

# Réaliser un télémètre à ultrasons

Lorsque l’on veut mesurer des distances, on utilise en général une règle ou un ruban mètre. Mais comment faire pour avoir une distance numérique en temps réel ?

Une réponse : utiliser la technologie des ultrasons pour réaliser un **télémètre**.

Bien que peu précis à l’échelle des centimètres, ces derniers sont relativement fiables à l’échelle de la dizaine de centimètres. Les ultrasons étant des ondes sonores se propageant sous la forme d’un cône (ils sont peu directifs), ils font de très bons détecteurs d’obstacles.

Nous allons donc nous lancer dans la réalisation d’un petit télémètre à ultrasons via une Arduino et avec un affichage intégré sur un écran LCD  :

# [Rappel sur les ultrasons](https://zestedesavoir.com/tutoriels/343/realiser-un-telemetre-a-ultrasons/#1-10799_rappel-sur-les-ultrasons)

Faisons un peu de sciences et rappelons ce que sont des ultrasons.

Un ultrason est une onde sonore à haute fréquence. Par haute fréquence j’entends toutes les fréquences sonores inaudibles pour l’oreille humaine, soit celles au-delà de 20 kHz. Elles sont l’opposé des *infrasons* qui sont les ondes sonores dont la fréquence est inférieure à la plus faible audible pour l’Homme et qui est de 20 Hz.

Bon, c’est bien beau mais cette histoire de fréquence ça ne nous en dit pas beaucoup plus !

### Une onde sonore c’est quoi ?

Une onde sonore est un phénomène physique de compression et décompression. Lorsqu’une vibration est produite (par n’importe quel objet qui vibre), l’air subit alors une onde de choc qui se traduit en mouvement des atomes. Il y a alors ce phénomène de compression et décompression (des "trous" dans l’air) que des récepteurs dans nos oreilles convertissent en bruit.

Plus les compressions sont proches et plus la fréquence est élevée. On parle alors de son aigu. Au contraire, plus les compressions sont éloignées et plus la fréquence est faible, on parle d’un son grave. Une fréquence s’exprime en Hertz et traduit la répétition d’un motif d’un phénomène durant une seconde. Par exemple si je cligne des yeux trois fois par seconde, on peut dire que je cligne des yeux à 3 Hz.

Dans le cas des ultrasons, les compressions/décompressions sont très courtes. En effet, le motif se répète plus de 20 000 fois par seconde, donc à plus de 20 kHz. En général, en électronique on utilise un *transducteur piézo* pour générer cela. C’est une sorte de petit buzzer capable de vibrer très vite. Très souvent, les télémètres à ultrasons vibrent à une fréquence de 40 kHz.

Une dernière caractéristique des ondes sonores est leur capacité à être réfléchie par les obstacles. En effet, les ondes sonores ont tendance à "rebondir" sur les obstacles. On entend alors l’onde de départ et un peu plus tard la même avec un retard et une plus faible intensité. C’est exactement le même phénomène qu’un écho dans une pièce vide ou en montagne. L’onde sonore se déplace, rebondit sur les murs lisses et revient à votre oreille avec un retard entre le moment où vous avez parlé et celui où vous l’entendez (et une puissance sonore plus faible). En général, dans le domaine de l’acoustique et de la musique, on cherche à supprimer cette caractéristique en recouvrant les murs de matériaux spéciaux. Cependant, dans le cas d’une mesure de distance, on va exploiter cet effet.

### Principe de la mesure

Comme dit précédemment, on va tirer parti du fait que l’onde sonore rebondit sur les obstacles et revient souvent vers l’expéditeur. On va aussi exploiter une autre chose connue, sa vitesse !

En effet, la vitesse de déplacement d’une onde sonore dans l’air est connue depuis longtemps. Elle est d’environ 340 mètres par seconde à 25 degrés Celsius (plutôt lent comparé à la lumière et ses 300 000 km/s ). À partir de là, si on sait quand l’onde est partie et quand on la reçoit de nouveau (après le rebond), on est en mesure de calculer un *temps de vol* de l’onde. On a alors une durée, une vitesse, et on peut en déduire une distance !

Comme l’onde fait un aller-retour (le voyage depuis l’émission de l’onde, le rebond, puis le retour sur le récepteur), il faudra diviser le temps de vol par deux pour ne considérer qu’un trajet (l’aller ou le retour). Le calcul sera alors simple. Une vitesse s’exprime par une distance divisée par un temps *v=d/tv = d/t*v=d/t donc la distance sera la vitesse multipliée par le temps *d=v×td = v \times t*d=v×t.

Passons un peu à la pratique pour mieux comprendre !



Un capteur à ultrasons

# [Mise en œuvre du télémètre](https://zestedesavoir.com/tutoriels/343/realiser-un-telemetre-a-ultrasons/#2-10800_mise-en-uvre-du-telemetre)

Le télémètre que nous allons utiliser est assez simple. Son nom est HC-SR04 et existe en différentes variations. Découvrons-le et voyons comment le faire fonctionner avec notre Arduino.

# Présentation du composant

Le HC-SR04 est ce que l’on appelle communément un "Télémètre à ultrasons". Il est trouvable relativement facilement sur de nombreux sites de fournisseurs de composants électroniques. Les sites livrant depuis la Chine proposent des prix inférieurs à 5 euros en général.

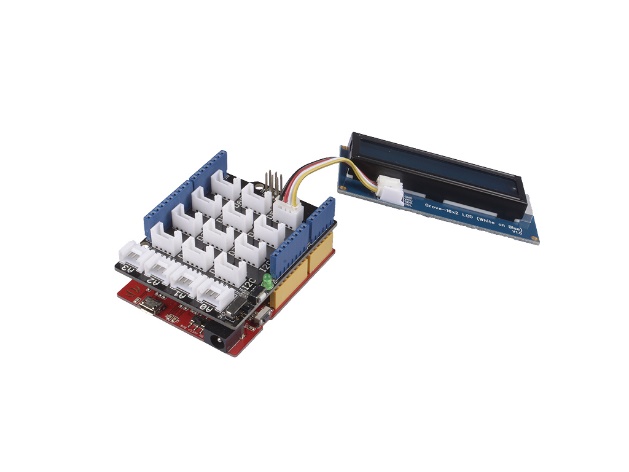
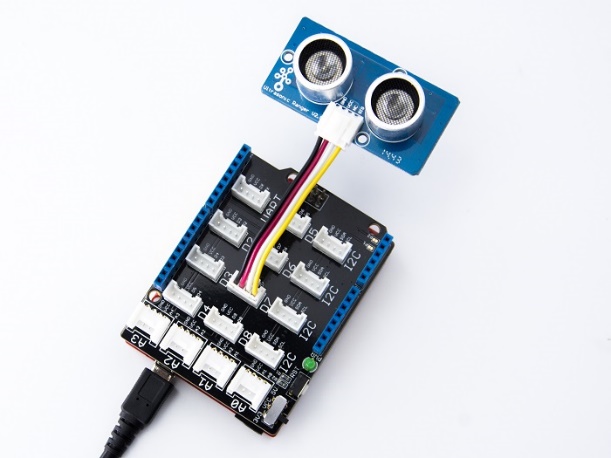
Ce composant possède plusieurs petites choses. Tout d’abord, sur la face avant on peut voir l’émetteur US et son récepteur. Ce sont des petites cellules piézo-électriques qui vont soit vibrer lorsqu’une tension est appliquée (émetteur), soit au contraire produire une tension lorsque une vibration est reçue (récepteur).

Sur la face arrière on trouve plusieurs petits circuits permettant la génération du signal et le traitement de ce dernier. Ainsi, un composant va générer une onde de 40 kHz lors d’un "top départ" et la partie restante s’occupera de la mise en forme de la réception (amplification et filtrage) et de mettre en forme cela proprement sur une broche de sortie. Parlons d’ailleurs des broches. On en trouve 4.

Les premières sont comme toujours VCC et GND qui vont accueillir l’alimentation (respectivement 5V et masse). On trouve ensuite la broche "echo" sur laquelle sera présent le signal de sortie. Enfin, une broche nommée "Trig". Cela signifie "Trigger" soit "déclencheur" ou "gâchette". En mettant cette broche à l’état haut pendant 10µs vous allez déclencher le *ping* pour la mesure. Un "ping" représente le lancement d’une onde ultrason. Pour reprendre l’exemple de l’écho dans la pièce vide, le ping correspondrait au moment où vous émettez un son en parlant.

Le signal de sortie est assez simple à exploiter. Il est initialement à 0, puis passe à 1 lorsque le *ping* est envoyé. Il repasse ensuite à 0 quand l’écho est revenu au récepteur OU s’il n’y a pas de retour durant les 30ms après l’envoi (l’onde est alors considérée perdue).

# Branchement



Les branchements sont eux-mêmes très simples. Il suffira de relier le connecteur par exemple sur le connecteur D7

# Exploitons le capteur avec Arduino

Passons maintenant à la pratique avec Arduino. Le but ici sera d’être capable de faire une mesure de distance puis de l’afficher en millimètres dans la console de la voie série.

# **Déclaration et Setup**

Déclaration de la librairie pour le capteur à ultrason: #include "Ultrasonic.h"

Déclaration des librairies pour l'afficheur lcd : #include <Wire.h>

#include "rgb\_lcd.h"

Activation de l’afficheur : rgb\_lcd lcd;

Déclaration de la broche ou l’on a mis le connecteur du capteur : Ultrasonic ultrasonic(7);

Déclaration du nombre de ligne et de colonne de l’afficheur lcd.begin(16, 2);

# **Loop**

void loop()

{

long RangeInInches; // déclaration des variables

long RangeInCentimeters;

lcd.setCursor(0, 0); // positionnement colonne 0 ligne 0

lcd.print("La mesure est : ");

RangeInInches = ultrasonic.MeasureInInches(); // lecture du résultat en inch

lcd.setCursor(4, 1);

lcd.print(RangeInInches); //0~157 inches

lcd.print(" inch");

delay(1000);

}

# **Travail**

**Pour chaque programme faire l’organigramme et le programme.**

1. Tester le programme et contrôler s’il correspond bien à l’énoncé.

Le cas échéant faites les modifications.

1. Dans ce programme nous voulons afficher le résultat en ‘m’ sur un écran LCD.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 0 | **M** | **E** | **S** | **U** | **R** | **E** | **:** |  |  |  |  |  |  |  |  |  |
| 1 |  |  |  |  | **X** | **X** | **X** | **X** |  | **m** |  |  |  |  |  |  |

1. Dans ce programme nous voulons afficher le résultat en ‘M’ ou ‘mM’ suivant la valeur

de la distance sur un écran LCD. Le passage de mètre à millimètre se fera par automatiquement

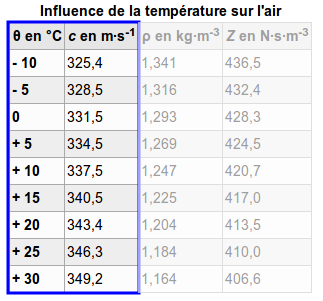
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 0 | **M** | **E** | **S** | **U** | **R** | **E** | **:** |  |  |  |  |  |  |  |  |  |
| 1 |  |  |  |  | **X** | **X** | **X** | **X** |  | **m** |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 0 | **M** | **E** | **S** | **U** | **R** | **E** | **:** |  |  |  |  |  |  |  |  |  |
| 1 |  |  |  |  | **X** | **X** | **X** | **X** |  | **m** | **m** |  |  |  |  |  |

# [**Aller plus loin**](https://zestedesavoir.com/tutoriels/343/realiser-un-telemetre-a-ultrasons/#4-10828_aller-plus-loin)

La température a un impact sur la vitesse des ondes (ainsi que la pression et d’autres choses). Une amélioration de notre télémètre serait donc d’avoir une correction de la vitesse via une mesure de la température !

Pour débuter, prenons connaissance des données. Le cours nous dit que la vitesse de propagation du son dans l’air suit à peu près le tableau suivant :



Vitesse du son en fonction de la température

Si l’on fait un graphique avec ce tableau, on peut déduire les valeurs de la relation *V=f(t)V = f(t)*V=f(t) (vitesse en fonction de la température) qui est presque linéaire et donc peut être modélisée par une fonction affine. En faisant le calcul de la variation via les deux points les plus extrêmes du bout de droite, on obtient un coefficient de 0.6 m/s par degrés Celsius (*325.4−349.2−10−30\frac{325.4-349.2}{-10-30}*−10−30325.4−349.2 ). On observe aussi une ordonnée à l’origine (soit à 0 degré) de 331.5 mètres par seconde.

Tout cela nous donne la relation : *V=at+bV = at+b*V=at+b (avec *VV*V la vitesse en mètres par seconde et *tt*t la température en degrés Celsius). *aa*a vaudra ainsi 0.6 et *bb*b vaudra 331.5.

Je ne donnerai pas le détail du montage ni le code ici cependant. Je vous le laisse en tant qu’évolution/exercice si vous le souhaitez, en rajoutant un capteur de température au montage (un LM35 par exemple). Voici cependant la logique de code qu’il faudrait suivre pour implémenter cela :

void loop() { // On fait la mesure de température

// lecture de la température en degrés Celsius .

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}