NOM : ..... PRENOM : ..... CLASSE : .....

PREMIERE SPECIALITE Physique et chimie	CHAPITRE O1 ONDES MECANIQUES TP 20 : Télémètre - Arduino

CORRECTION

Objectifs

Reconstituer un radar de recul automobile simplifié à l'aide d'un microprocesseur Arduino.

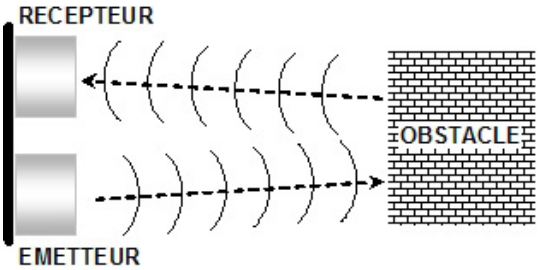
/30
/20

Le cahier des charges est le suivant :

→ le système devra émettre un son ainsi qu'un signal lumineux rouge si la distance est inférieure à 100cm

DOCUMENTS

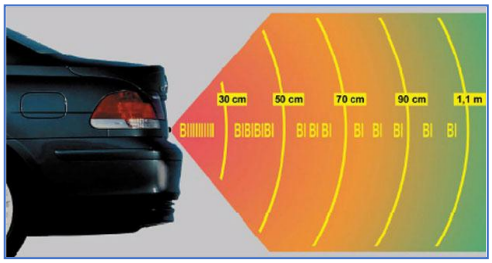
DOC 1 : Qu'est-ce qu'un télémètre ?



Un télémètre un appareil qui permet de mesurer des **distances** à l'aide d'une onde ultrasonore ou lumineuse.

Pour cela, l'émetteur envoie une est brève impulsion vers la cible et en chronométrant la **durée** mise par l'onde pour parcourir l'aller-retour on peut calculer la distance entre l'émetteur-récepteur et la cible.

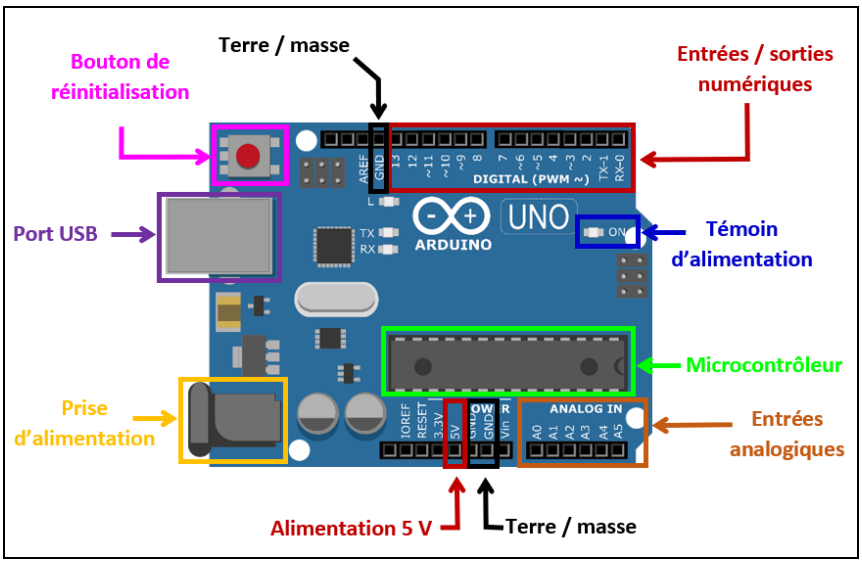
C'est le principe des radars de recul des automobiles



Les principaux constituants de ce type de système sont :

- Les capteurs à ultrasons, composés d'émetteurs et de récepteurs
- Le calculateur, qui estime la distance entre l'obstacle et l'arrière du véhicule, à partir des informations délivrées par les capteurs à ultrasons
- Un système d'émission sonore (parfois accompagné de signal lumineux) qui avertit le conducteur. Habituellement, un bip sonore est émis lorsque la distance obstacle/véhicule devient inférieure à une distance limite, puis, plus l'obstacle se rapproche, plus la fréquence des bips est élevée, jusqu'à devenir une émission sonore continue lorsque le véhicule est très proche de l'obstacle

DOC 2 : Qu'est-ce qu'un microcontrôleur Arduino ?



Un microcontrôleur est un microprocesseur que vous **programmer** :

→ vous allez écrire ou modifier quelques lignes de programmes, il va les interpréter et appliquer ce que vous lui avez demandé de faire.

EXEMPLES :

« S'il fait trop chaud ou trop ensoleillé, j'aimerais que tu baisses les rideaux ! » ;

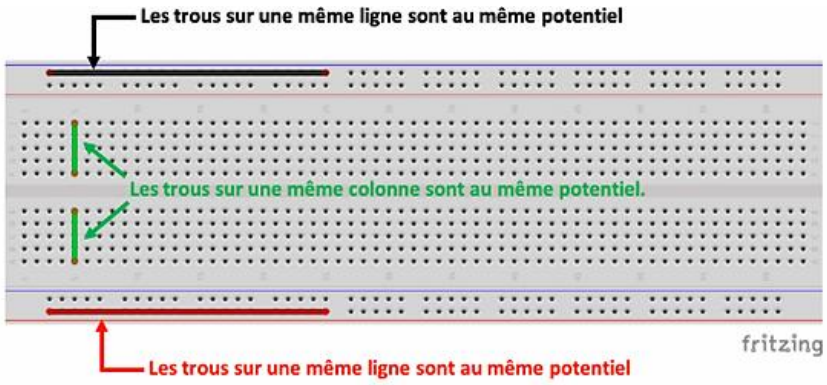
« Si le niveau d'eau est bas, rajoute de l'eau » etc…

Pour ce faire, il faudra que vous communiquiez avec le même langa
capteurs (de température, d'éclairement) et des actionneurs (moteurs, buzzer, LED)

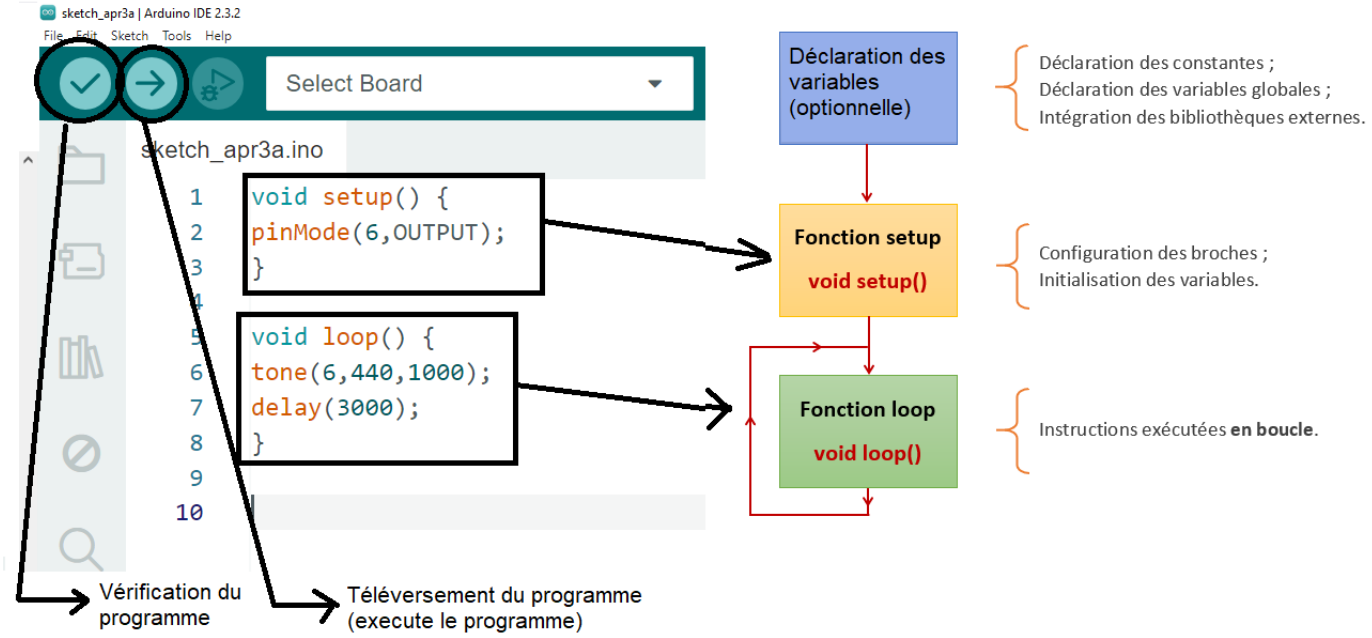
DOC 3 : Matériel = la platine d'essais

La platine d'essai est une plaque où l'on dispose les composants.

- Sur les 2 premières lignes et les 2 dernières on connecte l'alimentation 5 V et la masse (GND)
- Sur les lignes du centre on connecte les différents capteurs ou actionneurs



DOC 4 : La programmation



LES FONCTIONS :

La fonction **noTone** : stoppe la génération d'impulsion produite par l'instruction tone(). N'a aucun effet si aucune impulsion n'est générée. Il faut juste indiquer la sortie entre les parenthèses **noTone(6)**

La fonction **digitalWrite** : met un niveau logique HIGH (HAUT en anglais) ou LOW (BAS en anglais) sur une broche numérique. Si la broche a été configurée en SORTIE avec l'instruction pinMode(), sa tension est mise à la valeur correspondante : 5V pour le niveau HAUT, 0V (masse) pour le niveau BAS.

La fonction **delay()** : sert à mettre en pause le programme téléversé sur la carte pendant le temps spécifié entre les parenthèses. Ce nombre est exprimé en millisecondes

SYNTAXE D'UNE BOUCLE

```
if (condition)
{
    Action à réaliser si la condition est vraie
}
else
{
    Action à réaliser sinon
}
```

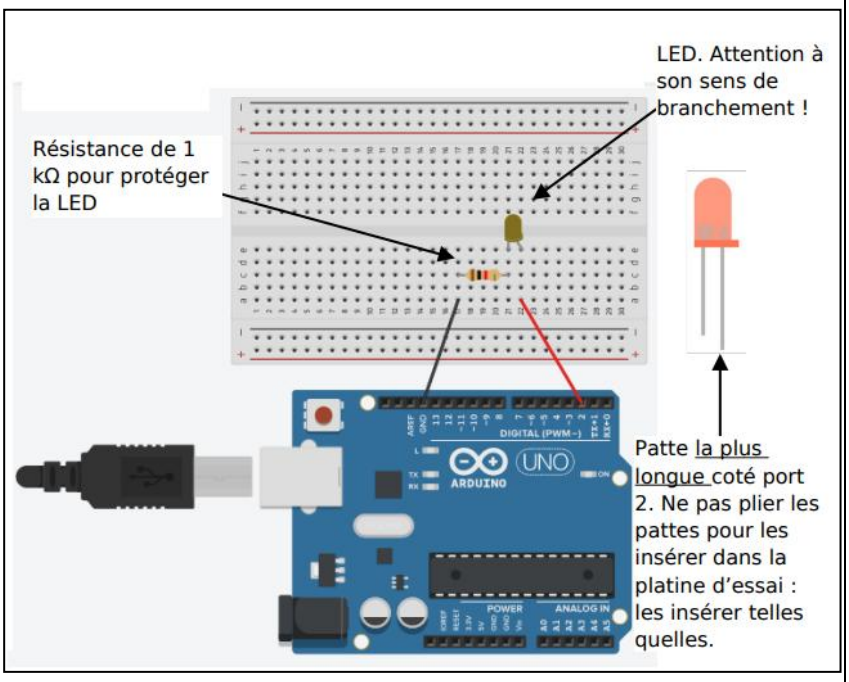
Remarque : Elle s'intègre à l'intérieur de la void loop

TRAVAIL A FAIRE

DECOUVERTE DU MICOCONTROLEUR ARDUINO

1 **EXPERIMENTATION** : allumer une LED

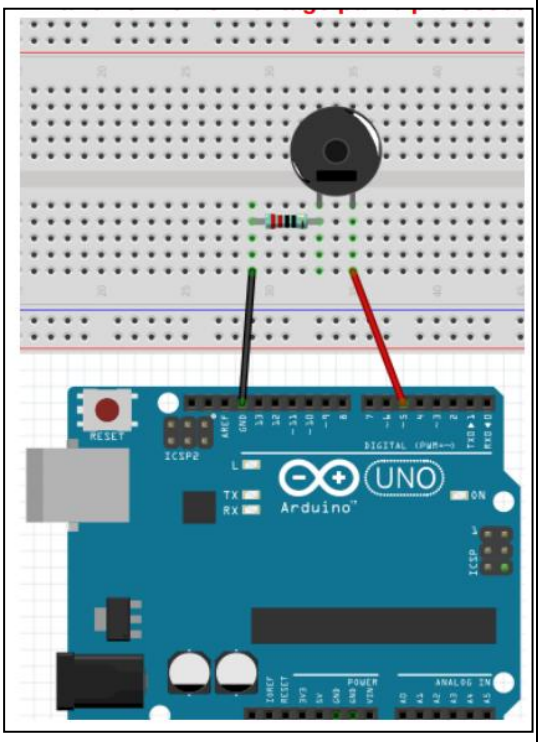
- Câbler le circuit ci-contre comportant une résistance et une LED sans brancher le câble USB à l'ordinateur.
- Appeler par le professeur pour vérifier le montage
- Ouvrir le programme Arduino
- A l'aide des documents, écrire un programme permettant de faire clignoter votre LED.
  → Elle doit s'allumer pendant 2 secondes et s'éteindre pendant 2 secondes.
- Téléverser le programme et vérifiez que la LED clignote.
- Modifier le programme de façon à ce que la LED rouge clignote
  → Elle doit s'allumer pendant 3 secondes et s'éteindre pendant 1/2 seconde
- Enregistrer votre programme sur votre session sous le nom « led »



2 **EXPERIMENTATION** : produire un son

- Câbler le circuit ci-contre comportant une résistance et un buzzer sans brancher le câble USB à l'ordinateur.

  Attention au branchement : Ne pas essayer de plier les pattes du buzzer : les insérer telles quelles sur la platine d'essai
- Appeler par le professeur pour vérifier le montage
- Ouvrir un nouveau programme Arduino et écrire les lignes de code afin de produire un La3 (440Hz)
- Téléverser votre programme
- Enregistrer votre programme sur votre session sous le nom « son »



REALISATION DU TELEMETRE

PRINCIPE basé sur l'écholocation :

ETAPE 1

Une impulsion ultrasonore de 10 µs est envoyée sur la broche TRIGGER du capteur.

ETAPE 2

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

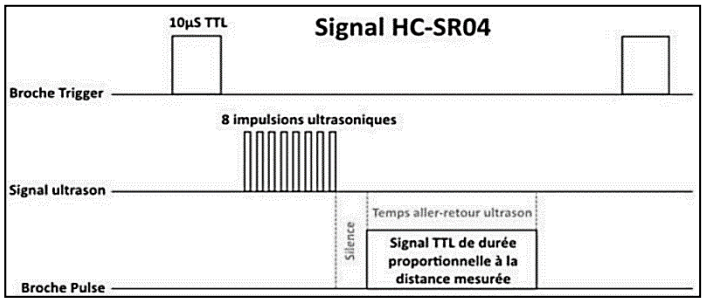
ETAPE 3

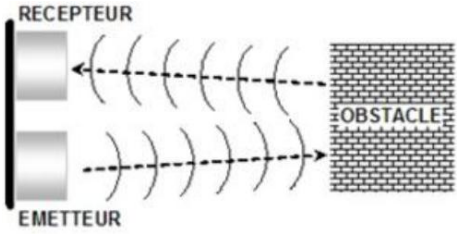
Les ultrasons se propagent dans l'air jusqu'à l'obstacle et sont réfléchis en direction du capteur.

ETAPE 4

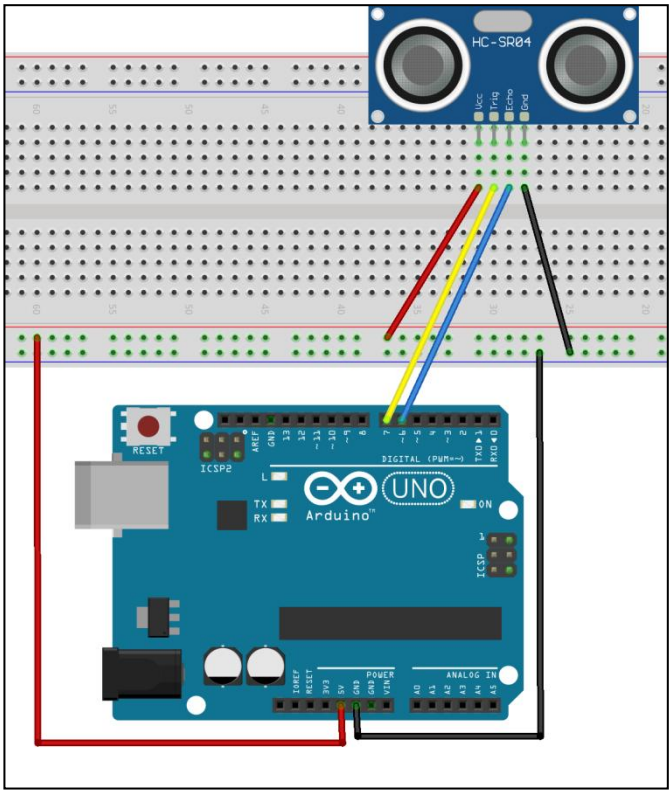
Le récepteur détecte alors l'écho et stoppe l'acquisition des mesures.

La durée écoulée entre l'émission et la réception du signal permet de déterminer la distance entre le capteur et l'obstacle.





3 **EXPERIMENTATION** : brancher le télémètre

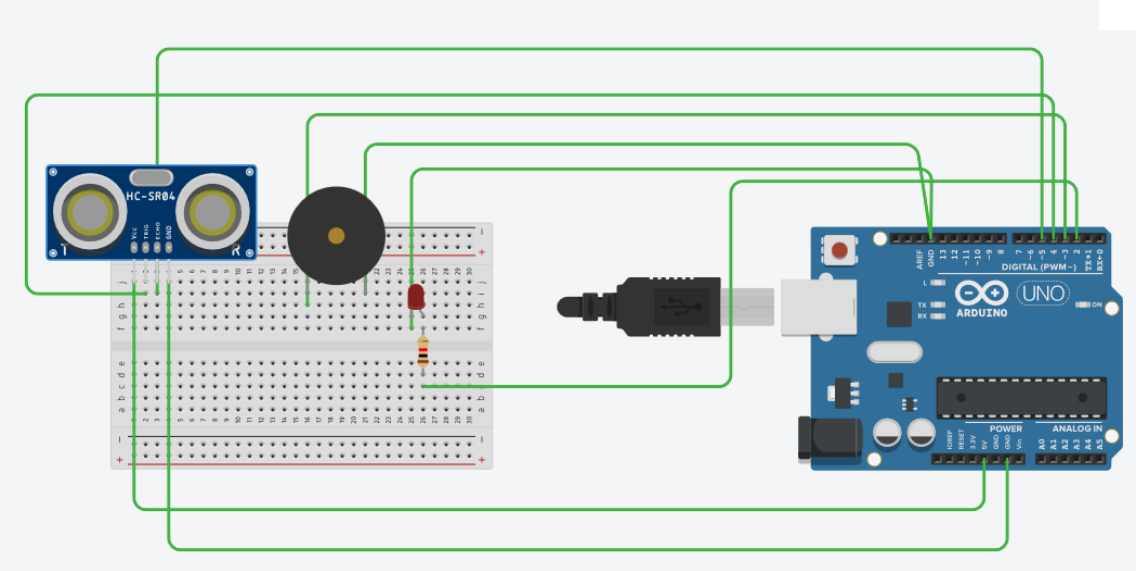


- ☐ Sur la platine d'essais, relier une broche +5V d'Arduino à la ligne « + » et une broche « GND » (c'est la masse ou 0V) à la ligne « - ».
- ☐ Placer l'émetteur ultrasons sur la plaque sur 4 colonnes, comme indiqué ci-contre.
- ☐ Relier ensuite la broche « GND » de l'émetteur à la ligne « - » de la plaquette, la broche « VCC » à la borne « + » de la plaquette, la borne « Trig » à la broche n°4 et la borne « Echo » à la broche n°5.
- ☐ Appeler le professeur pour lui faire vérifier le montage
- ☐ Copier le fichier : radar_recul.ino dans votre répertoire personnel, puis ouvrez le
- ☐ Compléter le programme
  → là où il y a ***// A COMPLETER***
- ☐ Lancer la simulation en insérant un obstacle devant l'émetteur récepteur ultrasons.
- ☐ Cliquer à droite pour ouvrir le moniteur de série pour voir les distances ainsi mesurées apparaitre.
- ☐ Changer la distance de l'obstacle.
- ☐ Vérifier que les mesures de distance changent dans le moniteur de série

4 **EXPERIMENTATION : régler les paramètres du radar de recul**

- ☐ Ajouter des éléments à votre circuit puis ajouter des lignes de code dans votre programme actuel afin de répondre au cahier des charges du TP
- ☐ Ouvrir le programme « radar_recul_avec_bip_adapte.ino »
- ☐ Compléter le programme et le téléverser
- ☐ Partir d'un obstacle assez loin et le rapprocher lentement !

Correction :




```
int trigPin = 4;  // Pins utilisés dans ce projet
int echoPin = 5;
long retard ; // définition des variables
int distance;

void setup() {
pinMode(trigPin, OUTPUT); // Règle le trigPin en sortie
pinMode(echoPin, INPUT); // Règle le echoPin entrée
pinMode(2, OUTPUT);
pinMode(3, OUTPUT);
Serial.begin(9600); // Starts the serial communication
}

void loop() {
digitalWrite(trigPin, LOW); //  Ces 5 lignes fabriquent une impulsion de 10ms
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

retard = pulseIn(echoPin, HIGH);    //  Mesure le retard à la réception, détection d'un pulse
distance = retard * 1e-6 * 34000 /2;  // calcule la distance correspondante

if(distance<100)
{
  digitalWrite(2,HIGH);
  tone(3,440,1000);
}
else
{
  digitalWrite(2,LOW);
  noTone(3);
}
Serial.print("Distance en cm : ");
Serial.println(distance);
delay(500);
}
```