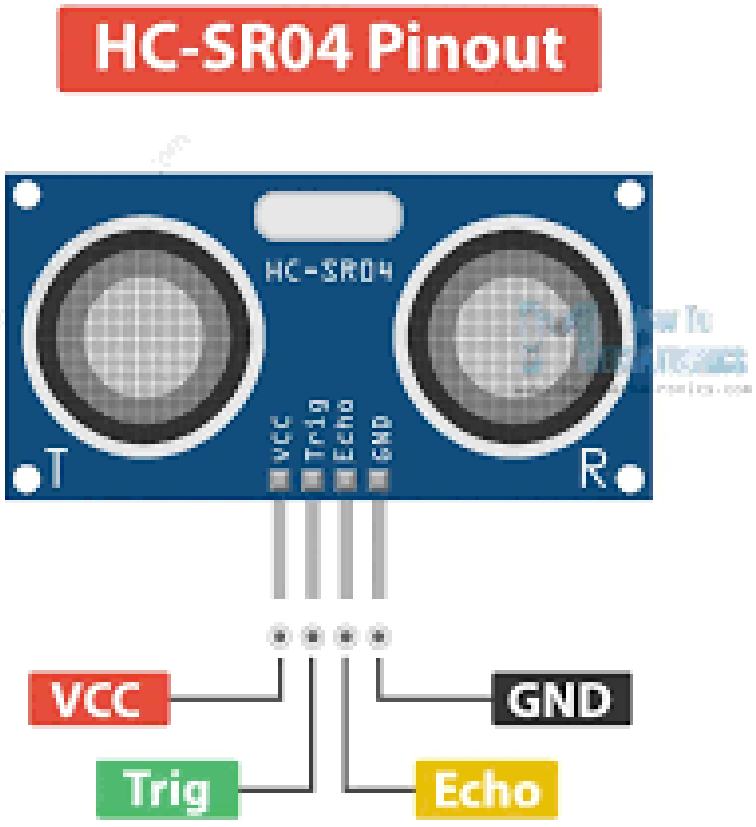
ARDUINO CAPTEUR

HC-SR04

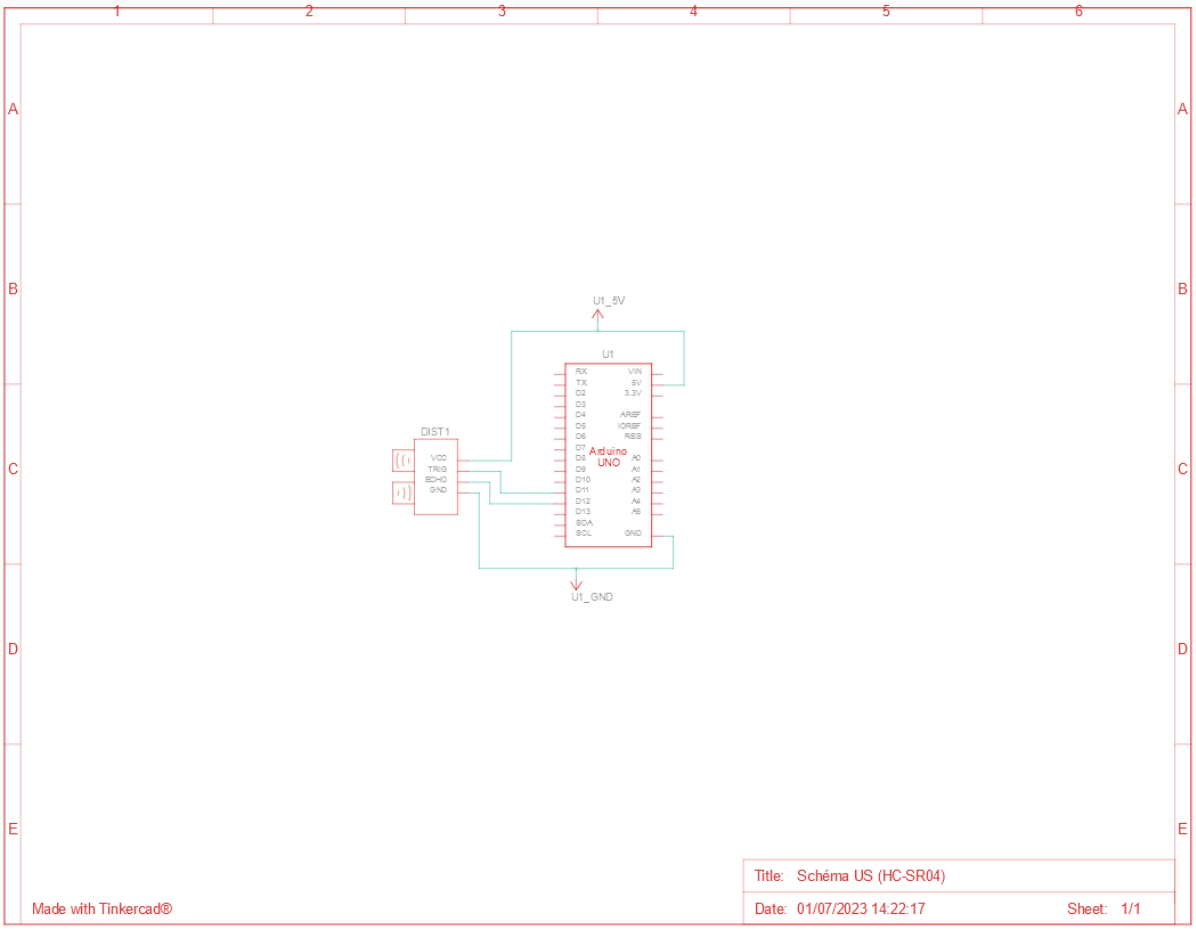


SOMMAIRE

INTRODUCTION

Ce document va traiter de l'utilisation du capteur ultrason (HC-SR04) via une carte arduino uno. Celui-ci va expliquer le schéma de connexion entre la carte et le capteur ainsi que deux codes de mesure. Le premier permet d'afficher la mesure dans le moniteur série et le second est une variante du premier mais en mettant le code dans une fonction.

SCHÉMA ELECTRONIQUE



Le schéma entre la carte arduino et le capteur ultrason est simple, il ne nécessite que quatre fils :

- Le +5V de la carte au Vcc du capteur qui l'alimente.
- Le GND 0V à relier entre eux.
- Le pin trig à relier à une sortie digitale qui permet de piloter le capteur.
- Le pin echo à relier à une entrée digitale qui permet de recevoir les informations du capteur.

Le schéma de connexion entre l'arduino et le capteur influence le programme. Il est possible de changer les broches de connexion de trig et echo en prenant une broche

digitale de l'arduino. Le changement dans le programme se fera au niveau des lignes du define pour le trig et l'echo.

CODE GÉNÉRAL

Introduction

Le code général se sert du capteur ultrason, pour obtenir sa valeur il faut la convertir en cm puis l'afficher dans le moniteur série. L'actualisation des valeurs se fait toutes les 2 secondes. Le code a été fait en complet dans le main, il y a un autre code qui transforme celui-ci en fonction, on le verra un peu plus loin dans le document.

Define

```
//------------Define------------//
#define pinTrig 11 //Pin broche trig
#define pinEcho 12 //Pin broche echo
//----------------------------//
```

Les defines représentent les différentes liaisons entre la carte arduino et le capteur en physique. On reprend le schéma précédant et le pin Trig est connecté au pin digital 11 quant à echo il est connecté à la borne 12. Dans le cadre où on veut changer la connexion des pins il suffit de modifier le numéro ici. Le reste du code va appeler pinTrig et pinEcho.

Variable

```
//-----------Variable-----------//
long temps;//cette variable gère le temps de l'aller-retour du capteur
ultrason
float distance;//c'est variable sert à calculer la distance du capteur detecté
par le capteur ultrason.
//----------------------------//
```

Il y a deux variables dans le code qui vont être utilisé :

- temps : cette variable est la valeur (en temps) de l'aller-retour entre l'objet et le capteur ultrason. Si cette valeur est supérieure à 30 000 soit l'objet est trop loin et donc pas dans zone de mesure du capteur soit il n'y a pas d'objet.
- distance : cette variable est le résultat en cm entre le capteur et l'objet, si la mesure n'a pas échoué. Elle dépend directement de la variable temps.

Void setup

Une fois la création des variables et des defines réalisées, on va intialisé les entrées sorties ainsi que d'activer le moniteur série qui va afficher le résultat.

```
void setup() {

```

```
  //-----------Initialisation-----------//
  Serial.begin(9600);
  //activation du moniteur série à 9600 bauds
  //------------------------------------//

  //-----------Entrée/Sortie------------//
  pinMode(pinTrig, OUTPUT);
  pinMode(pinEcho, INPUT);
  //------------------------------------//

  //--------Initialisation-Sortie------//
  digitalWrite(pinTrig, LOW);
  //------------------------------------//

}
```

La première ligne active le moniteur série à 9600 bauds, il est possible de changer cette valeur mais celle-ci est conseillée. On initialise la broche pinTrig en tant que sortie car c'est elle qui permet de piloter le capteur ultrason. Le pin pinEcho est initialisé en tant qu'entrée car c'est là que le capteur enverra le signal de réponse. Puis, on initialise la sortie pinTrig pour être sûr de ne pas activer le capteur par erreur.

Void loop

Le but du programme dans la void loop est d'effectuer une mesure en activant le capteur ultrason puis de traiter l'information reçue. L'affichage se fera dans le moniteur série et on attendra 2 secondes avant de recommencer. Le temps d'attente peut être raccourcie.

```
void loop() {

  /*
   * Ce programme est un programme qui grâce un capteur
   * ultrason de déterminer la distance (ici en cm)
   *
   * Jérémy Clémente 16/05/2023
  */

  //---------Activation Capteur---------//
  digitalWrite(pinTrig, HIGH);
  delayMicroseconds(10);
  digitalWrite(pinTrig, LOW);
  //------------------------------------//

  //récupération valeur
  temps = pulseIn(pinEcho, HIGH);

  //---------Affichage Résultat--------//
```

```
  if (temps > 30000)//valeur du timeout donc échec de la mesure
  {
    Serial.println("Echec de la mesure");//affichage erreur de mesure
  }
  else
  {
    //division par 2 pour aller et retour
    temps = temps/2;
    //multiplication par 340 car vitesse du son en m/s division par 10000
    //pour avoir des cm .0 sert à avoir des nombres après la virgule.
    distance = (temps*340)/10000.0;
    //affichage
    Serial.print("Distance: ");
    Serial.print(distance);
    Serial.println(" cm");
  }
  //------------------------------------//

  //Attente avant nouveau cycle
  delay(2000);
}
```

Dans un premier temps, on met la sortie pinTrig à l'état haut pendant 10 microsecondes (durée vu dans la datasheet) pour activer le capteur. Ensuite, on compte le temps du capteur à l'état haut avec la fonction pulseIn(). Si le temps est supérieur à 30 000, la mesure est un échec et on affiche donc « Echec de la mesure » dans le moniteur série. Dans le cas où la mesure n'est pas un échec, on divise la variable par deux pour obtenir le temps que l'ultrason a mis pour l'aller car sinon on a l'aller-retour au niveau du temps.

Pour obtenir la distance on utilise la formule suivante :

$$vitesse = \frac{distance}{temps} \rightarrow distance = vitesse \times temps$$

La vitesse du son dans l'air étant de 340 m/s on prend cette vitesse comme référence et on multiplie le temps par 340. Puis, on convertit la mesure en cm en divisant par 10 000.0 le .0 permet d'obtenir un nombre à virgule dans le code.

Une fois la conversion effectuée, on affiche la valeur obtenue dans le moniteur série en précisant « Distance : » et « cm ». Puis on attend 2 secondes et on recommence le cycle.

CODE FONCTION US

Différence entre code

La grande différence entre le code complet avec affichage dans le moniteur série et la version du code en fonction est l'utilisation de la fonction. Celle-ci va renvoyer la valeur de la mesure en centimètre et si le résultat est un échec, elle renvoit -1. La raison du -1 pour un échec de mesure est qu'une mesure entre deux objet ne peut pas être négative.

Fonction Capteur US

```
int CapteurUS()
{
  /*
   * Cette fonction sert à déterminer la distance
   * d'un objet grâce à un capteur ultrason (ici en cm)
   * si la mesure est fausse on retourne -1
   *
   * Jérémy Clémente 16/05/2023
  */

  //-----------Variable-----------//
  long temps;//cette variable gère le temps de l'allé retour du capteur ultrason
  float distance;//c'est variable sert à calculer la distance du capteur detecter par le capteur ultrason.
  //------------------------------//

  //-----------Initialisation-----------//
  temps = 0;
  distance = 0;//on impose les variables à zéro
  //----------------------------------//

  //Début

  //---------Activation Capteur---------//
  digitalWrite(pinTrig, HIGH);
  delayMicroseconds(10);
  digitalWrite(pinTrig, LOW);
  //----------------------------------//

  //récupération valeur
  temps = pulseIn(pinEcho, HIGH);

  //------------- Résultat ------------//
  if (temps > 30000)//valeur du timeout donc échec de la mesure
  {
```

```
    distance = -1;//résultat en cas d'erreur de mesure
  }
  else
  {
    //division par 2 pour aller et retour
    temps = temps/2;
    //multiplication par 340 car vitesse du son en m/s division par 10000
    //pour avoir des cm .0 sert à avoir des nombres après la virgule.
    distance = (temps*340)/10000.0;
  }
  //-----------------------------------//

  //On retourne la valeur
  return distance;

  //Fin

}
```

La fonction crée, les deux variables locales « temps » et « distance » vu précédemment. Elle possède le même programme que précédemment. Elle a toujours besoin des define pinTrig et pinEcho pour utiliser le capteur US, il faudra donc les définir en début de programme.

Comme dit précédemment, la fonction renvoit la distance entre l'objet et le capteur via le return distance. La mesure s'effectuera à chaque appel de la fonction. Il faudra veiller à ne pas l'utiliser trop rapidement pour éviter que les mesures se parasitent entre elles. La durée attente entre deux mesures est de préférence au moins 60 millisecondes.

CONCLUSION

Voici la fin du rapport sur l'utilisation du capteur US via une arduino uno. Mais il est tous à fait possible d'utiliser une autre carte arduino. Le lien vers le code source et les autres documentations sont dans le qr code en bas de la conclusion. Il serait aussi possible d'améliorer le code en intégrant un capteur de température qui permettrait d'obtenir la température de l'air et donc de calculer plus précisément la valeur de la vitesse de son dans l'air.

QR CODE