



Capteurs - Distance - Ultrasons

[Mise à jour le 7/7/2022]

- **Lectures connexes**

- [Bien choisir un capteur de proximité](#)
- [Comment choisir le meilleur capteur ultrason](#)
- Vidéo - [Exemples d'applications](#)

1. Généralités

L'ultrason est une onde mécanique et élastique, qui se propage au travers de supports fluides, solides, gazeux ou liquides. La gamme de fréquences des ultrasons se situe entre **16 000 et 10 000 000 Hertz**.



Le nom vient du fait que leur fréquence est trop élevée pour être audible pour l'oreille humaine (le son est trop aigu : la gamme de fréquences audibles par l'homme se situe entre **20 et 20 000 Hertz**. Ces seuils sont cependant variables avec l'âge), de la même façon que les infrasons désignent les sons dont la fréquence est trop faible pour être perceptible par l'oreille humaine. Lorsque la fréquence est audible pour l'oreille humaine, on parle tout simplement de son.

Les ultrasons sont utilisés dans l'industrie ainsi que dans le domaine médical. [Wikipédia](#)

2. Capteurs numériques

2.1 HC-SR04



- **Source** : [wiki](#)

Ce module permet d'évaluer les distances entre un objet mobile et les obstacles rencontrés. Il suffit d'envoyer une impulsion de 10 μ s en entrée et le capteur renvoie une largeur d'impulsion proportionnelle à la distance.

- **Distributeur** : [Gotronic](#)

- **Caractéristiques**

- Alimentation: 5 Vcc
- Consommation: 15 mA
- Fréquence: 40 kHz
- Portée: de 6...10 cm à 4 m
- Déclenchement: impulsion TTL positive de 10µs
- Signal écho: impulsion positive TTL proportionnelle à la distance.
- Calcul: distance (cm) = impulsion (µs) / 58
- Dimensions: 45 x 21 x 18 mm

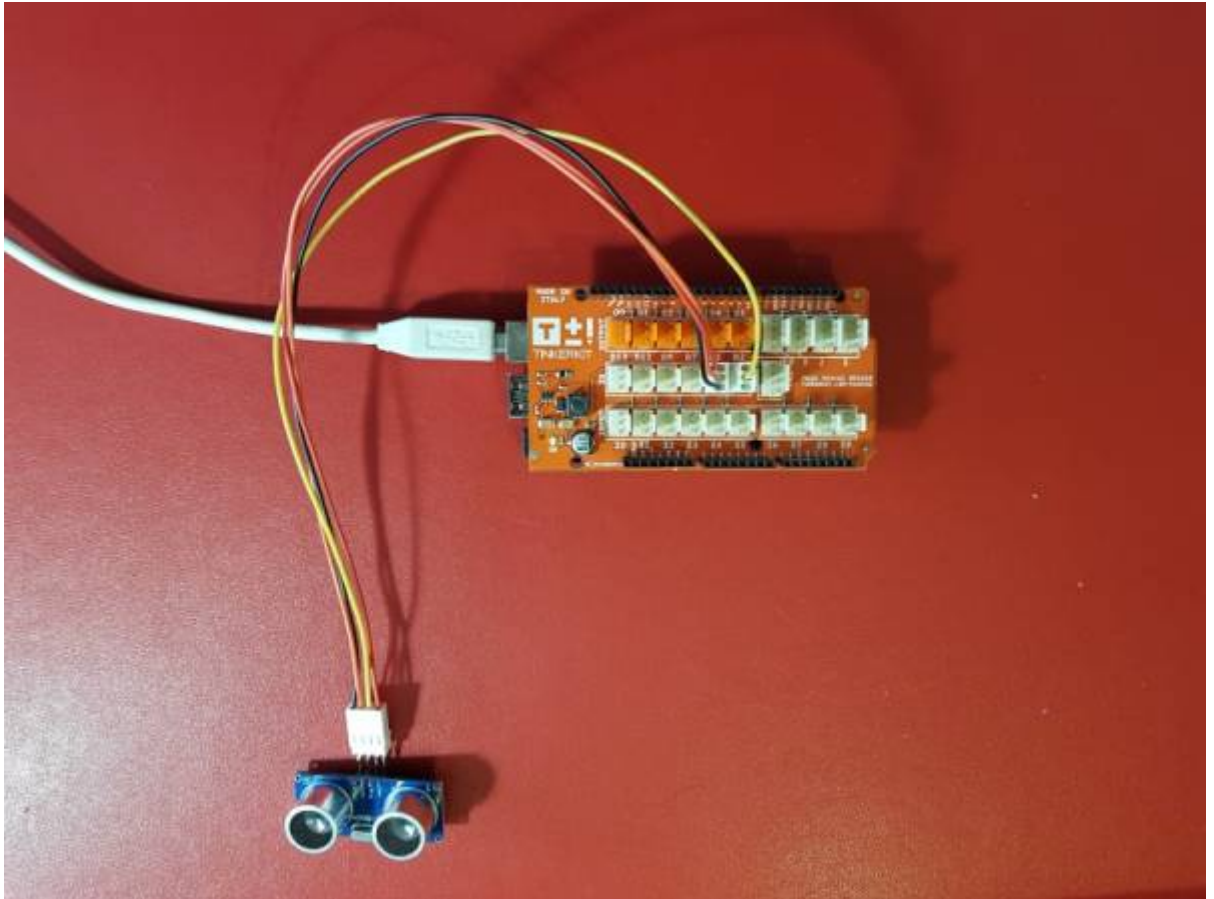


- **Documentation**

- Fichier Acrobat Reader à télécharger [ici](#)

- **Programmation d'une carte Arduino Mega 2560**

- Bibliothèques à installer dans l'IDE : aucune
- Connexion à un shield [Tinkerkit Mega v2](#).



- Un premier exemple



[HCSR04.cpp](#)

```
/*  
 * Code d'exemple pour un capteur à ultrasons HC-SR04.  
 * Carte Arduino Mega 2560  
 */
```

```
/* Constantes pour les broches */
const byte TRIGGER_PIN = 2; // Broche TRIGGER
const byte ECHO_PIN = 4;    // Broche ECHO

/* Constantes pour le timeout */
const unsigned long MEASURE_TIMEOUT = 25000UL; // 25ms = ~8m à 340m/s

/* Vitesse du son dans l'air en mm/us */
const float SOUND_SPEED = 340.0 / 1000;

/** Fonction setup() */
void setup() {

    /* Initialise le port série */
    Serial.begin(115200);

    /* Initialise les broches */
    pinMode(TRIGGER_PIN, OUTPUT);
    digitalWrite(TRIGGER_PIN, LOW); // La broche TRIGGER doit être à LOW
    // au repos
    pinMode(ECHO_PIN, INPUT);
}

/** Fonction loop() */
void loop() {

    /* 1. Lance une mesure de distance en envoyant une impulsion HIGH de
    10µs sur la broche TRIGGER */
    digitalWrite(TRIGGER_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIGGER_PIN, LOW);

    /* 2. Mesure le temps entre l'envoi de l'impulsion ultrasonique et
    son écho (si il existe) */
    long measure = pulseIn(ECHO_PIN, HIGH, MEASURE_TIMEOUT);

    /* 3. Calcul la distance à partir du temps mesuré */
    float distance_mm = measure / 2.0 * SOUND_SPEED;

    /* Affiche les résultats en mm, cm et m */
    Serial.print(F("Distance: "));
    Serial.print(distance_mm);
    Serial.print(F("mm ("));
    Serial.print(distance_mm / 10.0, 2);
    Serial.print(F("cm, "));
    Serial.print(distance_mm / 1000.0, 2);
    Serial.println(F("m"));

    /* Délai d'attente pour éviter d'afficher trop de résultats à la
    seconde */
}
```

```
delay(500);  
}
```



[Télécharger](#) le projet PlatformIO pour VSCode.

2.2 Module VMA306 à capteur HC-SR05



- **Source** : [pdf](#)

Ce module est basé sur un capteur à ultrasons HC-SR05 et permet d'évaluer les distances entre un objet mobile et les obstacles rencontrés.

- **Distributeur** : [Gotronic](#)

- **Caractéristiques**

- Alimentation: 4,5 à 5,5 Vcc
- Consommation:
 - mini: 10 mA
 - maxi: 40 mA
- Fréquence: 40 kHz
- Portée: de 2 cm à 4,5 m
- Déclenchement: impulsion TTL positive de 10µs
- Signal écho: impulsion positive TTL proportionnelle à la distance.

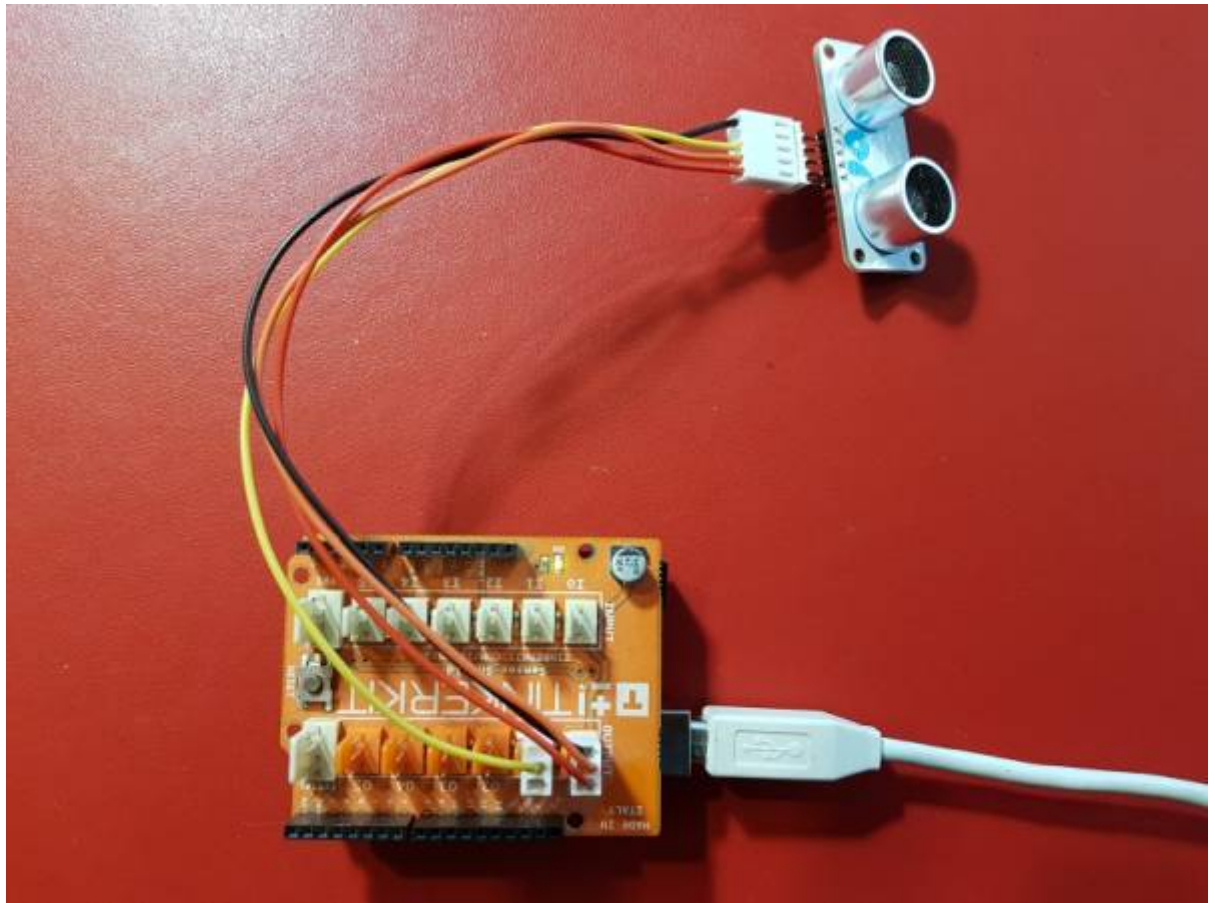


- **Documentation**

- Manuel d'utilisation du capteur à ultrasons VMA306 à télécharger [ici](#)

- **Programmation d'une carte Arduino Uno R3**

- Bibliothèques à installer dans l'IDE : aucune
- Connexion à un shield [Tinkerkit v2](#).



◦ Un premier exemple



HCSR05.cpp

```
/////////////////////////////////////////////////////////////////
// Programme test pour capteur HC-SR05 //
// Go Tronic 2017 //
/////////////////////////////////////////////////////////////////
#define trigPin 10 // Tinkerkit 01
#define echoPin 11 // Tinkerkit 00

long duration, distance;

void setup()
{
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.println("== Debut du programme ==");
}

void loop()
{
  // Envoie de l'onde
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
```

```
digitalWrite(trigPin, LOW);  
// Réception de l'écho  
duration = pulseIn(echoPin, HIGH);  
  
// Calcul de la distance  
distance = (duration / 2) / 29.1;  
if (distance >= 400 || distance <= 0)  
{  
    Serial.println("Hors plage");  
}  
else  
{  
    Serial.print("distance = ");  
    Serial.print(distance);  
    Serial.println(" cm");  
}  
delay(500); // délai entre deux mesures  
}
```



Télécharger le projet PlatformIO pour VSCode.

C#

- Programmation d'une carte **FEZ Duino** sous **TinyCLR OS** avec l'IDE Visual Studio Community

A venir

From:

<http://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:

<http://webge.fr/dokuwiki/doku.php?id=materiels:capteurs:distance:distus&rev=1657179601>

Last update: **2022/07/07 09:40**

