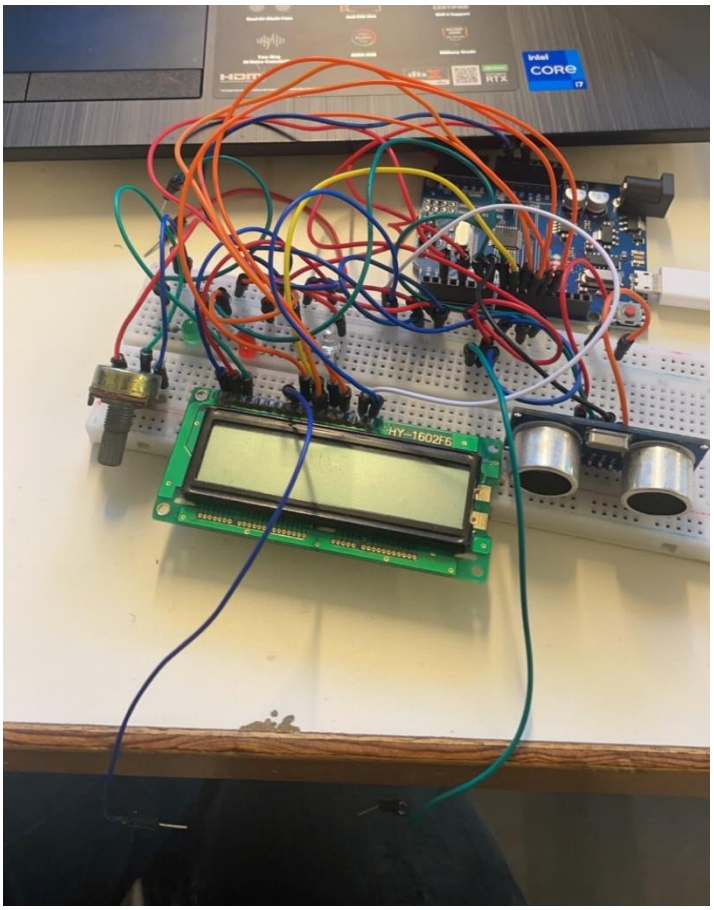
Rahmani Aubin Peip2 G1

RAPPORT DE SEANCE DU PROJET ARDUINO 10/02/2021

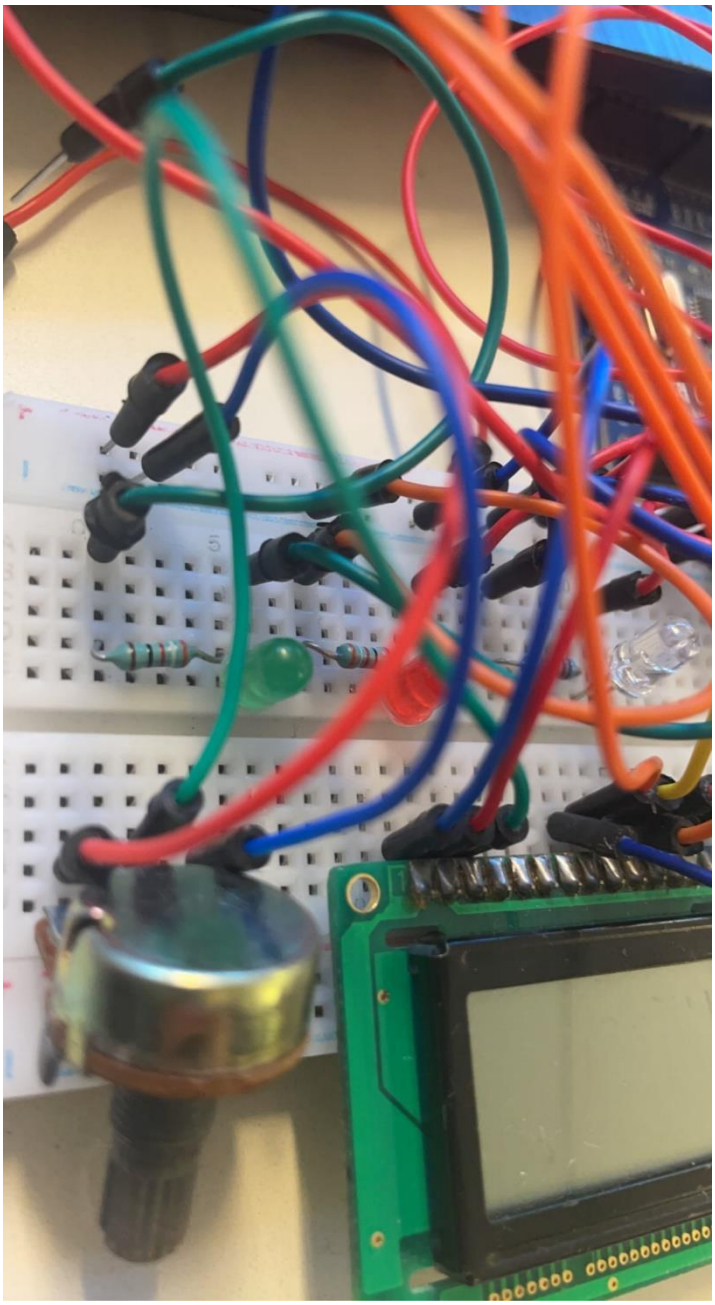
A- Finalisation du radar pédagogique

Bien que la séance dernière ne fut pas très fructueuse, cette séance fut nettement meilleure avec de bon résultats.

J'ai commencé par tout débrancher et tout rebrancher les fils de l'écran lcd. Je l'ai fait minutieusement. Cela m'a pris un peu de temps mais au moins j'étais sûre que tout était branché correctement.



Cette fois ci j'ai bien branché mon potentiomètre



J’ai toujours gardé mon code couleur pour les fils : rouge – 5V

Bleu – terre

vert, orange, jaune, sortie(Output)

Cependant mon écran lcd ne s’allumait toujours pas. J’ai donc décidé de commencer à programmer un programme simple qui me permet d’afficher un message quelconque.

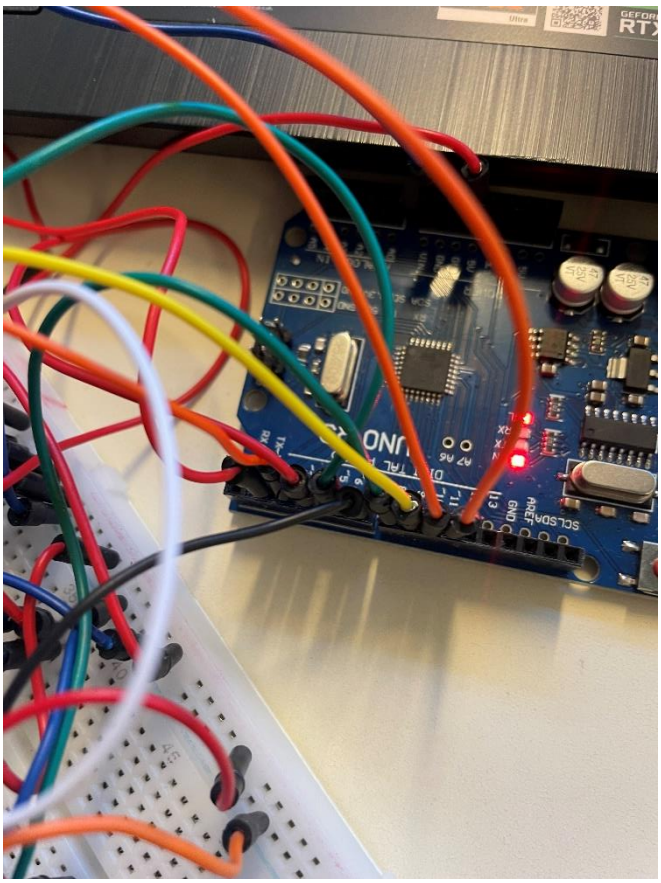
radar_affichage §

```
//j'affiche le lcd
//liquide crystal i2c
#include<LiquidCrystal.h>

LiquidCrystal lcd(2,3,4,9,10,11); // initialise les commandes avec mes numéro de broches

void setup() {
  lcd.begin(16,2); // initialiser le nombre de colonnes et de lignes (c'est un lcd 16 *2)
  lcd.print("Votre vitesse est de :");
  lcd.setCursor(0,1);// le curseur se positionne à la lère colonne, 2ième ligne
  lcd.print("Veuillez ralentir - dos d'ane");

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

On fait donc appel à la librairie « LiquidCrystal »

Ici vu qu'on souhaite seulement afficher un message fixe on n'écris pas de message dans le loop.

J'initialise les broches (2,3,4,9,10,11) car j'ai branché les broches sur ces sorties.



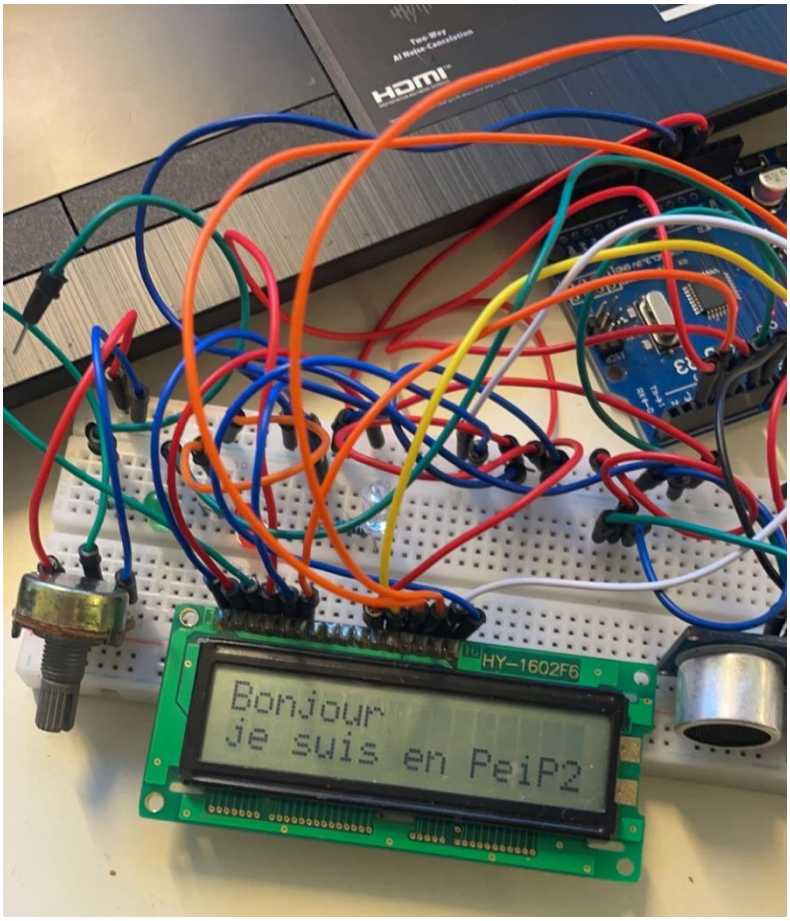
J’ai tout d’abord mis « Bonjour »

Et ensuite « je suis en Peip2 »

Cependant après le transfert et la compilation rien ne s’affiche.

Après plusieurs vérification, c’est le potentiomètre qui fonctionnait mal. Je l’ai donc changé et j’ai donné une valeur de 5v (j’ai tourné au maximum).

Et miracle j’obtient mon message :

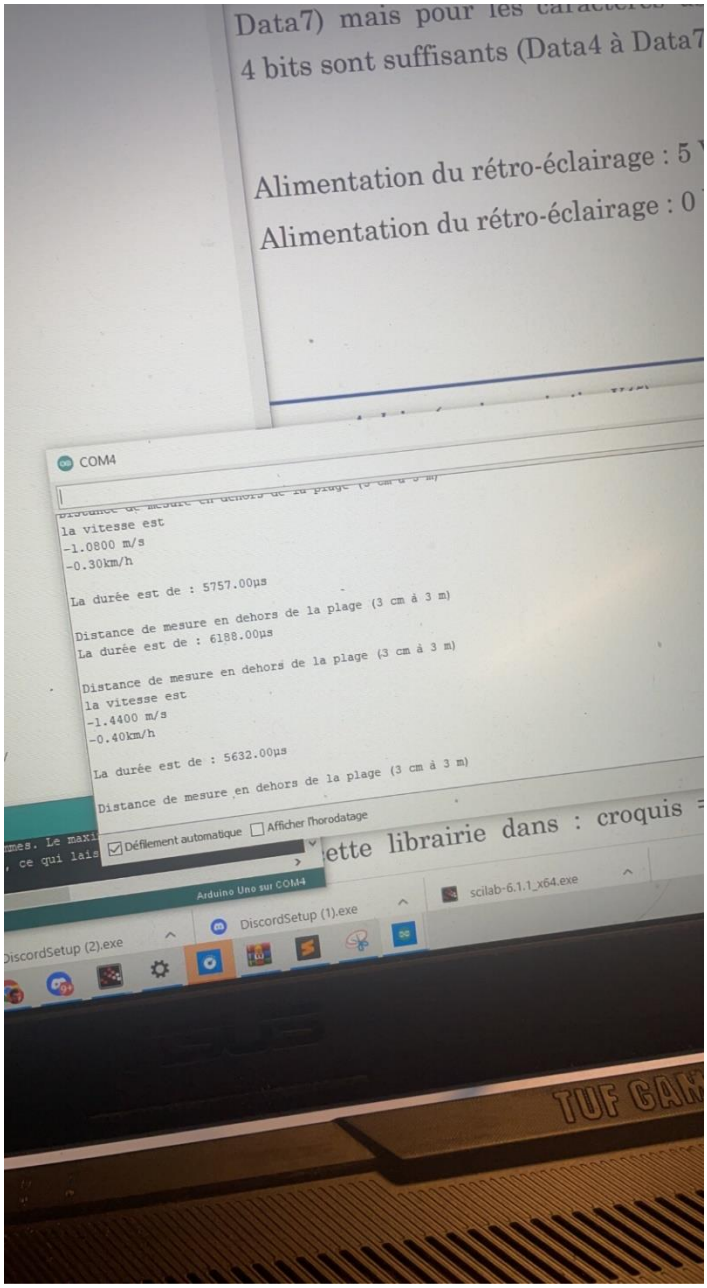


Maintenant il faut que j’affiche ma vitesse constamment sur l’écran avec un message personnalisé en plus !

Je vérifie tout d’abord si mon programme qui me permet d’avoir la vitesse est optimale.

Et il ne l’était pas, j’avais confondu par erreure km/h et m/s. J’ai donc changé tout ca.

Cependant quand j’affiche ma vitesse sur l’écran, elle reste fixe. Alors que dans le terminal elle change constamment (si j’ai des valeurs négatives c’est parce que je reculais)



```
void loop() {

  float distCm1 = distanceCm();
  float deltaT = 500; //temps avant prochaine mesure
  delay(deltaT);
  float distCm2 = distanceCm();
  float deltaD = distCm1 - distCm2;

  float vitesseS = deltaD / deltaT * 10;
  float vitesse = vitesseS * 3.6;



  lcd.setCursor(0, 0);
  lcd.print("Speed : ");
  lcd.print(vitesseS,2);
  lcd.print("km/h");

  Serial.println("la vitesse est ");
  Serial.print(vitesse,4);
  Serial.println(" m/s");
  Serial.print(vitesseS);
  Serial.println("km/h");
  Serial.println(" ");
```

J'avais bien mis dans loop, cependant j'avais fais une erreure. J'avais mis des const float au lieu de float tout court.

Une fois modifié j'ai bien eu ma vitesse qui change constamment (plus ou moins rapide en fonction du temps entre deux mesures du deltaT) .

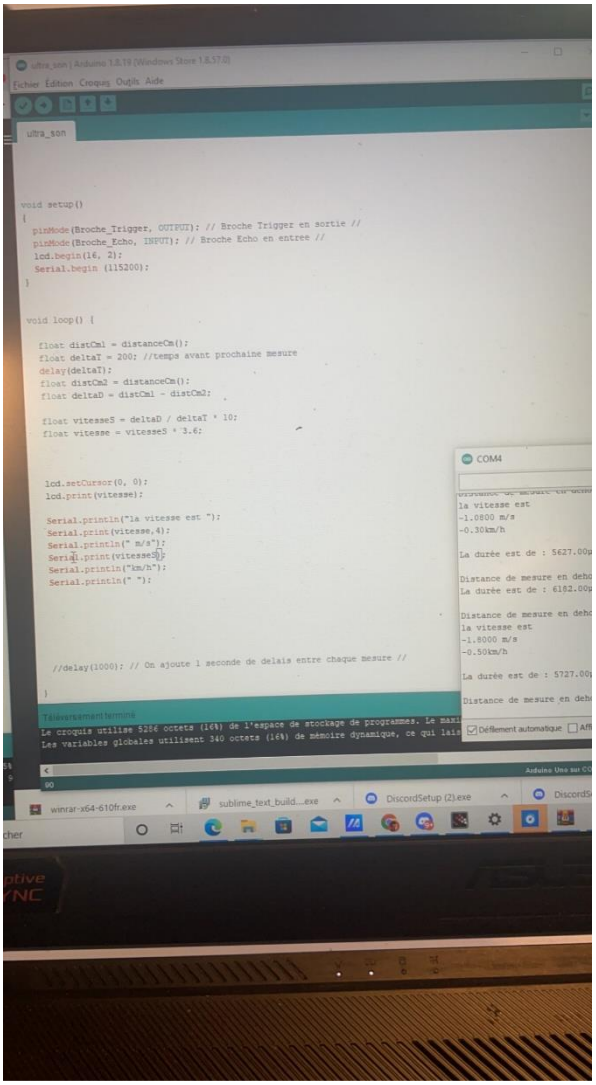
Autre petit problème, l'écran affiche la vitesse mais avec beaucoup de chiffres après la virgule.

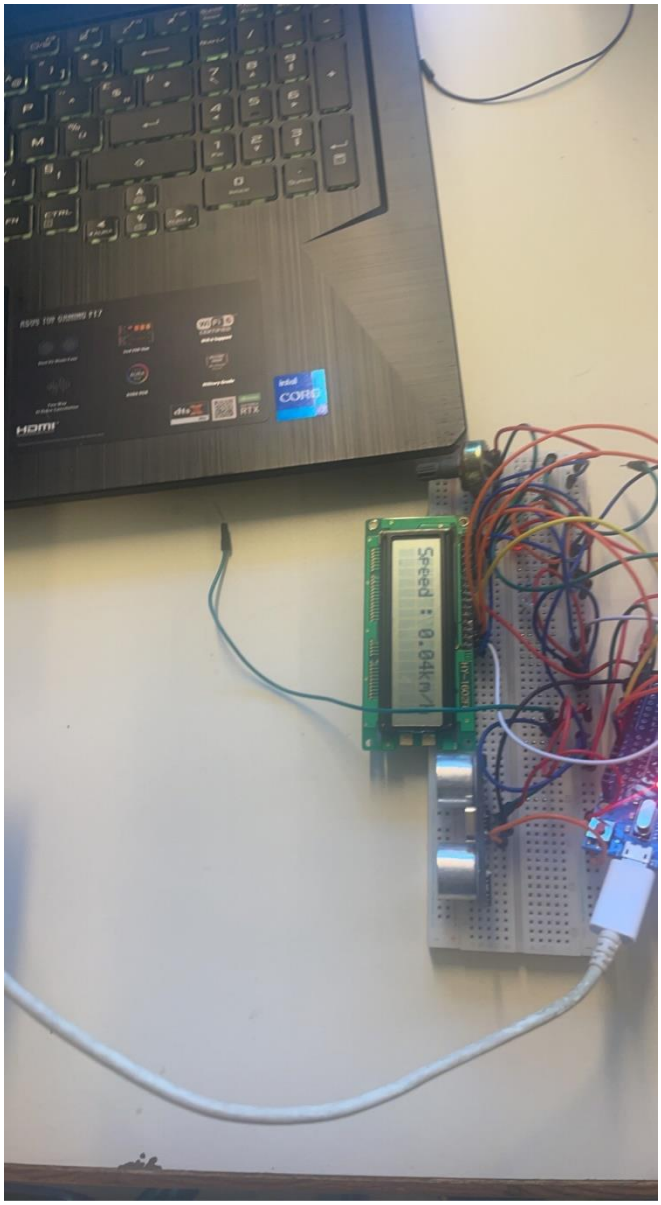
Il suffisait de mettre le nombre de chiffre significatif que l'on voulait avoir dans le lcd.print

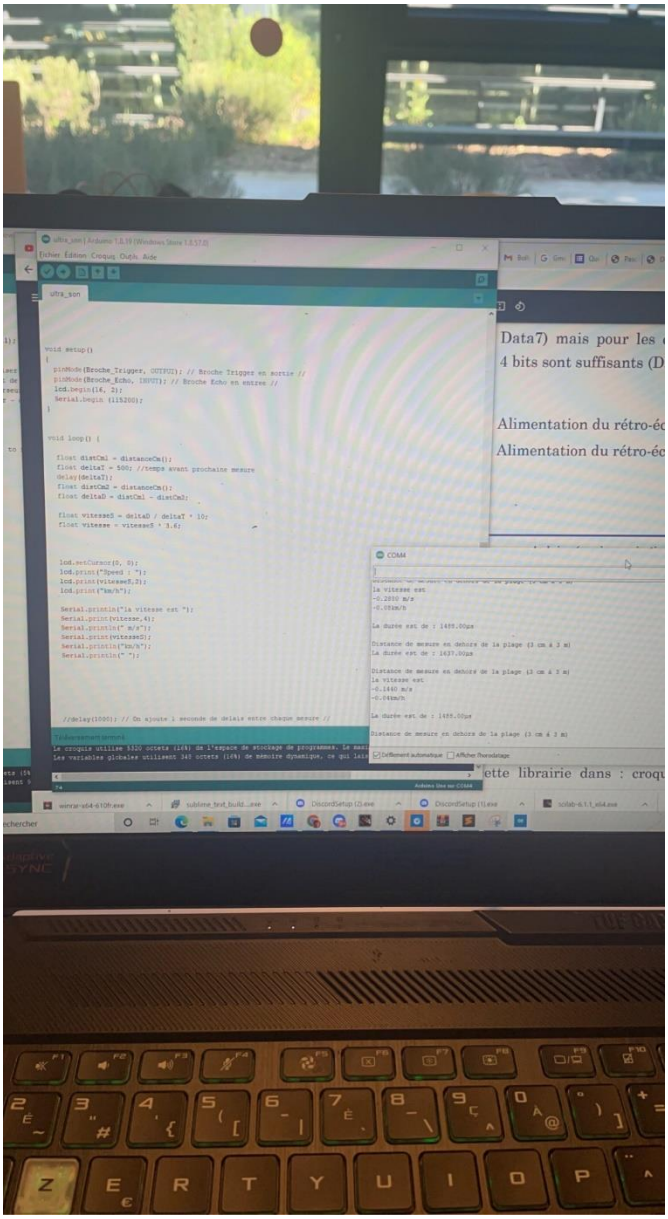
Ici 2.

Ma vitesse s'affiche bien, change constamment selon la vitesse et est bien en km/h.

J'ai choisi de mettre le mot « speed » car nous avons seulement 16 pixels de libres et la vitesse en prend déjà minimum 4.


ultra_son | Arduino 1.8.19 (Windows Store 1.8.57.0)
Fichier Edition Croquis Outils Aide
ultra_son
void setup()
{
pinMode(Broche_Trigger, OUTPUT); // Broche Trigger en sortie //
pinMode(Broche_Echo, INPUT); // Broche Echo en entree //
lcd.begin(16, 2);
Serial.begin (115200);
}
void loop() {
float distCm1 = distanceCm();
float deltaT = 200; //temps avant prochaine mesure
delay(deltaT);
float distCm2 = distanceCm();
float deltaD = distCm1 - distCm2;
float vitesseS = deltaD / deltaT * 10;
float vitesse = vitesseS * 3.6;
lcd.setCursor(0, 0);
lcd.print(vitesse);
Serial.println("la vitesse est ");
Serial.print(vitesse,4);
Serial.println(" m/s");
Serial.print(vitesseS);
Serial.println("km/h");
Serial.println(" ");
//delay(1000); // On ajoute 1 seconde de delais entre chaque mesure //
}
Téléversement terminé
Le croquis utilise 5284 octets (16%) de l'espace de stockage de programmes. Le maxi
Les variables globales utilisent 340 octets (16%) de mémoire dynamique, ce qui lais
COM4
la vitesse est
-1.0800 m/s
-0.30km/h
La durée est de : 5627.00
Distance de mesure en deh
La durée est de : 6182.00
Distance de mesure en deh
la vitesse est
-1.8000 m/s
-0.50km/h
La durée est de : 5727.00
Distance de mesure en deh
Défilement automatique
winrar-x64-610fr.exe
sublime_text_build....exe
DiscordSetup (2).exe


Speed : 0.04km/



Tout est donc fonctionnel et le programme est quasimément parfait. J’ai rajouterai un if avec deux cas de figure. Si la voiture roule trop vite, on affichera « ralentissez »,et si la voiture roule à la bonne vitesse, on affichera « bonne vitesse »

ultra_son | Arduino 1.8.19 (Windows Store 1.8.57.0)

Fichier Édition Croquis Outils Aide

ultra_son

```
#include<LiquidCrystal.h>

#define Broche_Echo 7 // Broche Echo du HC-SR04 sur D7 //
#define Broche_Trigger 8 // Broche Trigger du HC-SR04 sur D8 //

LiquidCrystal lcd(2, 3, 4, 9, 10, 11);


// Definition des variables
int MesureMaxi = 300; // Distance maxi a mesurer //
int MesureMini = 3; // Distance mini a mesurer //

boolean bobo = true;
long Duree;
long Distance;
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-retour))

  // 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.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 //
  lcd.begin(16, 2);
  Serial.begin (115200);
}


void loop() {

  float distCm1 = distanceCm();
  float deltaT = 500; //temps avant prochaine mesure
  delay(deltaT);
  float distCm2 = distanceCm();
  float deltaD = distCm1 - distCm2;

  float vitesseS = deltaD / deltaT * 10;
  float vitesse = vitesseS * 3.6;



  lcd.setCursor(0, 0);
  lcd.print("Speed : ");
  lcd.print(vitesseS,2);
  lcd.print("km/h");

  Serial.println("la vitesse est ");
```

```
void setup()
{
  pinMode(Broche_Trigger, OUTPUT); // Broche Trigger en sortie //
  pinMode(Broche_Echo, INPUT); // Broche Echo en entree //
  lcd.begin(16, 2);
  Serial.begin (115200);
}


void loop() {

  float distCm1 = distanceCm();
  float deltaT = 500; //temps avant prochaine mesure
  delay(deltaT);
  float distCm2 = distanceCm();
  float deltaD = distCm1 - distCm2;

  float vitesseS = deltaD / deltaT * 10;
  float vitesse = vitesseS * 3.6;



  lcd.setCursor(0, 0);
  lcd.print("Speed : ");
  lcd.print(vitesseS,2);
  lcd.print("km/h");

  Serial.println("la vitesse est ");
  Serial.print(vitesse,4);
  Serial.println(" m/s");
  Serial.print(vitesseS);
  Serial.println("km/h");
  Serial.println(" ");







  //delay(1000); // On ajoute 1 seconde de delais entre chaque mesure //

}
```

Pour la prochaine fois j'utiliserai un écran lcd avec seulement 4 sorties car avec celui-ci j'utilise beaucoup trop de sorties.

recorded-2006078676883.MP4

J'utiliserai un « liquide crystal i2c »