

Noms : \_\_\_\_\_  
 Prénoms : \_\_\_\_\_  
 Classe : \_\_\_\_\_  
 Date : \_\_\_\_\_

**Note : /20**



## Objectifs du TP

L'objectif de ce TP est de programmer un radar de recul pour un véhicule.

La démarche suivie sera la suivante :

- on affichera sur la console la distance mesurée par le capteur
- on allumera des diodes en fonction de la proximité de l'obstacle
- on fera sonner un buzzer en cas d'objet très proche
- on modifiera le code pour qu'en cas d'utilisation d'une remorque, l'utilisateur puisse désactiver le radar de recul
- on modifiera les seuils d'alerte pour que le radar de recul puisse être utilisé avec un attelage à vélos.



## Barème

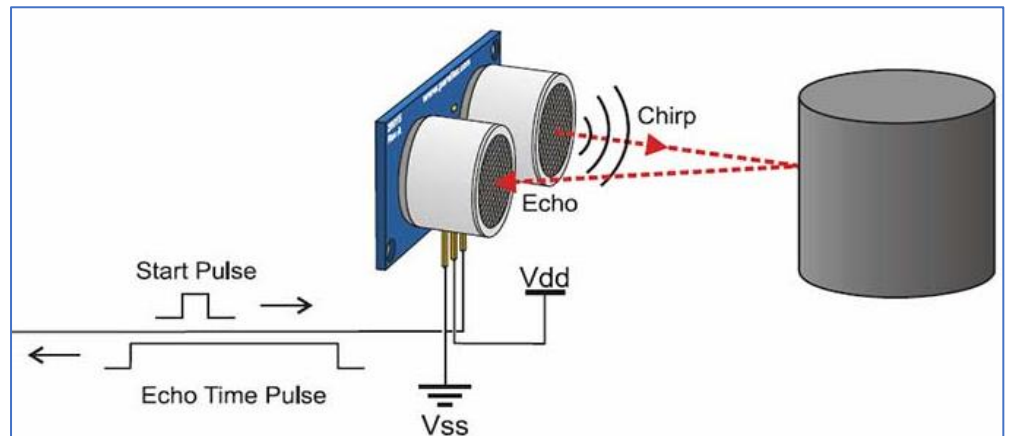
Attitude, quantité de travail, qualité du compte-rendu	/1
Fonctionnement du capteur à ultrasons	/5
Affichage sur la console de la distance	/4
Ajout de diodes et d'un buzzer	/6
Ajout d'un bouton poussoir pour désactiver le radar	/2
Modification des seuils d'alerte	/2

## 1. Capteur à ultrasons

- Q1:** Quelle est la vitesse du son dans l'air ?
- Q2:** Quelle est la plage de fréquences sonores audibles par l'oreille humaine ?
- Q3:** A quelle fréquence un capteur à ultrasons émet-il ses ondes sonores. Pouvons-nous entendre ces sons ?
- Q4:** Ecrivez la formule qui permet de calculer à quelle distance se trouve un objet connaissant la durée mise par l'écho pour revenir au récepteur
- Q5:** Application numérique avec les durées :  $\Delta T_1 = 180 \mu s$ ,  $\Delta T_2 = 1 ms$ ,  $\Delta T_3 = 20 ms$ .

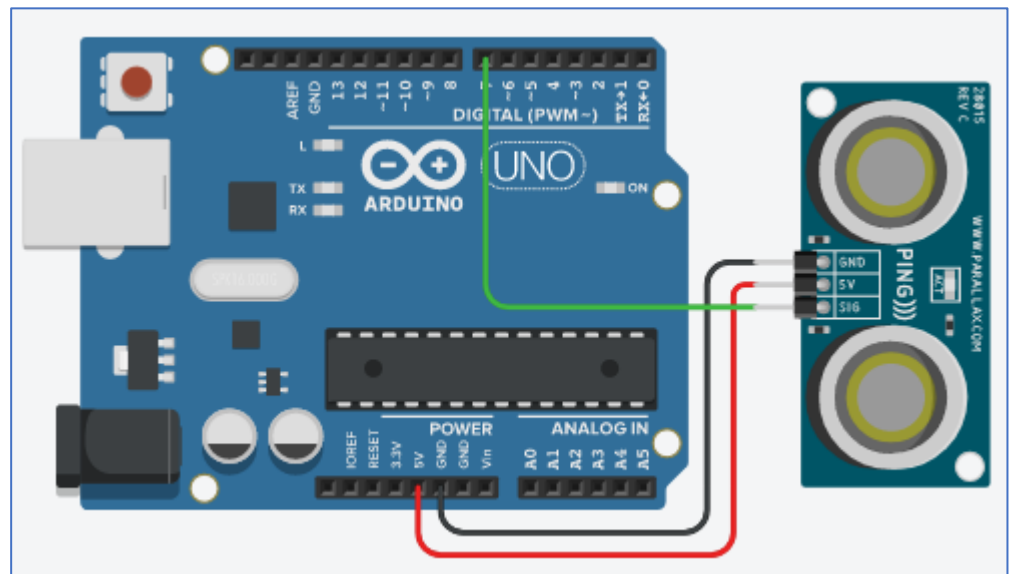
## 2. Affichage de la distance sur la console

Un capteur à ultrasons possède un émetteur d'ondes sonores et un récepteur d'ondes sonores



Connecter le capteur à la carte Arduino. La broche signal doit être branchée sur une broche numérique (digital), par exemple la broche 7.

Recopiez le programme ci-après, faites-le fonctionner puis répondez aux questions associées.



```

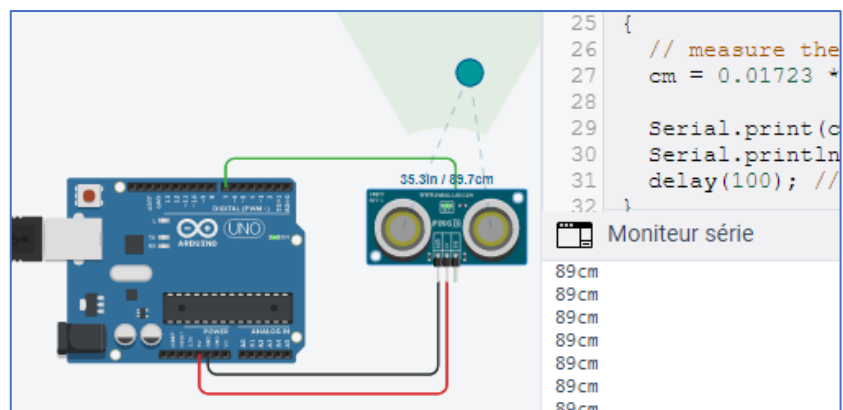
1
2 int cm = 0;
3
4 long readUltrasonicDistance(int triggerPin, int echoPin)
5 {
6     pinMode(triggerPin, OUTPUT); // Clear the trigger
7     digitalWrite(triggerPin, LOW);
8     delayMicroseconds(2);
9     // Sets the trigger pin to HIGH state for 10 microseconds
10    digitalWrite(triggerPin, HIGH);
11    delayMicroseconds(10);
12    digitalWrite(triggerPin, LOW);
13    pinMode(echoPin, INPUT);
14    // Reads the echo pin, and returns the sound wave travel time
15    // in microseconds
16    return pulseIn(echoPin, HIGH);
17 }
18
19 void setup()
20 {
21     Serial.begin(9600);
22 }
23
24 void loop()
25 {
26     // measure the ping time in cm
27     cm = 0.01723 * readUltrasonicDistance(7, 7);
28
29     Serial.print(cm);
30     Serial.println("cm");
31     delay(100); // Wait for 100 millisecond(s)
32 }
33

```

- Q6:** Trouver sur le web la signification de « int » et de « long »  
**Q7:** Expliquer la différence entre « Serial.print » et « Serial.println ».  
**Q8:** Expliquer la valeur « 0,01723 » utilisée ligne 27

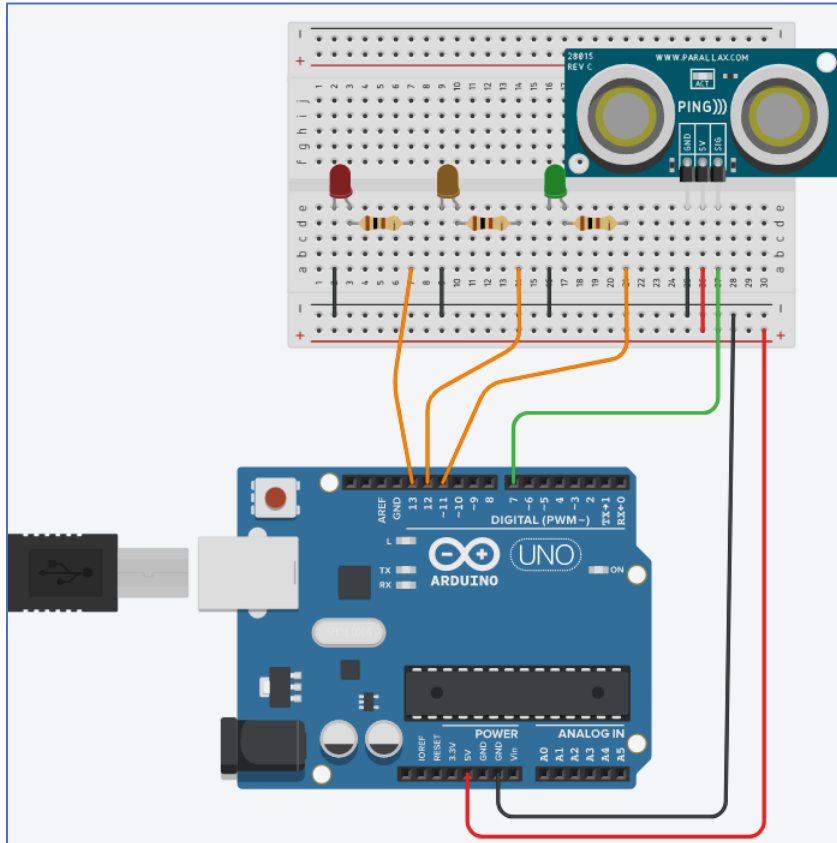
Lancer la simulation, cliquer sur le moniteur série, cliquer sur le capteur et déplacer le point bleu (ce point bleu représente l'obstacle)

- Q9:** Quelles sont les distances min et les distances max que ce modèle de capteur peut détecter ?



The screenshot shows an Arduino Uno board connected to an ultrasonic sensor. The sensor's display shows a range of 35.31m / 89.7cm. A blue dot represents the obstacle. The serial monitor on the right shows the output of the code, displaying '89cm' repeatedly.

## 3. Ajout de diodes et d'un buzzer



Réaliser le montage ci-contre en ajoutant 3 LED (1 rouge, 1 verte, 1 bleu) et 3 résistances de 220  $\Omega$ .

**Q10:** Indiquer pour chacun des blocs repérés de 1 à 4 à quoi il sert

```
int distance = 0;
```

```
long readUltrasonicDistance(int triggerPin, int echoPin)
```

```
{
  pinMode(triggerPin, OUTPUT);
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  return pulseIn(echoPin, HIGH);
}
```

N°1

```
void setup() {
  Serial.begin(9600);
```

```
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
```

N°2

```
}  
  
void loop() {  
  distance = 0.01723 * readUltrasonicDistance(7, 7);  
  Serial.println(distance);  
  // Allumer LED VERTE si distance > 2 m  
  if (distance >= 200) {  
    digitalWrite(11, HIGH);  
    digitalWrite(12, LOW);  
    digitalWrite(13, LOW);  
  } else {  
    // Allumer diode orange si d>80cm et d<200cm  
    if (distance < 200 && distance >= 80) {  
      digitalWrite(11, LOW);  
      digitalWrite(12, HIGH);  
      digitalWrite(13, LOW);  
    } else {  
      // Allumer diode rouge  
      digitalWrite(11, LOW);  
      digitalWrite(12, LOW);  
      digitalWrite(13, HIGH);  
    }  
  }  
  delay(10); // Delay a little bit to improve simulation performance  
}
```

**Q11:** Que devez-vous ajouter à ce programme pour pouvoir faire sonner un buzzer en cas d'objet situé à moins de 80cm ?

Modifiez le circuit électrique, faites-en une capture d'écran pour mettre dans votre compte-rendu puis faites une capture d'écran du code associé.

#### 4. Ajout d'un bouton poussoir

**Q12:** Modifiez le circuit électrique, puis le code et insérez les captures d'écran dans votre compte-rendu

#### 5. Modification des seuils de déclenchement

**Q13:** Quelles sont les lignes que vous devez changer pour que les 3 seuils soient 7cm, 20 cm et 50cm.

Réaliser les modifications et insérez les captures d'écran dans votre compte-rendu.