Rahmani Aubin Peip 2

Rapport de SEANCE 10/01/2022

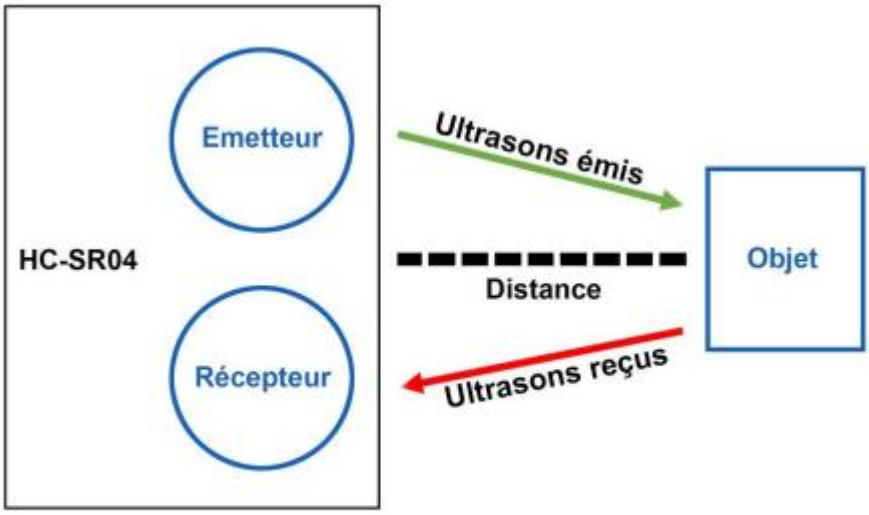
A – Creation d’un capteur de vitesse

Pour répondre à la problématique du radar et du calcul de la vitesse d’un véhicule nous avons comme choix d’utiliser un module laser ou un module qui émet des vagues d’ultrasons.

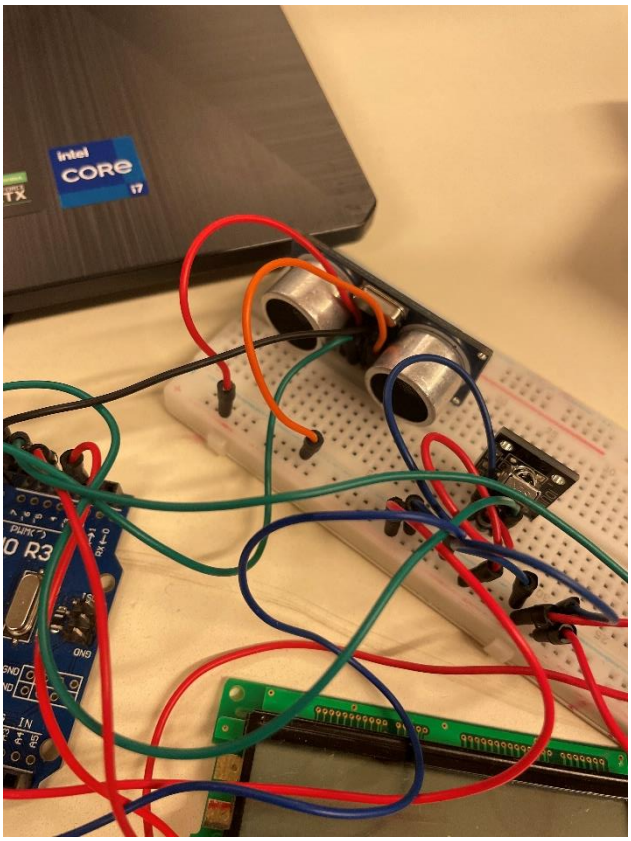
Le module hc-sr04 permet d’evaluer les distances entre un objet mobile et les obstacles rencontrés avec une amplitude de 3 metre maximum.

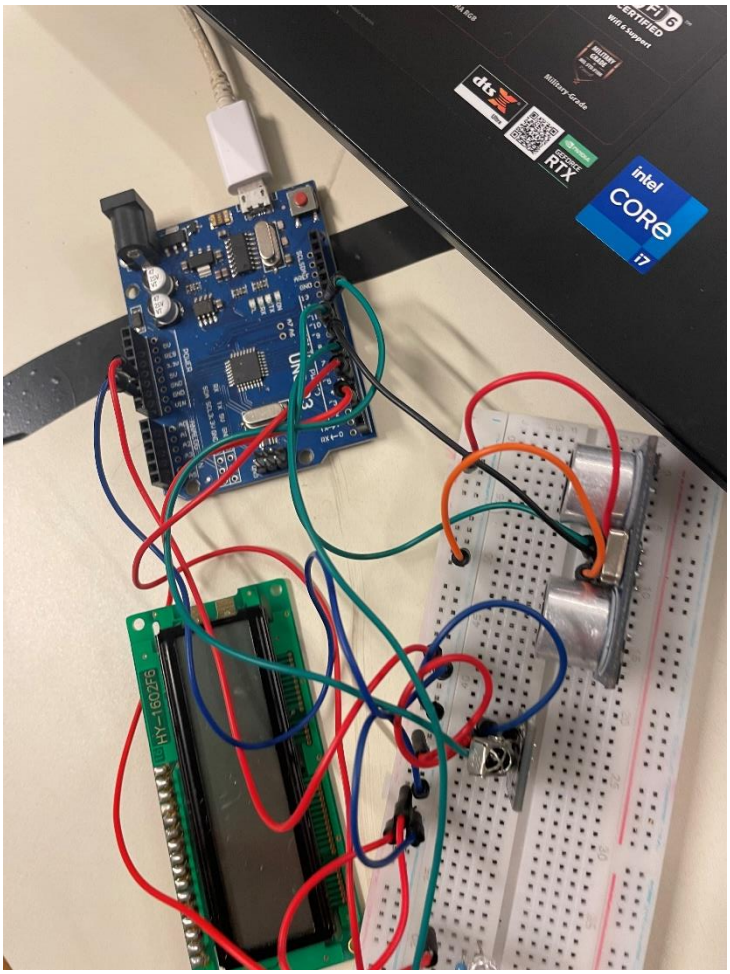
Il y a un emetteur et un recepteur, l’emetteur est branché sur la broche Trigger qui va envoyé un signal de 10 micro seconde afin d’obtenir des ultrasons a 40kHz.

Le récepteur réceptionne les ultras sons réfléchis sur l’objet, ce recpeteur est branché sur la broche Echo.

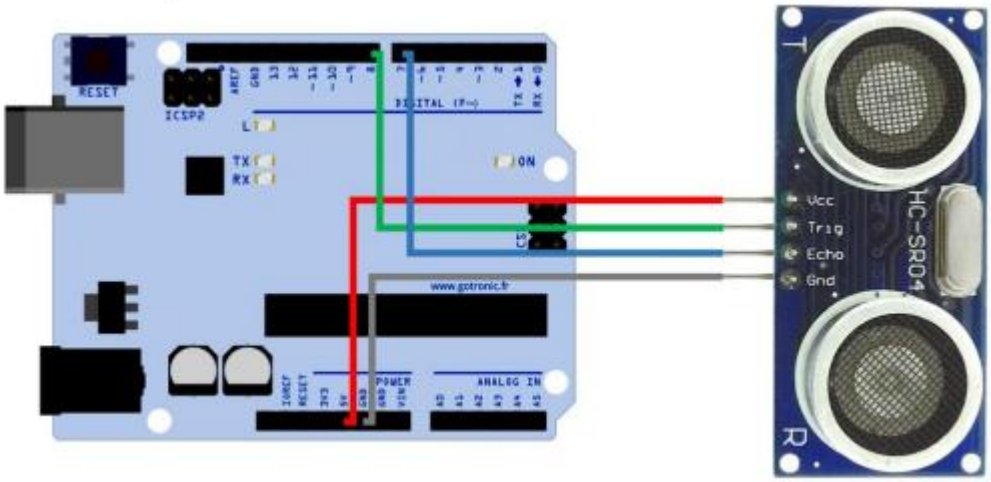


Nous branchons donc le fil rouge sur la broche des 5 Volts, le fil orange sur la branche GND et le fil vert (trig) la sortie 8 et le fil noir (echo) sur la sortie 7





On pouvait brancher avec des fils mâles/femelles comme ceci :



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

Maintenant place au code, nous voulons obtenir la vitesse mais nous connaissons que la durée totale et la vitesse du son (0,034 cm/microSecondes). Pour obtenir la distance il suffit donc de multiplier la durée allé retour par la vitesse du son.

Ce qui nous donne une distance en cm ! Il va falloir donc convertir en metre ou en kilometre.

```
#define Broche_Echo 7 // Broche Echo du HC-SR04 sur D7 //
#define Broche_Trigger 8 // Broche Trigger du HC-SR04 sur D8 //
// Definition des variables
int MesureMaxi = 300; // Distance maxi a mesurer //
int MesureMini = 3; // Distance mini a mesurer //
long Duree;
long Distance;


void setup()
{
  pinMode(Broche_Trigger, OUTPUT); // Broche Trigger en sortie //
  pinMode(Broche_Echo, INPUT); // Broche Echo en entree //
  Serial.begin (115200);
}


void loop() {
  // Debut de la mesure avec un signal de 10 µS applique sur TRIG //
  digitalWrite(Broche_Trigger, LOW); // On efface l'etat logique de TRIG //
  delayMicroseconds(2);
  digitalWrite(Broche_Trigger, HIGH); // On met la broche TRIG a "1" pendant 10µS //
  delayMicroseconds(10);
  digitalWrite(Broche_Trigger, LOW); // On remet la broche TRIG a "0" //

  // On mesure combien de temps le niveau logique haut est actif sur ECHO //
  Duree = pulseIn(Broche_Echo, HIGH);

  // Calcul de la distance grace au temps mesure //
  Distance = Duree * 0.034 / 2; // *** voir explications apres l'exemple de code *** //

  // Verification si valeur mesuree dans la plage //

  if (Distance >= MesureMaxi || Distance <= MesureMini) {
    // Si la distance est hors plage, on affiche un message d'erreur //
    Serial.println("Distance de mesure en dehors de la plage (3 cm à 3 m)");
  }
  else {
    // Affichage dans le moniteur serie de la distance mesuree //
    Serial.print("Distance mesuree :");
    Serial.print(Distance);
    Serial.println("cm");
  }
  delay(1000); // On ajoute 1 seconde de delais entre chaque mesure


}
```

Voici un premier code qui permet de calculer la distance en fonction de la vitesse du son et de la durée. Maintenant pour calculer ma vitesse il suffit coder la relation suivante : V = D/T.

Et c'est la mon erreur. J'avais une vitesse fixe en m/s et en km/h. J'ai tout converti et ait mis comme code (en initialisant la variable vitesse)

Vitess = DistanceKm/DureeH

Or la distance dependant déjà de la durée des ondes ultrasons, il était logique que ma vitesse soit constante.

J'ai passé beaucoup de temps avec mon camarade afin de comprendre ou était le probleme et on avait pensé à une erreur de conversion. Je n'ai pas sauvegardé les programmes de mes erreurs cependant nous avons trouvé une solution pour obtenir la vitesse.

Vitesse = deltaD/deltaT

Il suffit de prendre deux points, donc deux distances, de les soustraires et enfin soustraire les deux temps obtenus pour chaque point.

```
#define Broche_Echo 7 // Broche Echo du HC-SR04 sur D7 //
#define Broche_Trigger 8 // Broche Trigger du HC-SR04 sur D8 //
// Definition des variables
const int MesureMaxi = 300; // Distance maxi a mesurer //
const int MesureMini = 3; // Distance mini a mesurer //

int distanceCm()
{
  // Debut de la mesure avec un signal de 10 µS applique sur TRIG //
  digitalWrite(Broche_Trigger, LOW); // On efface l'etat logique de TRIG //
  delayMicroseconds(2);
  digitalWrite(Broche_Trigger, HIGH); // On met la broche TRIG a "1" pendant 10µS /
  delayMicroseconds(10);
  digitalWrite(Broche_Trigger, LOW); // On remet la broche TRIG a "0" //
  // On mesure combien de temps le niveau logique haut est actif sur ECHO //
  const float DureeMicro = pulseIn(Broche_Echo, HIGH);

  Serial.print("La durée est de :");
  Serial.print (" ");
  Serial.print(DureeMicro);
  Serial.println("µs");
  Serial.println(" ");

  // Calcul de la distance grace au temps mesure //

  const float DistanceCm  = DureeMicro * 0.034 / 2.0; // (Durée totale (aller-retou
  // Verification si valeur mesuree dans la plage //

  if (DistanceCm  >= MesureMaxi || DistanceCm  <= MesureMini) {
    // Si la distance est hors plage, on affiche un message d'erreur //
    Serial.println("Distance de mesure en dehors de la plage (3 cm à 3 m)");
  }
  else {
    // Affichage dans le moniteur serie de la distance mesuree //
    Serial.print("Distance mesuree :");
    Serial.print(DistanceCm );
    Serial.println("cm");
    Serial.println(" ");
  }

  return DistanceCm;
}

void setup()
{
  pinMode(Broche_Trigger, OUTPUT); // Broche Trigger en sortie //
  pinMode(Broche_Echo, INPUT); // Broche Echo en entree //
  Serial.begin (115200);
}
```

```
void loop() {

  const float distCm1 = distanceCm();
  const float deltaT = 2000; // temps avant prochaine mesure
  delay(deltaT);
  const float distCm2 = distanceCm();


  //Serial.println(distCm1);
  //Serial.println(distCm2);

  const float deltaD = distCm1-distCm2;
//  Serial.println(deltaD);
  const float vitesseS = deltaD / deltaT * 10;
  const float vitesse = vitesseS * 3.6;

  Serial.print("la vitesse est ");
  Serial.print(vitesse);
  Serial.println( "km/h");
  Serial.println(" ");
  Serial.print("la vitesse est ");
  Serial.print(vitesseS);
  Serial.println( "m/s");
  Serial.println(" ");



  delay(1000); // On ajoute

}
```

J'ai donc crée une fonction distance en cm, qui prenait donc en compte les valeurs et les calculs afin d'obtenir une distance pour un point 1.

Dans le loop, j'ai créer deux constantes contenant la fonction distance afin d'obtenir deux distance. J'ai mo même choisi un temps de 2000 micro secondes, afin d'être sure d'avoir une différence importante.

J'ai créer ensuite une constante VitesseS en convertissant ma vitesse en m/s

De même pour la vitesse en km/h (on rappelle que pour convertir des m/s en km/h, il faut multiplier par 3.6).

B <u>- Pour la prochaine séance</u>

Je n’ai pas de problème à régler pour la prochaine séance ou a la maison, je vais donc prendre un écran LCD, et afficher toutes mes valeurs dont la vitesse sur celui-ci afin de simuler un radar pédagogique visible par le conducteur car il est très important pour le conducteur de connaître sa vitesse

Le plus gros problème que j’ai eu dans ma séance est de faire mes calculs pour convertir et de trouver le vrai calcul de la vitesse