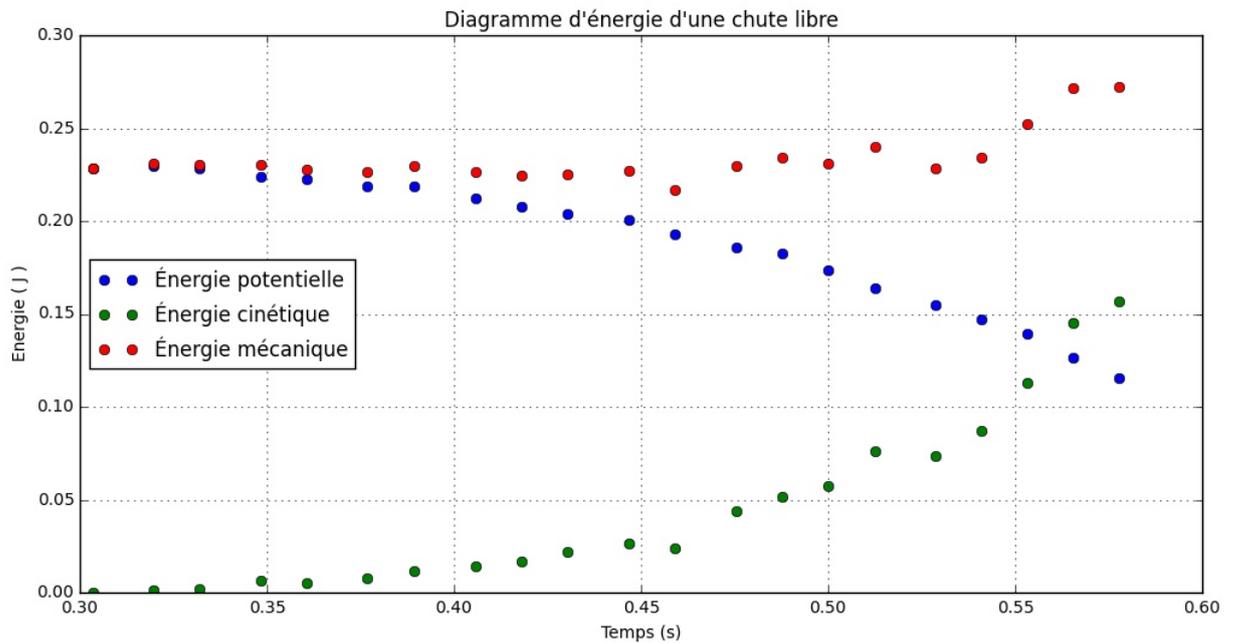
Afin de préparer la suite de la formation, je vous donne quelques idées de montage à réaliser avec la carte Arduino UNO en lien avec le programme de TS. Pour chaque montage je vous donne la représentation graphique des résultats que j'ai obtenus

6.2.1 Chute libre d'un corps sans vitesse initiale.

Avec le matériel dont vous disposez essayez d'imaginer un montage et son programme permettant d'obtenir la courbe représentative de la loi horaire de la chute libre d'un corps.

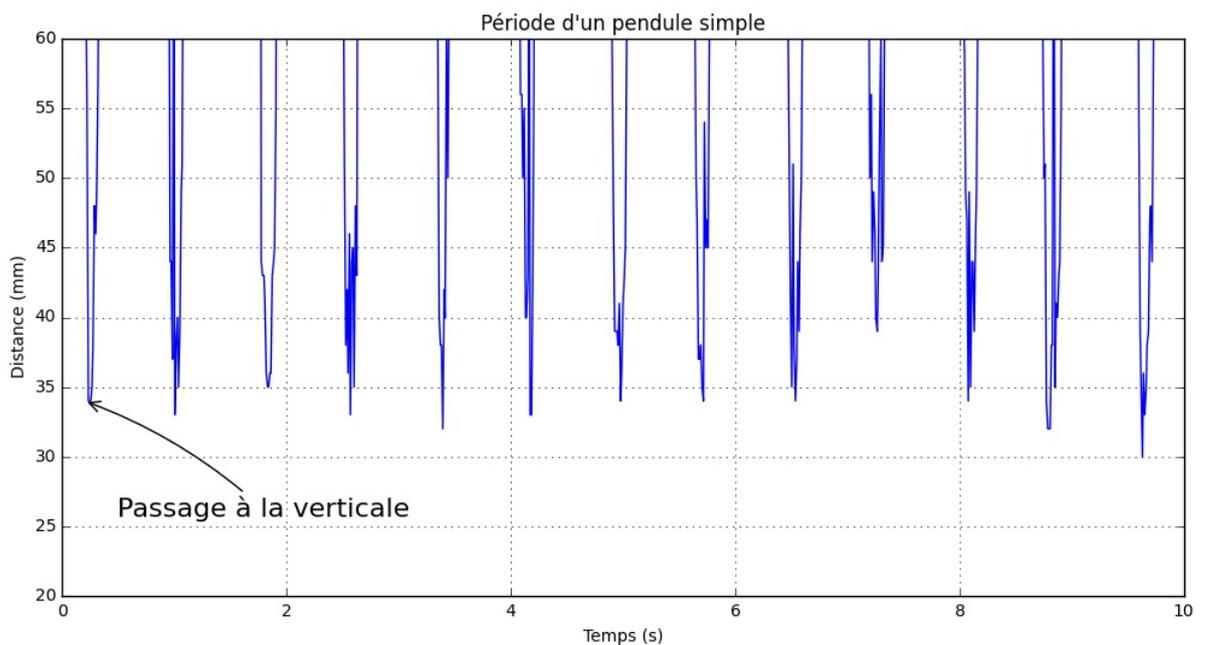


On pourra également tracer les courbes d'énergie cinétique, potentielle et mécanique.



6.2.2 Période d'un pendule simple.

Avec le matériel dont vous disposez essayer d'imaginer un montage et son programme permettant de mettre en évidence la période d'un pendule simple.



6.2.3 Oscillateur : système solide-ressort vertical.

Avec le matériel dont vous disposez essayer d'imaginer un montage et son programme permettant d'obtenir l'évolution temporelle de l'allogement d'un ressort.

