



FICHE-TP

Mesure de la célérité des US dans l'air

NIVEAU	Spécialité 1ère
Partie du programme officiel	Ondes et signaux
Capacité expérimentale ou numérique	Déterminer, par exemple à l'aide d'un microcontrôleur ou d'un smartphone, une distance ou la célérité d'une onde

Présentation du TP :

Objectif : Utiliser un microcontrôleur pour déterminer la célérité des ultrasons dans l'air en mesurant la durée de propagation d'une salve US sur une distance connue.

Démarche de la mesure et outils :

Il s'agit de programmer la carte UNO pour qu'elle mesure la durée d'aller-retour d'une salve US sur un obstacle grâce à un module US, puis calcule la célérité du son dans l'air connaissant la distance module-obstacle. L'interface de programmation utilisée ici est l'EDI Arduino.

Instructions et explications détaillées sur l'EDI Arduino : [Document-3-Programmation-avec-Arduino-IDE.pdf](#).

Matériel : carte type UNO + module US + mini-platine + 4 fils Dupont M-M (ou câble Grove), obstacle (boîtier de CD...), feuille A4 ou règle...



← Module US HC-SR04

(envoient et reçoivent des salves d'ondes ultrasonores de fréquence 40 kHz)

Module Grove Ultrasonic Ranger →

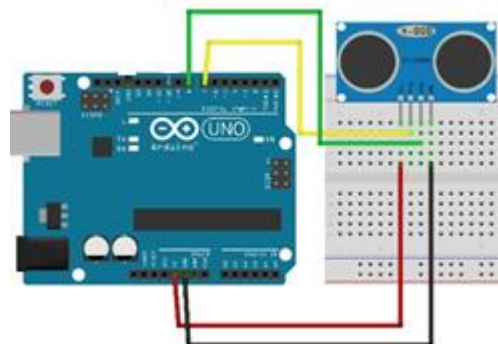


1) Schéma du montage sur carte UNO

HC-SR04 :

Carte microcontrôleur	HC-SR04
5 V	Vcc
GND	Gnd
7	Echo
8	Trig

La mini-platine sert aussi de support pour le module US.



Grove Ultrasonic Ranger : brancher le câble Grove sur le connecteur Grove D7 de l'interface Grove.

2) Travail demandé aux élèves : *exemple de déroulé possible*

1) **Introduction** (accroche et objectif ou problème posé).

2) **Prise en main du dispositif de mesure :**

Réaliser le montage à l'aide du schéma fourni, placer le module US et l'obstacle de part et d'autre d'une feuille A4. Avec l'IDE Arduino, ouvrir le programme donné [mesure t US.ino](#) (ou [mesure t US grove.ino](#) - cf. ci-après) qui permet mesurer puis d'afficher la durée de propagation entre l'émission et la réception d'une salve US par le module US. Le téléverser puis ouvrir le moniteur série pour visualiser son exécution.

3) **Adaptation du code pour répondre au problème posé :**

Modifier le code pour que la carte UNO calcule et affiche la valeur de la vitesse de propagation de la salve US dans l'air (pour cela il est nécessaire de créer les variables d et v. Sauvegarder sous un autre nom.

Aide : fichier donné [mesure_v_US_aide.ino](#) (cf. ci-après).

4) **Validation :**

Comparer à la valeur théorique fournie dans les conditions de l'expérience et proposer une amélioration de la méthode de mesure (augmenter d, faire une moyenne sur plusieurs mesures...).

3) Fichiers ino

Fichier de départ proposé aux élèves :

Nom :

mesure_t_US.ino

Ce script est adapté au module HC-SR04

Capture
d'écran :

```
3 // Déclaration des constantes
4 const byte Broche_echo = 7; // Broche Echo du HC-SR04 sur broche D7 de la carte UNO
5 const byte Broche_trigger = 8; // Broche Trigger du HC-SR04 sur broche D8 de la carte UNO
6
7 // Déclaration des variables
8 unsigned int t; // durée A/R de la salve US en µs : entier positif limité à 2*32768 (codage
9
10 void setup()
11 {
12 // Initialisation des ports
13 Serial.begin(9600); // établissement d'une liaison de communication série av
14 pinMode(Broche_trigger,OUTPUT); // broche Trigger de la carte UNO en sortie (émettra l'i
15 pinMode(Broche_echo,INPUT); // broche Echo de la carte UNO en entrée (recevra le sig
16 // La sortie Echo du HC-SR04 est par défaut à 0V (LOW)
17 // Impulsion électrique de 10 µs sur la broche trigger (émetteur US) qui produit un train
18 digitalWrite(Broche_trigger,LOW); // (re)met la sortie Trigger à 0V
19 delay(10); // il est important d'attendre au moins 10ms (sinon mauvais fonctionnement)
20 digitalWrite(Broche_trigger,HIGH); // met la sortie Trigger à 5V
21 delayMicroseconds(10); // pendant 10 µs
22 digitalWrite(Broche_trigger,LOW); //
23
24 // Mesure de la durée de propagation A/R du train d'impulsions US : la sortie Echo du modu
25 // une impulsion haute dont la durée est égale à la durée de propagation A/R du train US (
26 t = pulseIn(Broche_echo,HIGH); // mesure une durée (unsigned long en µs) d'état haut
27 Serial.print("t = "); // affiche via le port série
28 Serial.print(t); // affiche la valeur de t
29 Serial.println(" µs"); // affiche l'unité et passe à la ligne
30 }
```

Nom :

mesure_t_US_grove.ino

Ce script est adapté au module Grove Ultrasonic Ranger

Lignes
changées

```
3 // Déclaration des constantes
4 const byte Broche_US = 7; // Broche unique Echo/Trigger du module US sur broche D7 de la c
5
6
7 // Déclaration des variables
8 unsigned int t; // durée A/R de la salve US en µs : entier positif limité à 2*32768 (codage
9
10 void setup()
11 {
12 // Initialisation des ports
13 Serial.begin(9600); // établissement d'une liaison de communication série av
14 pinMode(Broche_US,OUTPUT); // broche_US de la carte UNO en sortie (trigger : émettra l'
15
16 // Impulsion électrique de 10 µs qui produit un train de 8 impulsions US de 40kHz
17 digitalWrite(Broche_US,LOW); // (re)met la sortie à 0V
18 delay(10); // il est important d'attendre au moins 10ms (sinon mauvais fonctionnement)
19 digitalWrite(Broche_US,HIGH); // met la sortie à 5V
20 delayMicroseconds(10); // pendant 10 µs
21 digitalWrite(Broche_US,LOW); //
22
23 // Mesure de la durée de propagation A/R du train d'impulsions US : le module US émet vers
24 // une impulsion haute dont la durée est égale à la durée de propagation A/R du train US (
25 pinMode(Broche_US,INPUT); // broche_US de la carte UNO en entrée (recevra le signal US
26 t = pulseIn(Broche_echo,HIGH); // mesure une durée (unsigned long en µs) d'état haut
```

Fichier d'aide (solution partielle) :

Nom :

mesure_v_US_aide.ino

Lignes
ajoutées
ou
changées

```
7 // Definition des variables
8 unsigned int t; // durée A/R de la salve US en µs : entier positif limité à 2*32768 (codage
9 float d = 0.297; // distance module US - obstacle (feuille A4 29.7cm) : valeur décimale en m
10 unsigned int v; // vitesse de propagation du son : valeur entière positive en m/s
11
12 -----
27 v = ; // LIGNE A COMPLETER : écrire la formule qui calcule la vitesse v à partir de t et d
28
29 Serial.print("v = "); // affiche via le port série
30 Serial.print(v); // // affiche la valeur de v
31 Serial.println(" m/s"); // affiche l'unité et passe à la ligne
```