

Activité 1 : Aide au stationnement

Une petite carte **Arduino** peut gérer des moteurs, des systèmes d'affichage, des capteurs (accéléromètres, température, pression atmosphérique, luminosité, et la liste est loin d'être exhaustive)...D'accord ! Mais concrètement, on fait quoi avec ?

Je vous propose de réaliser une aide au stationnement que j'avais remarqué sur la chaîne vidéo « le labo d'Heliox ».

<https://www.youtube.com/watch?v=kXqyvjjYSE>

Mesurer une distance

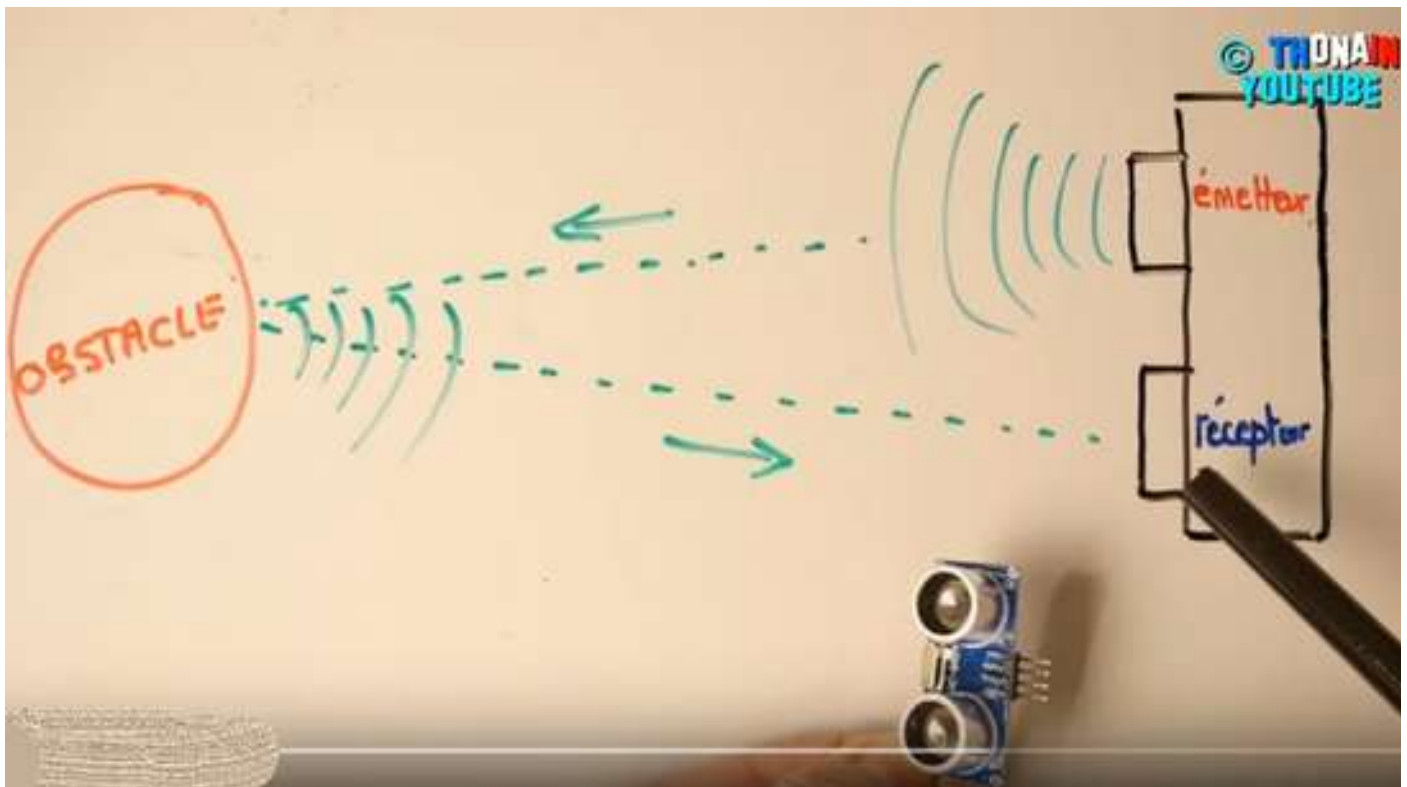
Webographie : <https://www.carnetdumaker.net/articles/mesurer-une-distance-avec-un-capteur-ultrason-hc-sr04-et-une-carte-arduino-genuino/>

Matériel :

- PC + logiciel Arduino
- Carte Arduino + câble USB
- Platine d'essai + câbles
- Capteur à ultrason **HC-SR04**



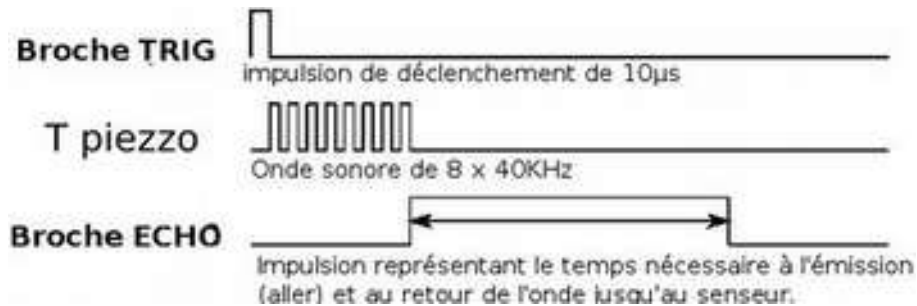
Le principe de fonctionnement du capteur est entièrement basé sur la **vitesse du son**.



Les ultrasons se propagent dans l'air à la vitesse $v = 340 \text{ m.s}^{-1}$.

Ouvrir le programme **activit__1_initiation_ARDUINO**

- 1) On met la broche **TRIGGER** à l'état haut (**HIGH** pendant **10 μs**).
- 2) L'émetteur envoie alors 8 impulsions ultrasoniques (40 kHz). La broche **ECHO** passe à l'état **HIGH**.
- 3) Les ultrasons se propagent dans l'air jusqu'à réfléchir sur l'obstacle.
- 4) Le récepteur **ECHO** détecte alors l'onde réfléchie, passe à l'état bas **LOW**.



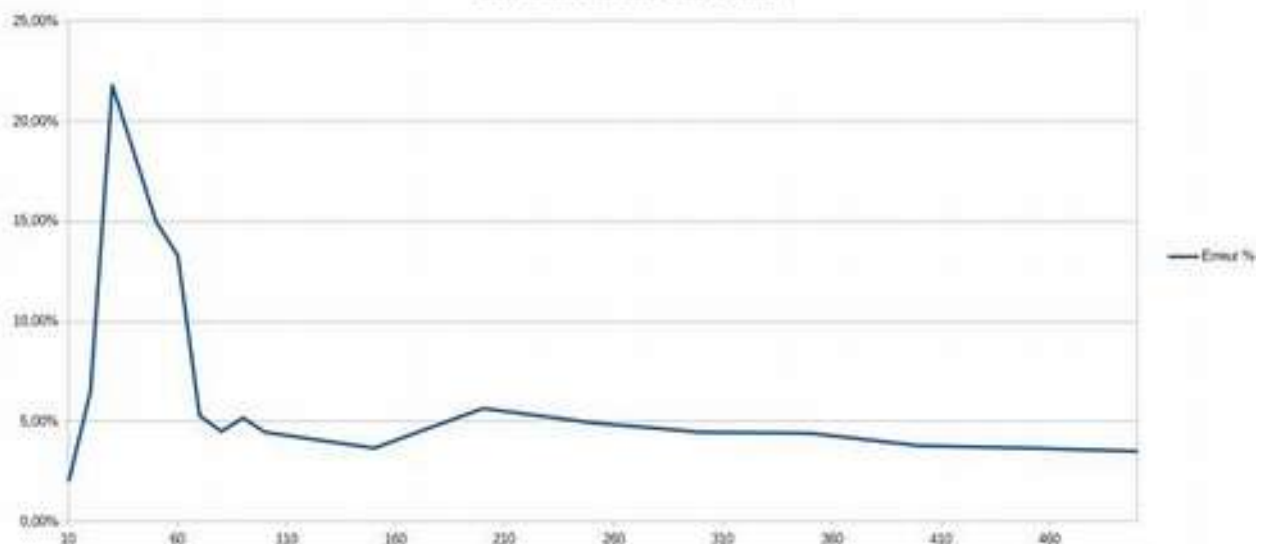
Il est possible d'écrire des commentaires dans le programme. Un commentaire commence par les 2 signes //.

Une instruction se finit toujours par un point-virgule ;

- Écrire la relation manquante reliant la **distance**, la **durée** et **vitesse** sachant que la durée est mesurée en microsecondes et que la vitesse est exprimée en mètre par seconde.
- Associer les commentaires proposés à une des lignes du programme téléchargé.
- Sauvegarder le programme et les commentaires.
- Téléverser le programme puis vérifier qu'il fonctionne correctement.
- Proposer un protocole expérimental pour valider la précision du capteur en étudiant l'écart relatif.



Erreur mesure HC-SR04 (page 1-50cm)



Graphique d'erreurs du capteur sur la plage 1-50cm

Activité 2 : Mesurer une distance et allumer un voyant lumineux

Consigne 1 :

- Brancher une **DEL ROUGE** en série avec une résistance de 100 Ω sur la sortie numérique D8.
- Brancher une **DEL VERTE** en série avec une résistance de 100 Ω sur la sortie numérique D9.

Aide :

Par exemple, une broche peut-être testée et une action A réalisée si la broche est inférieure à 500, et une autre action B réalisée si la broche est supérieure ou égale à 500.

Le code ressemblera à cela :

```
if (broche < 500)
{
  // action A
}
else
{
  // action B
}
```

A partir du programme initial, vous devez :

- Allumer la DEL verte uniquement si la distance mesurée est supérieure à 60 cm.
- Allumer la DEL rouge uniquement si la distance mesurée est inférieure à 60 cm

Consigne 2 :

NB : else peut contenir un autre test if, et donc des tests multiples.

A partir du programme initial, vous devez :

- Éteindre les 2 DEL si la distance mesurée est supérieure à 4,00 m
- Allumer la DEL verte uniquement si la distance mesurée est comprise entre 60 cm et 4,00 m.
- Allumer la DEL rouge uniquement si la distance mesurée est inférieure à 60 cm

Activité 3 : Mesurer une distance avec avertisseur sonore

Consigne :

- Brancher un buzzer polarisé sur la sortie numérique D8.

Aide : L'instruction *tone* (**broche** , **fréquence**) génère une **onde carrée** à la **fréquence** spécifiée sur une **broche**.

A partir du programme de l'activité 2, vous devez :

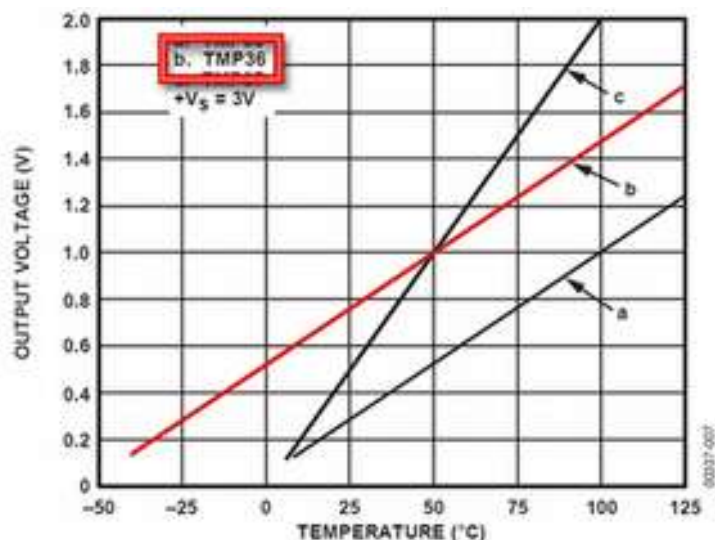
- Générer un son strident si la distance mesurée est inférieure à 60 cm

Activité 4 : Mesurer une température et une distance

Webographie : <https://zestedesavoir.com/tutoriels/343/realiser-un-telemetre-a-ultrasons/>

Le capteur choisi est le capteur **analogique actif** TMP36

- **Size:** TO-92 package (about 0.2" x 0.2" x 0.2") with three leads
- **Price:** \$1.50 at the Adafruit shop
- **Temperature range:** -40°C to 150°C / -40°F to 302°F
- **Output range:** 0.1V (-40°C) to 2.0V (150°C) but accuracy decreases after 125°C
- **Power supply:** 2.7V to 5.5V only, 0.05 mA current draw
- **Datasheet**



- Calculer la sensibilité S du **capteur de température TMP36** en $V/°C$
- Donner l'**équation mathématique (1)** reliant la température à la tension du capteur TMP 36.
- Relier les 2 pattes le TMP36 au +5V et au GROUND 0V.
- Relier la patte du milieu à l'**entrée analogique A1** de la carte arduino.

Consigne 1 :

- Utiliser les instructions ci-dessous pour afficher l'information reçue sur **l'entrée analogique A1**.

```
int brochetemp = 1;
int tension ;
void setup() {
    Serial.begin(9600);
}
void loop() {
    // On lit la tension analogique sur le bon port:
    tension = analogRead(brochetemp);
    Serial.print(tension);
    Serial.println(" volts");
}
```

- Faire varier la température et expliquer les informations que vous lisez sur le moniteur série

Vous ne lisez pas la tension du capteur en volts mais la tension numérisée qui est un entier qui va de **0 à 1023**. En effet la carte Arduino possède un **C.A.N** qui numérise sur 10 bits des tensions analogiques variant de **0 à 5V**. Le pas de ce C.A.N. est 5/1024.

- Rajouter l'instruction `float tensioncapt = (tension * 5) / 1024 ;` dans le corps du void loop.
- Afficher la tension mesurée par le capteur dans le moniteur série.
- Compléter enfin le programme pour afficher la température (°C) en fonction de tensioncapt en utilisant l'équation mathématique (1) dans le moniteur série.

Aide/réponse : https://medspx.fr/blog/Blog/arduino_temp%C3%A9rature/

Consigne 2 :

La température a un impact sur la vitesse des ondes sonores.

La relation est $v=a\times\theta+b$ avec v la vitesse en mètres par seconde et θ la température en degrés Celsius. $A = 0,6$ et $b = 331,5$.

Une amélioration de notre télémètre installé dans un garage serait donc d'avoir une correction de la vitesse via une mesure de la température.

- Remplir le programme par les instructions associées aux commentaires.

```
void loop() {  
    // On fait la mesure de température  
    // On stocke la température en degrés Celsius dans "degres".  
    float degres;  
    // puis on ajuste la vitesse en fonction de la température  
float vitesse = 0.6*degres + 331.5;  
    // et enfin, on fait tout le reste comme avant (mesure + affichage)  
    // La seule différence sera que l'on utilisera la variable "vitesse" ...  
    // ... au lieu de la constante VITESSE  
}
```

- Améliorer le programme initial pour mesurer une distance en fonction de la température de l'air.

Activité 5 : Aide au stationnement

A votre tour de réfléchir pour concevoir une aide au stationnement avec un témoin lumineux qui clignote lorsqu'on arrive à 60 cm du mur et qui s'éteint 15 secondes après l'arrêt du véhicule. Bonne chance !

Vous pouvez demander l'aide des collègues de NSI...