

TELEMETRIE A ULTRASONS AVEC ARDUINO

PREAMBULE

Pour mesurer des distances, on utilise en général une règle ou un ruban mètre. Mais comment faire pour avoir une distance numérique en temps réel ? Parmi les solutions possibles: utiliser la technologie des ultrasons pour réaliser un **télémètre**.

En effet, 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, dans ce TP, réaliser un petit télémètre à ultrasons via une Arduino et avec un affichage intégré sur un écran LCD.

Il nous faut pour cette manipulation:

- Un arduino Uno
- Un écran LCD alphanumérique.
- Un capteur HC-SR04 ;
- Quelques fils de liaison
- Un bon savoir!

TELEMETRIE A ULTRASONNS AVEC ARDUINO

Rappel sur les ultrasons

Un ultrason est une onde sonore à haute fréquence. Par haute fréquence on entend 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.

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.

TELEMETRIE A ULTRASONS AVEC ARDUINO

Une onde sonore c'est quoi ? (suite)

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 réfléchiée 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.

TELEMETRIE A ULTRASONS AVEC ARDUINO

Principe de la mesure

Comme indiqué 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 $v=d/t$

donc la distance sera la vitesse multipliée par le temps $d=v \times t$.

TELEMETRIE A ULTRASONS AVEC ARDUINO

Présentation du HC-SR04

Le HC-SR04 est ce que l'on appelle communément un "Télémètre à ultrasons".

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.

TELEMETRIE A ULTRASONS AVEC ARDUINO

Présentation du HC-SR04

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 $10\mu\text{s}$ 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).

TELEMETRIE A ULTRASONS AVEC ARDUINO

Utilisation du module Ultrason HC-SR04 avec l'Arduino

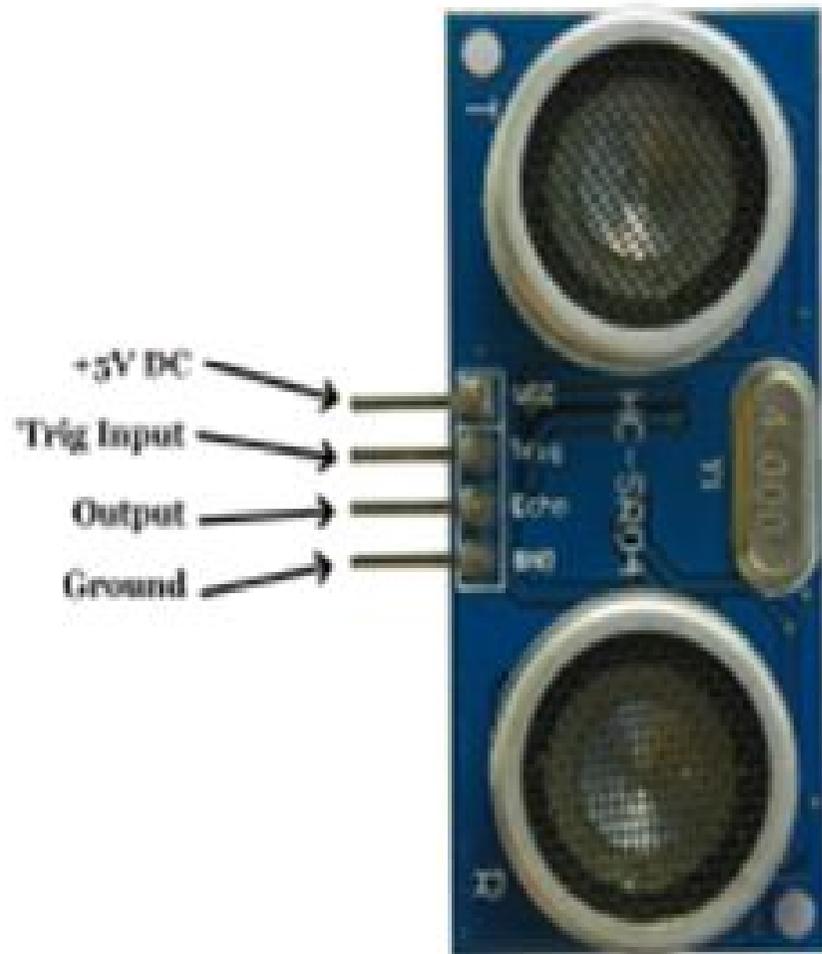
Nous allons tester l'utilisation du module Ultrason HC-SR04.

L'intérêt de ce module est son prix, relativement faible.

Les caractéristiques techniques du module sont les suivantes :

- Alimentation : 5v.
- Consommation en utilisation : 15 mA.
- Gamme de distance : 2 cm à 5 m.
- Résolution : 3 mm.
- Angle de mesure : $< 15^\circ$.

Le brochage du module est le suivant :



TELEMETRIE A ULTRASONS AVEC ARDUINO

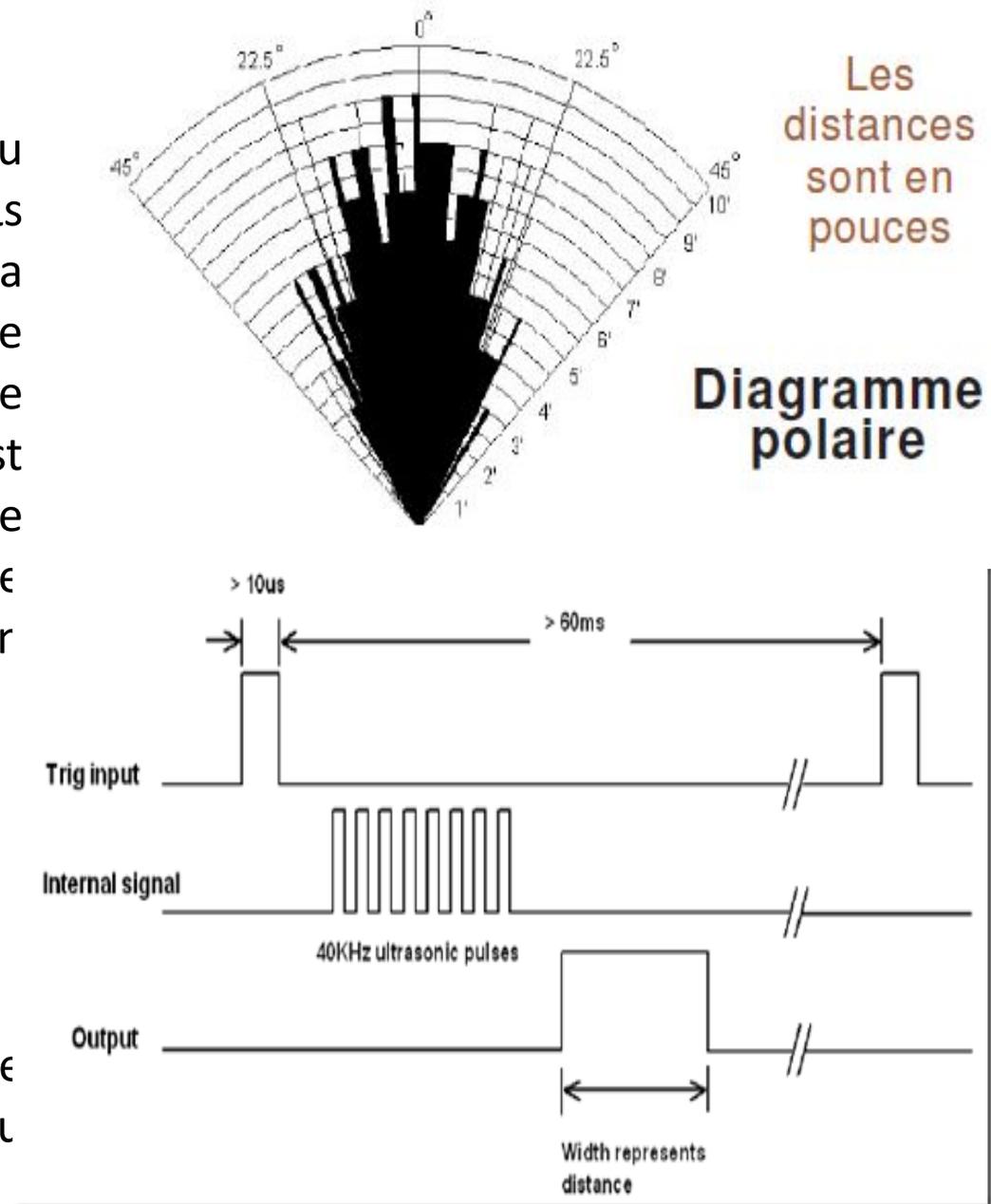
Le fonctionnement du module:

Il faut envoyer une impulsion niveau haut (à + 5v) pendant au moins 10 μ s sur la broche 'Trig Input'; cela déclenche la mesure. En retour la sortie 'Output' ou 'Echo', va fournir une impulsion + 5v dont la durée est proportionnelle à la distance si le module détecte un objet. Afin de pouvoir calculer la distance en cm, on utilisera la formule suivante :

$$\text{distance} = \text{durée de l'impulsion} / 58$$

La durée de l'impulsion est en μ s

Voici une représentation graphique de la séquence de fonctionnement du module :

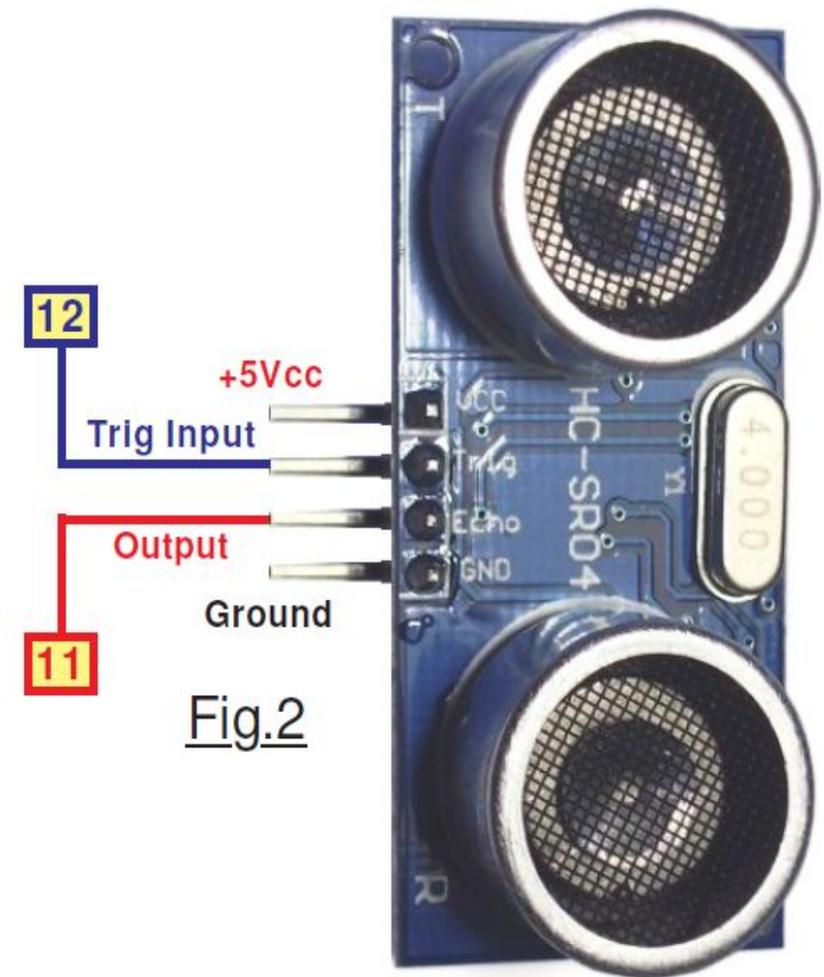
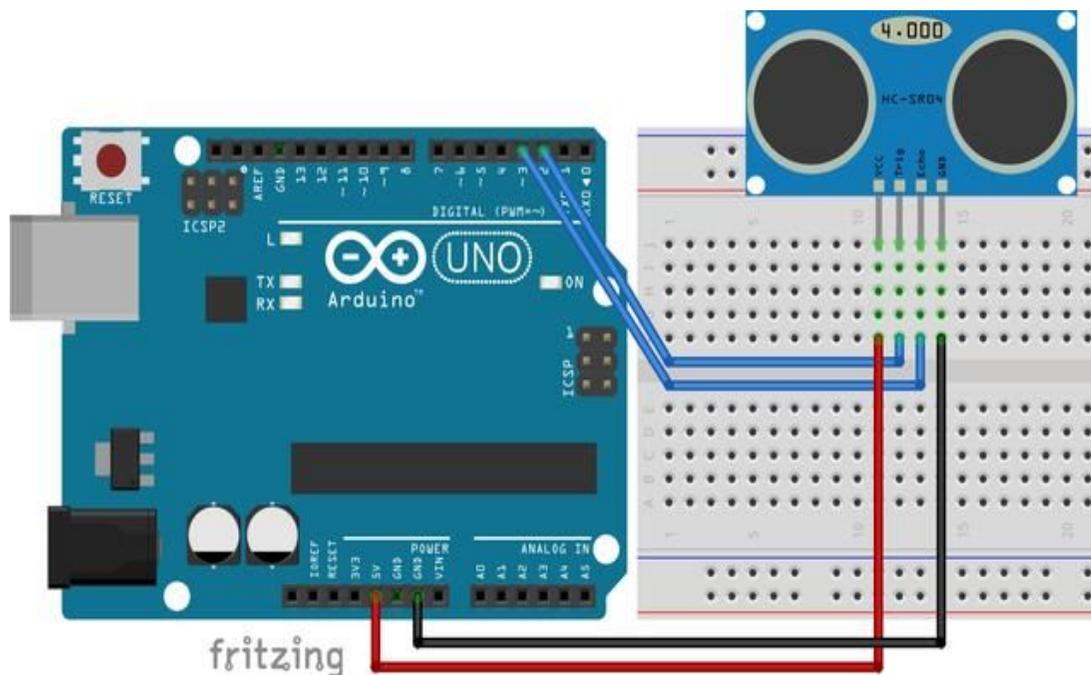


TELEMETRIE A ULTRASONS AVEC ARDUINO

Câblage

Le câblage du module à l'Arduino sera le suivant :

- broche 12 de l'Arduino vers Trig.
- broche 11 de l'Arduino vers Echo.



TELEMETRIE A ULTRASONNS AVEC ARDUINO

Le programme sera le suivant :

```
/* Utilisation du capteur Ultrason HC-SR04 */  
// définition des broches utilisées  
int trig = 12;  
int echo = 11;  
long lecture_echo;  
long cm;  
void setup()  
{  
  pinMode(trig, OUTPUT);  
  digitalWrite(trig, LOW);  
  pinMode(echo, INPUT);  
  Serial.begin(9600);  
}
```

TELEMETRIE A ULTRASONNS AVEC ARDUINO

Le programme sera le suivant :

```
void loop()
{
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  lecture_echo = pulseIn(echo, HIGH);
  cm = lecture_echo / 58;
  Serial.print("Distancem : ");
  Serial.println(cm);
  delay(1000);
}
```

TELEMETRIE A ULTRASONS AVEC ARDUINO

L'écran avec le télémètre

Maintenant que le télémètre fonctionne bien en émettant la mesure vers le port série, il est plus intéressant que l'affichage soit directement obtenu sur un afficheur LCD. Ainsi, nous pouvons prétendre à une autonomie vis-à-vis du PC.

```
#include "LiquidCrystal.h"  
#define VITESSE 340
```

```
// VÉRIFIEZ AVEC VOS PROPRES BROCHES  
LiquidCrystal lcd(6,7,2,3,4,5); //liaison 4 bits de données  
const int USTrig = 8; // Déclencheur sur la broche 8  
const int USEcho = 9; // Réception sur la broche 9
```

```
void setup() {  
  lcd.begin(2, 16); // Initialisation de l'écran (2 lignes 16 caractères pour ma part)  
  pinMode(USTrig, OUTPUT);  
  pinMode(USEcho, INPUT);  
  lcd.print("-+ Eskimetre +-"); }  
}
```

TELEMETRIE A ULTRASONS AVEC ARDUINO

```
void loop() { // 1. Un état haut de 10 microsecondes est mis sur la broche "Trig"  
digitalWrite(USTrig, HIGH);  
delayMicroseconds(10); //on attend 10 µs  
// 2. On remet à l'état bas la broche Trig  
digitalWrite(USTrig, LOW);  
  
// 3. On lit la durée d'état haut sur la broche "Echo"  
unsigned long duree = pulseIn(USEcho, HIGH);  
if(duree > 30000)  
{  
//si la durée est supérieure à 30ms, l'onde est perdue  
lcd.clear();  
lcd.home(); // Replace le curseur en haut à gauche  
lcd.print("Onde perdue :(");  
lcd.setCursor(0,1);  
lcd.print("Mesure echouee !");  
}  
Else  
{
```

TELEMETRIE A ULTRASONS AVEC ARDUINO

```
// 4. On divise cette durée par deux pour n'avoir qu'un trajet  
duree = duree/2;  
// 5. On calcul la distance avec la formule  $d=v*t$   
unsigned int distance = duree*(VITESSE/1000.0);  
//on multiplie par la vitesse,  $d=t*v$   
// 6. On affiche !  
lcd.clear();  
lcd.home(); // Replace le curseur en haut à gauche  
char message[16] = "";  
sprintf(message, " Dist : %4d mm", distance);  
lcd.print(message);  
sprintf(message, "Temps : %4d us", duree);  
lcd.setCursor(0,1);  
lcd.print(message);  
}  
delay(250); // Petite pause  
}
```