

Thème : numérique

Faire de la physique avec ARDUINO - 5



Ce cinquième chapitre propose des activités pédagogiques à faire avec les élèves. Ces TP sont issus de travaux faits par des collègues enseignants lors du stage de formation continue qui s'est déroulé à l'ENSFEA en Septembre 2019.

Ces TP portent sur :

- L'étude de la loi de Mariotte en enseignement de spécialité (première)
- L'étude d'une thermistance en 2^{nde} GT
- La mesure de distances par écholocation (1ere STL)

TP arduino sur la loi de Mariotte

Jérôme Thurillat et Bénédicte Quentin

Cours : Description d'un fluide au repos (*SPE*)

La loi de Mariotte indique qu'à température constante et à quantité de matière constante, le produit de la pression P d'un gaz par le volume V qu'il occupe est constant.

Est-ce qu'un paquet de chips peut exploser au cours d'une randonnée en haute montagne ?

On propose de tester cette loi et de répondre à cette question...



1- Utilisation d'un capteur de pression et du logiciel Arduino

A- Montage expérimental

Brancher un capteur de pression sur la broche analogique A2 et alimenter la carte Arduino à l'aide du câble USB relié à l'ordinateur.

Régler le volume de la seringue à 60 mL.

Fixer la seringue au tuyau relié au capteur de pression.



B- Test du programme Arduino

Ouvrir le programme Arduino et le tester en faisant varier le volume et en relevant les différentes pressions.

```
// Capteur de pression MPXHZ6400AC6T1
// Déclaration des variables sur EDUCADUINO LAB
#define _NUMERO_BROCHE_ANALOGIQUE A2

void setup()
{
  Serial.begin(9600);
}
void loop(){
  float tension=analogRead(_NUMERO_BROCHE_ANALOGIQUE);
  //Conversion de la tension numérique en tension analogique
  float tension_lue=tension*5.0/1023;
  //Calcul de la pression à partir de "tension_lue" pour le capteur
  float pression = tension_lue *760.0 + 200.0;

  Serial.print("P = ");
  Serial.print(pression, 1);
  Serial.println(" hPa");
  delay(1000);
}
```

C- Etude et amélioration du programme Arduino

- 1- Donner l'unité de la pression donnée par le capteur
- 2- Proposer les modifications à apporter pour obtenir :
 - la pression en kPa,
 - un délai de 5 s entre chaque mesure,
 - la valeur de la pression avec le même nombre de décimales que le volume

2- Test de la loi de Mariotte

A- Résultats expérimentaux

P (.....)					
V (.....)					
$\frac{1}{V}$ (.....)					

B- Exploitation des résultats

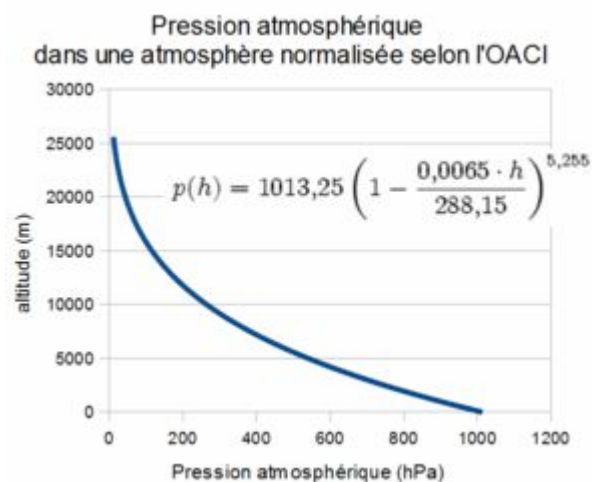
A partir du programme Python ci-dessous, permettant de tracer la caractéristique d'un conducteur ohmique de I en fonction de U, apporter les modifications permettant de tracer P en fonction de $V = \frac{1}{V}$

```
import matplotlib.pyplot as plt
# U et I sont à compléter.
U=[valeur1, valeur2, ....]
I=[valeur1, valeur2, ....]
#affichage graphique
plt.grid()
plt.title("Caractéristique tension-intensité d'une VDR")
plt.ylabel("U (en V)")
plt.xlabel("I (en A)")
plt.plot(I, U, 'o', label='Points issus de la mesure')
plt.legend()
plt.show()
nombre_U = len(U)
nombre_I = len(I)
#Calcul de la constante
xmoyen = sum(I)/nombre_I
ymoyen = sum(U)/nombre_U
constante = ymoyen/xmoyen
print("I=", constante, "xU")
```

Tester la loi de Mariotte à l'aide du programme précédent.

3-Et le paquet de chips....

Sachant que le volume d'explosion du paquet de chips est de 250 mL, calculer la pression qui le fera éclater



Quelques remarques et éléments de réponse:

Cours : Description d'un fluide au repos (*SPE*)

La loi de Mariotte indique qu'à température constante et à quantité de matière constante, le produit de la pression P d'un gaz par le volume V qu'il occupe est constant.

Est-ce qu'un paquet de chips peut exploser au cours d'une ra montagne ?

On propose de tester cette loi et de répondre à cette question...



1- Utilisation d'un capteur de pression et du logiciel Arduino

A- Montage expérimental

Brancher un capteur de pression sur la broche analogique A2 et alimenter la carte Arduino à l'aide du câble USB relié à l'ordinateur.

Régler le volume de la seringue à 60 mL.

Fixer la seringue au tuyau relié au capteur de pression.



Attention :

-manipulation à réaliser avec un capteur de pression JEULIN et non un capteur de pression ARDUINO qui est un simple baromètre.

-bien brancher le capteur de pression JEULIN sur une broche analogique du microcontrôleur (de type A0, A1, A2.....)

-Le programme ARDUINO proposé ci-dessous n'est pas adapté pour l'utilisation d'un écran LCD (comme sur la photo ci-contre).

B- Test du programme Arduino

Ouvrir le programme Arduino et le tester

```
// Capteur de pression MPXHZ6400AC6T1
// Déclaration des variables sur EDUCADUINO LAB
#define _NUMERO_BROCHE_ANALOGIQUE A2

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

void loop(){
  float tension=analogRead(_NUMERO_BROCHE_ANALOGIQUE);
  //Conversion de la tension numérique en tension analogique
  float tension_lue=tension*5.0/1023;
  //Calcul de la pression à partir de "tension_lue" pour le capteur
  float pression = tension_lue *760.0 + 200.0;

  Serial.print("P = ");
  Serial.print(pression, 1);
  Serial.println(" hPa");
  delay(1000);
}
```

C- Etude et amélioration du programme Arduino

- 1- Donner l'unité de la pression donnée par le capteur.
- 2- Proposer les modifications à apporter pour obtenir :
 - la pression en kPa,
 - un délai de 5 s entre chaque mesure,
 - la valeur de la pression avec le même nombre de décimales que le volume.

Modifications : cf ci-dessous

```
// Capteur de pression MPXHZ6400AC6T1
// Déclaration des variables sur EDUCADUINO LAB
#define _NUMERO_BROCHE_ANALOGIQUE A2

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

void loop(){
  float tension=analogRead(_NUMERO_BROCHE_ANALOGIQUE);
  //Conversion de la tension numérique en tension analogique
  float tension_lue=tension*5.0/1023;
  //Calcul de la pression à partir de "tension_lue" pour le capteur
  float pression = tension_lue *76.0 + 20.0;

  Serial.print("P = ");
  Serial.print(pression, 0);
  Serial.println(" kPa");
  delay(5000);
}
```

2- Test de la loi de Mariotte

La loi de Mariotte indique qu'à température constante et à quantité de matière constante, le produit de la pression P d'un gaz par le volume V qu'il occupe est constant.

A- Résultats expérimentaux

P (kPa)	68	71	74	79	83
V (mL)	60	55	50	44	40
$\frac{1}{V}$ (mL ⁻¹)	1,7E-02	1,8E-02	2,0E-02	2,3E-02	2,5E-02

B- Exploitation des résultats

A partir du programme Python ci-dessous, permettant de tracer la caractéristique d'un conducteur ohmique de I en fonction de U , apporter les modifications permettant de tracer P en fonction de $\text{inverse}V = \frac{1}{V}$

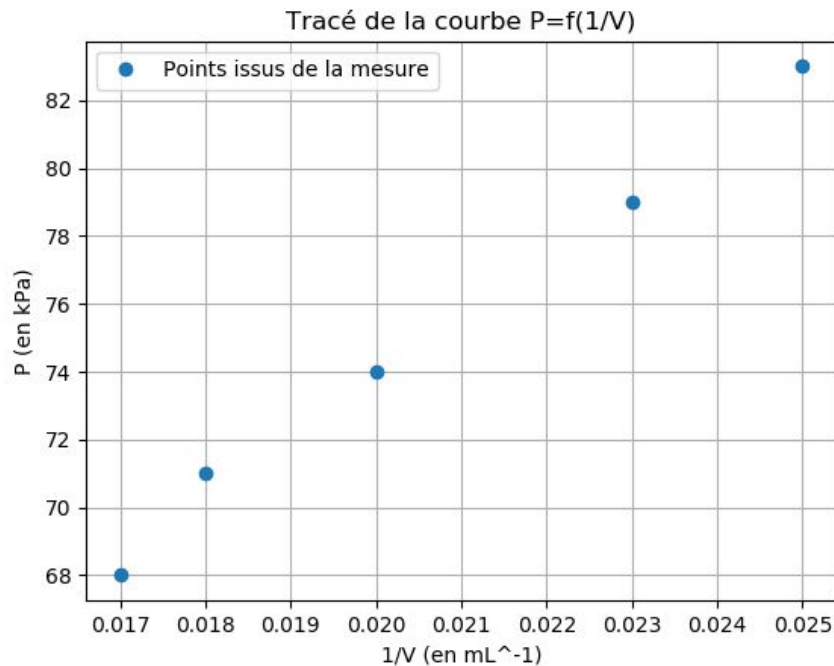
```
import matplotlib.pyplot as plt
# U et I sont à compléter.
U=[ valeur1, valeur2, ....]
I=[valeur1, valeur2, .... ]
#affichage graphique
plt.grid()
plt.title("Caractéristique tension-intensité d'une VDR")
plt.ylabel('U (en V)')
plt.xlabel('I (en A)')
plt.plot(I, U,'o',label='Points issus de la mesure')
plt.legend()
plt.show()
nombre_U = len(U)
nombre_I = len(I)
#Calcul de la constante
xmoyen = sum(I)/nombre_I
ymoyen = sum(U)/nombre_U
constante = ymoyen/xmoyen
print("I=",constante, "xU")

import matplotlib.pyplot as plt
# inverseV et P sont à compléter.
inverseV=[0.017,0.018,0.020,0.023,0.025]
P=[68,71,74,79,83]
#affichage graphique
plt.grid()
plt.title("Tracé de la courbe P=f(1/V)")
plt.ylabel('P (en kPa)')
plt.xlabel('1/V (en mL^-1)')
plt.plot(inverseV, P,'o',label='Points issus de la mesure')
plt.legend()
plt.show()
nombre_inverseV = len(inverseV)
nombre_P = len(P)
```

```
#Calcul de la constante
xmoyen = sum(inverseV)/nombre_inverseV
ymoyen = sum(P)/nombre_P
constante = ymoyen/xmoyen
print("P=",constante, "x 1/V")
```

Tester la loi de Mariotte à l'aide du programme précédent.

Résultats obtenus :



ATTENTION !

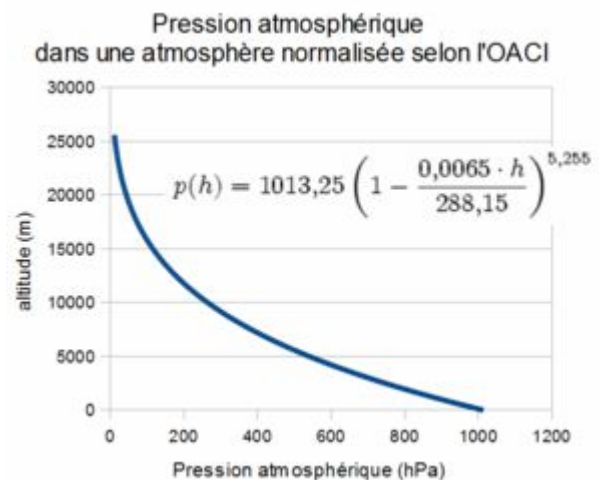
Il faut fermer la fenêtre du graphique pour obtenir l'équation ci-dessous dans EDUPYTHON.

$$P = 3640.776699029126 \times 1/V$$

3-Et le paquet de chips....

Sachant que le volume d'explosion du paquet de chips est de 250 mL, calculer la pression qui fera éclater le paquet de chips.

On obtient environ 15 kPa = 150 hPa.



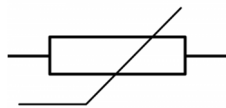
Etude d'un capteur de température

Richard Mariette

Documents

Document 1 : la thermistance

La thermistance est un élément qui permet de détecter la température. Elle se compose d'un matériau semi-conducteur qui permet de capter tous les changements de température, même si ces derniers restent relativement faibles, grâce à une variation de sa résistance électrique.



Document 2 : la carte arduino

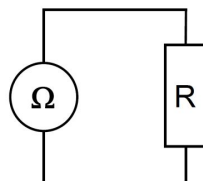
La carte Arduino Uno comporte un microcontrôleur, c'est-à-dire l'équivalent d'un petit microprocesseur. Cette carte peut recevoir des informations analogiques ou numériques sur ses entrées et renvoyer des informations numériques sur ses sorties.

Les entrées analogiques de la carte convertissent la valeur d'entrée (une tension entre 0 et 5V) en valeur numérique sur 10 bits, soient 1024 valeurs possibles allant de 0 à 1023.

Puisque l'on a 1024 valeurs possibles pour une plage de mesure de 5V, la résolution (plus petite variation de tension) de ce convertisseur est de : $\frac{5,0}{1024} = 0,0049 \text{ V}$

Document 3 : l'ohmmètre :

Un ohmmètre est un instrument qui permet de mesurer la résistance électrique d'un composant ou d'un circuit électrique. L'unité de mesure est l'ohm, noté Ω .



Document 4 : matériel disponible

Un bécher

Une plaque de montage

Un ohmmètre

Glaçons

Eau chaude

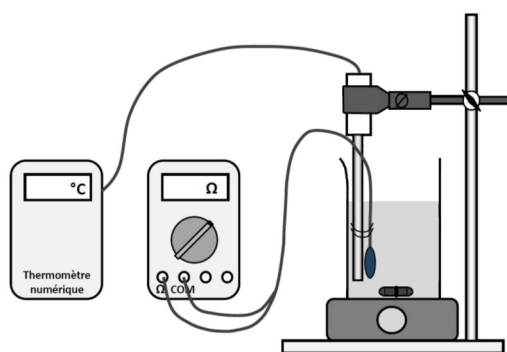
Une thermistance CTN 10 k Ω

Agitateur magnétique

TP 1 : étude de la thermistance

Vous allez devoir déterminer la courbe d'étalonnage reliant la résistance de la thermistance qui est à votre disposition en fonction de la température.

Dans un premier temps, proposer un montage, à l'aide du matériel mis à votre disposition, permettant de mesurer la valeur de la résistance de la thermistance pour des températures différentes (les températures seront comprises entre 1 et 80°C)



Effectuer le montage proposé puis réaliser une série de mesures (résistance ($k\Omega$), température ($^{\circ}C$)) puis tracer le graphique représentant **la résistance en fonction de la température**.

T ($^{\circ}C$)									
R ($k\Omega$)									

La résistance électrique de la thermistance est-elle fixe ou variable ? Que se produit-il lorsque la température augmente ?

TP : fabrication d'un thermomètre électronique

Vous avez pour objectif de concevoir un thermomètre électronique capable d'alerter quand la température de la pièce dépasse 25°C.

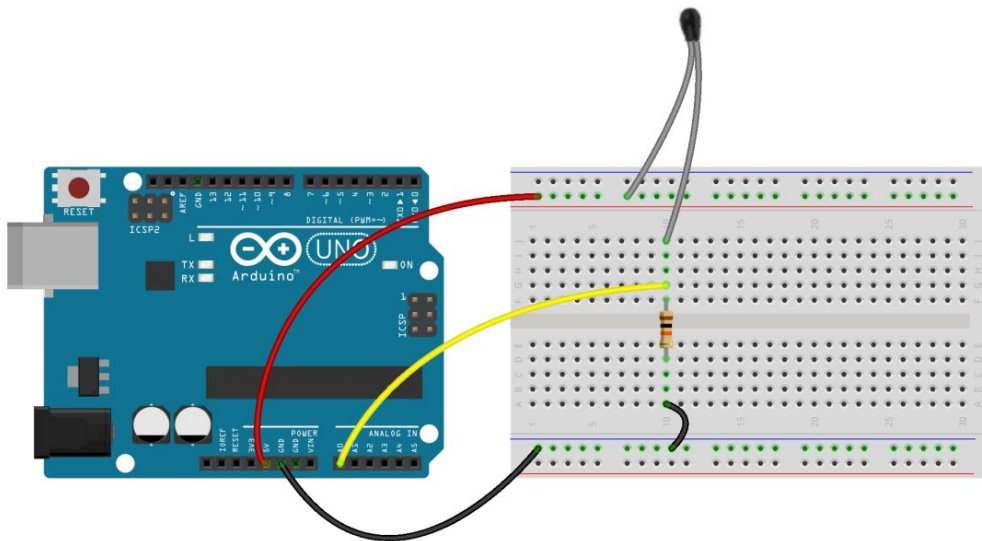
Pour cela, il s'agit de :

- Étalonner la thermistance
- Programmer le microcontrôleur pour le commander.

Etalonnage de la thermistance

1. Réaliser le montage et compléter le programme Arduino.

Schéma du montage



Programme Arduino

```
int ctnPin =0;
int valeur;
float tension;

void setup () {
  Serial.begin (9600);
  pinMode(ctnPin,INPUT);
}
void loop(){
  valeur= analogRead (ctnPin);
  tension = valeur*(5.0/1024);

  Serial.println (tension);
  delay (1000);
}
```

- Effectuer dans un premier temps un étalonnage, il vous faudra donc mesurer la valeur de la tension aux bornes de la résistance pour des températures allant de 10 à 40°C.

T (°C)						
U _r (V)						

- Tracer le graphique $T = f(U_r)$ et déterminer l'équation de la droite.
- En vous aidant de l'équation obtenue précédemment, compléter le programme afin de déterminer la température en fonction de U_r (dans le programme la variable « tension » correspond à U_r).

Commande du microcontrôleur

- Ecrire un programme qui permette d'afficher la valeur de la température et de commander le microcontrôleur afin d'allumer une diode si la température dépasse 25°C.

Correction

```
int ctnPin =A0;
int valeur;
int LED_ROUGE=4;
int LED_VERTE=5;
float tension;
float temperature;

void setup () {
  Serial.begin (9600);
  pinMode(ctnPin,INPUT);
  pinMode(LED_ROUGE,OUTPUT);
  pinMode(LED_VERTE,OUTPUT);
}
void loop(){
  valeur= analogRead (ctnPin);
  tension = valeur*(5.0/1024);
  temperature = tension*20.694-27.428;
  if (temperature >25) {
    digitalWrite (LED_ROUGE,0);
    digitalWrite (LED_VERTE,1);
  }
  else
  {digitalWrite (LED_ROUGE,1);
  digitalWrite (LED_VERTE,0);}

  Serial.println (temperature);
  delay (1000);
}
```

Mesure de distance par écho ultrason

Jean-François Barbier

Ressource :

<https://www.memorandum.ovh/tuto-arduino-utiliser-un-module-ultrason-hc-sr04/>

Capacité expérimentale exigible en 1re STL :

- Déterminer expérimentalement des distances à partir de la mesure d'un temps de vol d'une onde sonore ou ultrasonore.

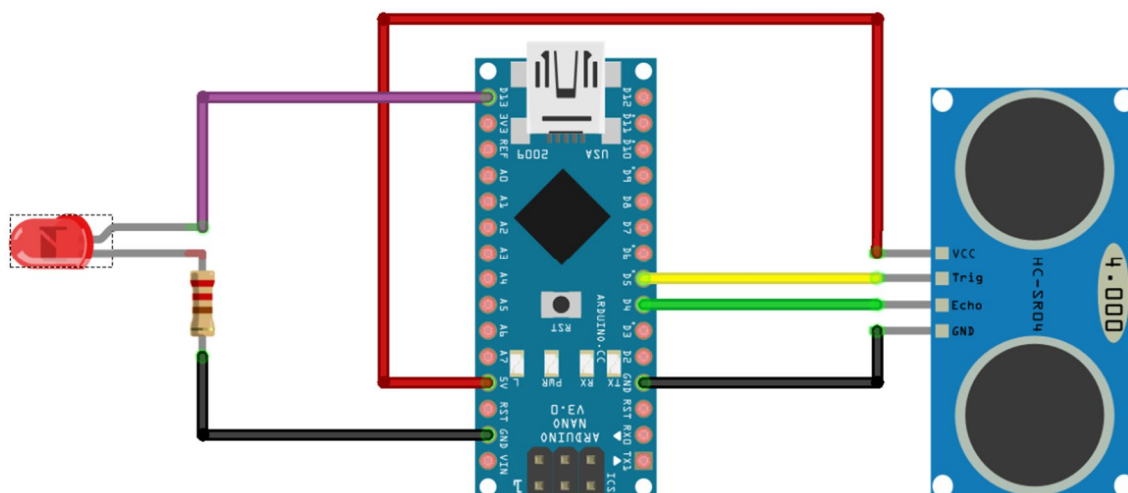
1) Matériel

- Carte Arduino
- Module de détection ultrason HC-SR04
- LED
- Résistance
- Connectique
- Plaque de montage
- Ordinateur équipé du programme Arduino

2) Montage

- Câblage module HC-SR04 :
 - 5V sur VCC
 - GND sur gnd
 - D5 sur TRIG
 - D4 sur ECHO
- Câblage de la LED :
 - Sur broche D13 : Résistance en série avec la LED
 - La longue branche de la LED du côté D13. La branche courte de la LED du côté GND ; c'est la masse du circuit.

Schéma :



3) Mise en œuvre

3.1 - S'approprier

1. Réaliser le montage décrit dans le lien ci-dessus.
2. Ouvrir avec le programme Arduino le fichier Distance_ultrason, le téléverser sur la carte Arduino.
3. Identifier dans le code Arduino la ligne de code permettant le calcul de la distance.
4. Modifier le programme pour que la LED s'allume à une distance inférieure à 50 cm.
5. Schématiser l'écho de l'ultrason. Expliquer comment la distance est mesurée.
6. Calculer la distance correspondant à un temps de parcours de l'ultrason de 5 ms

3.2 - Réaliser

1. A l'aide d'un ruban gradué, placer un obstacle à une distance d choisie entre 0,1 m et 2m
2. Mesurer à l'aide du banc ultrason la distance entre le dispositif et l'obstacle. Répéter dix fois la mesure.
3. Pour la série de valeurs de mesures :
 - a. Représenter un histogramme,
 - b. Calculer :
 - la valeur moyenne,
 - l'incertitude de type A
 - c. Écrire la valeur mesurée, avec son incertitude

3.3 - Valider

Comparer la distance mesurée à l'aide du dispositif ultrason à celle mesurée avec un ruban gradué.

3.4 - Raisonner - communiquer

Proposer une démarche permettant d'évaluer les indications extrêmes de ce dispositif de mesure.

Programme Arduino à téléverser :

```
int trigPin = 5; //Trig sur broche 5
int echoPin = 4; //Echo sur broche 4
int avertisseur = 13; //Led sur broche 13
float mindist = 1.2; //On indique la distance en dessous de laquelle nous souhaitons voir la led
s'allumer
float duree;
float distance;

void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT); //On défini Trig comme une sortie
  pinMode(echoPin, INPUT); //On défini Echo comme une entrée
  pinMode(avertisseur, OUTPUT); //On défini la led comme une sortie
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10); //Trig envois pendant 10ms
  digitalWrite(trigPin, LOW);

  // On calcul le temps pour l'aller retour du signal
  duree = pulseIn(echoPin, HIGH); // la fonction pulseIn retourne une durée en microsecondes
  distance = (duree/1000000)*340/2;

  if (distance < mindist) { // On allume la led si on est moins loin que "mindist", mindist étant
    défini en début de programme
    digitalWrite(avertisseur,HIGH);
  }
  else { //sinon on éteind la led
    digitalWrite(avertisseur,LOW);
  }
  //Pour le moniteur série
  Serial.print(distance);
  Serial.print(" mètres ");
  Serial.print(duree);
  Serial.println(" microsecondes");
  delay(2000);
}
```

Grille d'évaluation par compétences

Question	Compétence	Capacité	Attendu	Barème
3.1	1 Réaliser	Réaliser un montage sur carte Arduino	choix des broches sur carte Validité du circuit sur la plaque de montage	1
	2 Réaliser	Copier et téléverser un programme sur un carte	Fichier copié localement Connexion correcte du module Arduino	1
	3 S'approprier	Reconnaître un calcul dans un programme	$\text{distance} = (\text{durée} / 1000000) * 340 / 2$ $m = s * (m/s)$	1
	4 S'approprier	Modifier une valeur	$\text{mindist} = 0,5$	1
	5 S'approprier	Schématiser une expérience	schéma de l'aller retour, distance 2d indiquée	1
		Expliquer le principe d'une mesure	$\text{distance} = 2 * \text{longueur parcours}$ Réflexion de l'ultrason sur l'obstacle	1
	6 Réaliser	Calculer une distance, à l'aide du modèle identifié	$d = 0,85 \text{ m}$	2
3.2	1 Réaliser	Mettre en œuvre un protocole expérimental	0 du ruban au niveau du module émetteur-récepteur distance choisie respectée Obstacle plan perpendiculaire à la direction de propagation de l'ultrason	1
	Réaliser	Mettre en œuvre un protocole expérimental	valeurs mesurées notées	1
	3 Valider	représenter l'histogramme associé à une série de mesures.	Echelle et axes légendés ; 10 batons	1
	Valider	Procéder à une évaluation de type A d'une incertitude type	$u = \sigma / \text{racine}(N)$	2
	Valider	Exprimer un résultat de mesure avec le nombre de chiffres significatifs adaptés et l'incertitude-type associée.	valeur moyenne $\pm u$	1
3.3	Valider	Discuter de la validité d'un résultat en comparant la différence entre le résultat d'une mesure et la valeur de référence d'une part et l'incertitude-type d'autre part.	valeur de référence dans l'intervalle de mesure	1
3.4	Raisonner ; Communiquer	Présenter une démarche de manière argumentée, synthétique et cohérente ; utiliser un vocabulaire adapté et choisir des modes de représentation appropriés	Démarche cohérente et intelligible	2
	Communiquer	Rédiger un document structuré	Titre numérotation des questions respectée	1
				18