

Généralités

Le capteur HC-SR04 utilise les ultrasons pour déterminer la distance d'un objet.

L'écart est d'environ 3 cm avec un objet placé à 2 m, ce qui représente une erreur inférieure à 2 %.

Caractéristiques

- Dimensions : 45 mm x 20 mm x 15 mm
- Plage de mesure : 2 cm à 400 cm
- Résolution de la mesure : 0.3 cm
- Angle de mesure efficace : 15 °
- Largeur d'impulsion sur l'entrée de déclenchement : 10 µs



Broches de connexion

- Vcc = Alimentation +5 V DC
- Trig = Entrée de déclenchement de la mesure (Trigger input)
- Echo = Sortie de mesure donnée en écho (Echo output)
- GND = Masse de l'alimentation

Spécifications et limites

- Tension d'alimentation : 5.0 V à ± 0.5 V
- Courant de repos : 2.0 mA à ± 0.5 mA
- Courant de fonctionnement : 15 ± 5 mA
- Fréquence des ultrasons : 40 kHz

Fonctionnement

Déclenchement d'une mesure par envoi sur l'entrée Trigger d'une impulsion de 10 µs 5V

Le capteur émet alors une série de 8 impulsions ultrasoniques à 40 kHz

Puis il attend le signal réfléchi, lorsque celui-ci est détecté, une impulsion de **durée proportionnelle à la distance mesurée** est envoyée sur la sortie "Echo".

La distance de l'objet détecté est égale au temps de l'impulsion de sortie divisée par deux (le signal fait un aller-retour) et multipliée par la vitesse du son dans l'air (340m/s).

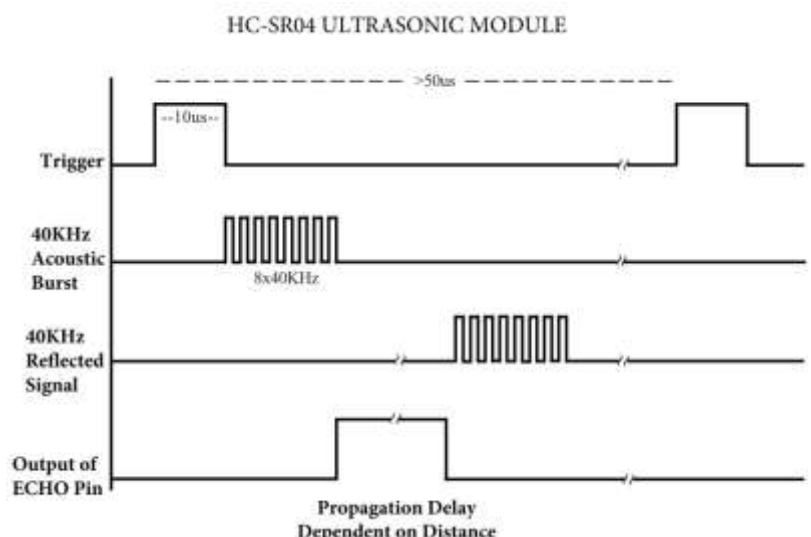
Calcul

Distance parcourue par le son : vitesse du son (340 m/s) * temps aller retour du son / 2

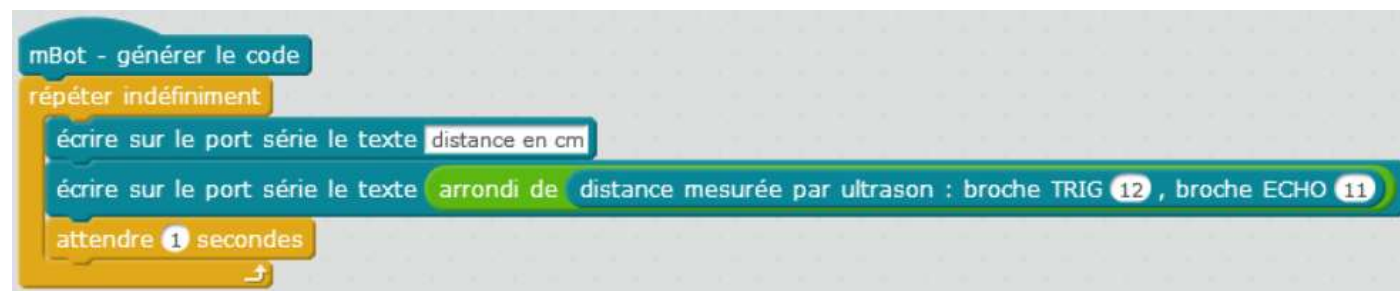
Distance = 340m/s * t en sec / 2 = 34000 cm / 1000000µs * t en µs / 2 = 17 / 1000 * t

T = Distance * 1000 / 17 = D * 58,82 en µs

Distance en cm = temps propagation en µs / 58



Utilisation de mBlock



```
float getDistance(int trig,int echo){
  pinMode(trig,OUTPUT);
  digitalWrite(trig,LOW);
  delayMicroseconds(2);
  digitalWrite(trig,HIGH);
  delayMicroseconds(10);
  digitalWrite(trig,LOW);
  pinMode(echo, INPUT);
  return pulseIn(echo,HIGH,30000)/58.0;
}

void setup(){
  Serial.begin(115200);
}

void loop(){
  Serial.println("distance en cm");
  Serial.println(round(getDistance(12,11)));
  delay(1);
  _loop();
}
```

Et si « connecter/ com3 » établie on a :

```
16:16:30.046 < distance en cm
32.00
16:16:31.046 < distance en cm
32.00
16:16:31.062 < 32.00
16:16:32.062 < distance en cm
32.00
```

send encode mode rcv encode mode
 mode binaire mode texte mode binaire mode texte

Code Arduino	Observations
<pre>float getDistance(int trig,int echo){ pinMode(trig,OUTPUT); digitalWrite(trig,LOW); delayMicroseconds(2); digitalWrite(trig,HIGH); delayMicroseconds(10); digitalWrite(trig,LOW); pinMode(echo, INPUT); return pulseIn(echo,HIGH,30000)/58.0; } }</pre>	<p>Fonction getDistance(int trig,int echo) Fonction avec un return de format « float », ce return est le résultat de $\text{pulseIn}(\text{echo}, \text{HIGH}, 30000) / 58.0$ $\text{pulseIn}(\text{pin}, \text{value}, \text{timeout})$ retourne la longueur, en μs, de l'impulsion présente sur la pin echo si celle-ci dure moins que le timeout (ici $30000\mu\text{s}$ ou $30000/48 = 625$ cm) sinon 0 si l'impulsion n'est pas terminée.</p> <p>Création d'une impulsion de $10\mu\text{s}$ sur la sortie pin12 de la carte Arduino, à envoyer sur l'entrée pin Trig du capteur ultrason :</p> <ol style="list-style-type: none"> 1. Configuration de la pin 12 (nommée trig) en OUTPUT 2. Ecriture d'un 0 (LOW) pendant $2\mu\text{s}$ 3. Ecriture d'un 1 (HIGH) pendant $10\mu\text{s}$ 4. Ecriture d'un 0 (LOW) <p>Configuration de la pin 11 (nommée echo) en INPUT, pour recevoir l'impulsion echo en provenance du capteur Lecture de l'impulsion echo à l'entrée pin 11 et transformation du temps aller et retour de l'echo en distance en cm (entre capteur et cible) $D \text{ cm} = \text{largeur écho} / 58$</p>

Utilisation de l'IDE ARDUINO (sans la bibliothèque NewPing)

Code Arduino croquis « sketch_Capteur_ultrason »

```
//sketch créé par Jacques le 4/02/2018

#define TRIG 12 //pin 12 carte UNO pour envoyer pulse de 10µs
#define ECHO 11 // pin 11 carte UNO pour recevoir pulse Echo

const long TIMEOUT = 30000UL; // 30ms soit mesure à moins de 6m25
unsigned long distance; //

void setup() {
  pinMode(TRIG,OUTPUT); // configuration pin TRIG en sortie, elle sera raccordée à l'entrée Trig du capteur HC-SR04
  pinMode(ECHO,INPUT); // configuration pin ECHO en entrée, elle sera raccordée à la sortie Echo du capteur HC-SR04
  digitalWrite(TRIG, LOW); //Elle doit être à 0 au démarrage

  Serial.begin(115200);
}

void loop() {
  //création impulsion trigger de 10µs
  digitalWrite(TRIG,HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);
  //mesure de la distance en cm capteur / cible
  distance = pulseIn(ECHO, HIGH, TIMEOUT)/48;
  //Affichage mesure toutes les 0.5 seconde
  Serial.print("Distance = ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(500);
}
```

En cliquant sur le moniteur série

Nota si la liaison USB n'est pas enlevée la liaison série est connectée



Utilisation de l'IDE ARDUINO (avec la bibliothèque NewPing)

Code Arduino croquis « sketch_Capteur_Ultrason_Lib_NewPing »

Cela suppose que la librairie NewPing est installée dans l'IDE ARDUINO
Vérifier l'existence dans Croquis/Inclure une bibliothèque//NewPing

```
//sketch créé par Jacques le 4/02/2018

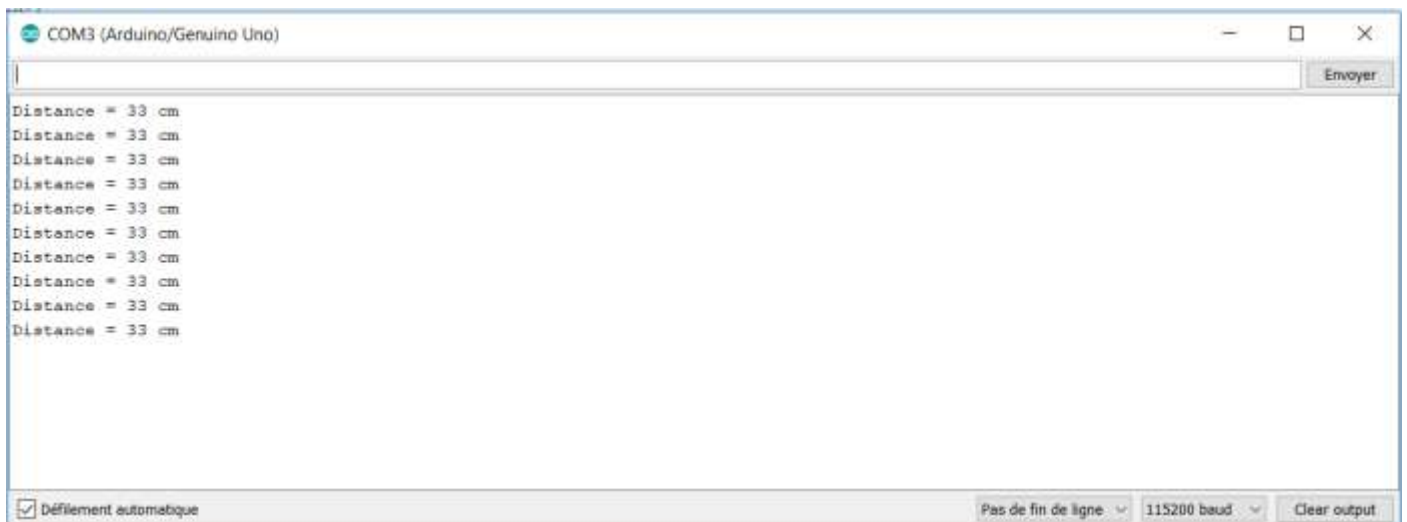
#include <NewPing.h>
NewPing sonar(12,11); //NewPing sonar(trigger pin,echo pin);
long distance;

void setup(){
  Serial.begin(115200);
}

void loop(){

distance = sonar.ping_cm();

//Affichage mesure toutes les 0.5 seconde
  Serial.print("Distance = ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(500);
  delay(500);
}
```



The screenshot shows the serial monitor window titled "COM3 (Arduino/Genuino Uno)". The output displays a series of ten lines, each containing the text "Distance = 33 cm". The window includes a text input field at the top right with an "Envoyer" button. At the bottom, there are checkboxes for "Défilement automatique" (checked) and "Pas de fin de ligne", along with a baud rate dropdown set to "115200 baud" and a "Clear output" button.