FICHE 1
Fiche à destination des enseignants

Accident barotraumatique

<table>
<tr><td>Type d'activité</td><td colspan="2">Démarche expérimentale</td></tr>
<tr><td rowspan="2">Programme</td><td>Notions et contenus du programme de 1ère</td><td>Compétences exigibles du programme de 1ère</td></tr>
<tr><td>Modèle de comportement d'un gaz : loi de Mariotte.</td><td>Utiliser la loi de Mariotte.
Tester la loi de Mariotte, par exemple en utilisant un dispositif comportant un microcontrôleur.</td></tr>
<tr><td>Commentaires sur l'activité proposée</td><td colspan="2">Cette activité permet d'exploiter dans un programme Python des données obtenues avec un microcontrôleur.
Le programme Arduino est fourni, quelques modifications doivent être faites par les élèves et les commentaires doivent être ajoutés.
La première partie du programme Python est fournie, elle permet de récupérer les données obtenues par Arduino. Les élèves doivent ajouter une partie permettant de trouver le meilleur polynôme décrivant les points expérimentaux obtenus. Ils doivent ensuite ajouter une dernière partie pour tracer la courbe.
La difficulté de cet exercice est de coupler Arduino et Python, et demande donc que les élèves aient déjà travailler plusieurs fois à la fois la programmation Arduino et la programmation Python. Le montage à réaliser est par contre très simple.</td></tr>
<tr><td>Durée</td><td colspan="2">2h</td></tr>
<tr><td>Pré requis (classe 2nde)</td><td colspan="2">▪ Réalisation de montage et programmation avec un microcontrôleur Arduino.
▪ Programmation avec Python</td></tr>
</table>

Fiche 2
Fiche à destination des élèves

Risque barotraumatique en plongée

Le barotraumatisme est une lésion tissulaire provoquée par une variation de pression qui comprime ou dilate les gaz contenus dans différentes parties de l'organisme.

- Les poumons, le tube digestif, la partie du visage couverte par un masque, les yeux, les oreilles ou les sinus peuvent être affectés.
- Les symptômes varient et peuvent comprendre des affections respiratoires ou des douleurs thoraciques (barotraumatisme pulmonaire), des yeux injectés de sang (barotraumatisme du masque), des vertiges ou des douleurs dans l'oreille (barotraumatisme de l'oreille) et des douleurs faciales ou des saignements de nez (barotraumatisme des sinus).
- Le risque de barotraumatisme est maximal entre la surface et 10 mètres.
- Certaines mesures peuvent être prises pour éviter le barotraumatisme, notamment l'ascension lente avec respiration pendant la remontée (barotraumatisme pulmonaire), l'expulsion d'air par le nez à l'intérieur du masque facial (barotraumatisme du masque) et des bâillements ou la déglutition en pinçant les narines avec prise d'un décongestionnant nasal (barotraumatisme des sinus et de l'oreille).

L'élévation de la pression à l'extérieur de l'organisme est transmise uniformément au sang et aux tissus, qui ne sont pas comprimés car ils sont principalement composés de liquides. Les gaz (comme l'air à l'intérieur des poumons, des sinus, de l'oreille moyenne ou à l'intérieur du masque de plongée ou des lunettes) se compriment ou se dilatent lorsque la pression externe augmente ou diminue. Ces effets de compression et de dilatation peuvent donc provoquer des douleurs et des lésions tissulaires.
Le barotraumatisme affecte souvent les oreilles. Mais c'est le barotraumatisme affectant les poumons (barotraumatisme pulmonaire) qui est le plus grave.

D'après le site : https://www.msdmanuals.com/

Comment la pression d'un gaz varie-telle en fonction du volume à température constante ?

Nous allons utiliser un capteur de pression relié à un microcontrôleur Arduino pour déterminer comment varie la pression d'un gaz en fonction du volume à température constante.
Réaliser le montage et utiliser les documents pour déterminer la relation reliant la pression et le volume d'un gaz à température constante. Commenter vos résultats.

Document 1 : Loi de Mariotte

La loi de Boyle-Mariotte relie la pression et le volume d'un gaz à température constante. Elle a été découverte indépendamment et à quelques années d'intervalle par l'Irlandais Robert Boyle (1662) et le Français Edme Mariotte (1676).

Cette loi stipule qu'à température constante, si l'on modifie le volume d'un gaz (d'une valeur V_1 à une valeur V_2), dans le même temps, sa pression varie d'une valeur P_1 à une valeur P_2 tel que :

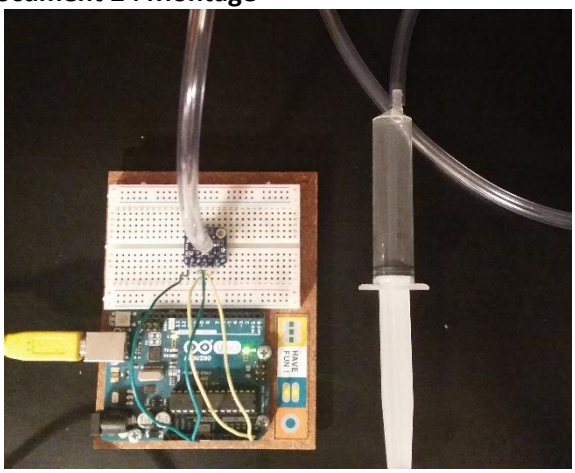
$$P_1V_1 = P_2V_2$$

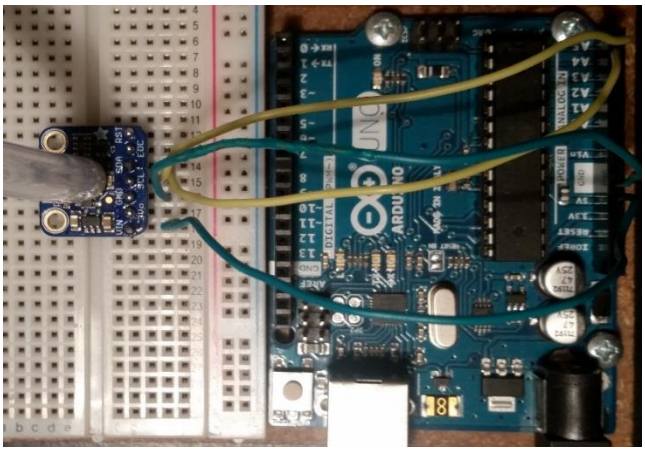
soit $P \times V = \text{constante}$

L'unité de pression est le Pascal (Pa)
L'unité de volume est le mètre cube m^3.

Pour vérifier expérimentalement cette relation, on pourra mesurer la valeur de la pression P d'un gaz à température constante pour différentes valeurs de volume V et tracer une courbe judicieusement choisie pour vérifier la loi de Mariotte.

Document 2 : Montage





Le capteur de pression MPRLS est relié par l'intermédiaire d'un tuyau souple à une seringue graduée en mL.

- → Connecter la borne Vin du capteur à la borne 5V du microcontrôleur.
- → Connecter la borne GND à la borne GND du microcontrôleur.
- → Connecter la borne SCL à la borne A5 du microcontrôleur.
- → Connecter la borne SDA à la borne A4 du microcontrôleur.

L'incertitude-type relative sur la mesure de la pression, indiquée par le constructeur, est de 1,5 % pour le capteur MPRLS 0025PSI.

Document 3 : Programme Arduino permettant de mesurer la pression à l'aide d'un capteur de pression Adafruit MPRLS

Ce programme se trouve dans la librairie propre au capteur MPRLS, qui a été préalablement téléchargé. Pour l'ouvrir : « fichier » → « exemples » → « adafruit MPRLS library » → « mprls_simpletest ».

⚠ Attention :

- quelques modifications doivent être faites dans la partie *void loop, voir ci-dessous*. En particulier la mesure de pression en PSI, unité anglo-saxonne(livre-force par pouce carré), doit être supprimée.
- Vous ajouterez des commentaires pour expliquer les instructions de la partie *void loop.*

```
#include <Wire.h>
#include "Adafruit_MPRLS.h"

// You dont *need* a reset and EOC pin for most uses, so we set to -1 and don't connect
#define RESET_PIN  -1  // set to any GPIO pin # to hard-reset on begin()
#define EOC_PIN    -1  // set to any GPIO pin to read end-of-conversion by pin
Adafruit_MPRLS mpr = Adafruit_MPRLS(RESET_PIN, EOC_PIN);

void setup() {
  Serial.begin(9600);  // 9600 indique le nombre de caractères que l'Arduino peut envoyer à l'ordinateur en 1 s
  Serial.println("MPRLS Simple Test"); // écrire le nom du programme
  if (! mpr.begin()) {   // si le capteur ne mesure aucune pression
    Serial.println("Failed to communicate with MPRLS sensor, check wiring?");  // alors écrire ce message d'erreur
  }
  Serial.println("Found MPRLS sensor");  // sinon indiquer que le capteur est bien détecté
}

void loop() {
  float pressure_hPa = mpr.readPressure();
  Serial.print(pressure_hPa);
  Serial.println(" hPa");
  delay(2000);
}
```

Document 4 : Programme python permettant d'exporter les résultats obtenus à l'aide du capteur de pression relié au microcontrôleur

```
from scipy import *   # importer la bibliothèque scipy
from pylab import *    # importer la bibliothèque pylab
import serial         # importer la bibliothèque serial qui permet de communiquer avec le port USB de l'ordinateur
serial_port= serial.Serial(port="COM3",baudrate = 9600)   # indique le port USB utilisé (à modifier
                                                            éventuellement) et la vitesse de communication

volume=np.linspace(11.0,2.0,10)    # définition du volume prenant 10 valeurs comprises entre 2.0 et 11,0 mL
print(volume)  # écrire les valeurs du volume

Pression = []  # liste correspondant à la pression, vide au début.
invPression=[ ]   # liste correspondant à l'inverse de la pression, vide au début.
for i in range(10):
   val=serial_port.readline().split()  # la grandeur notée val récupère via le port série une information du type « P
                                         (hPa)= 1113.26 » et split la découpe en trois blocs au niveau des blancs
   P=float(val[2])  # la grandeur notée P ne conserve que le bloc qui correspond à un nombre décimal
   invP=1/P   # introduction d'une nouvelle grandeur invP
   Pression.append(P) # chaque valeur de P est ajoutée dans la liste Pression
   invPression.append(invP)  # chaque valeur de invP est ajoutée dans la liste invPression

serial_port.close()      # le relevé des valeurs s'arrête au bout de 10 valeurs
print(invPression)       # écrire les valeurs de la liste invPression
print(Pression)       # écrire les valeurs de la liste Pression
```

Document 5 : Fonction python permettant de trouver le meilleur polynôme de degré n décrivant les points expérimentaux

- Pour déterminer le meilleur polynôme décrivant les points expérimentaux :

```
modele=np.polyfit( __ , __ , __ )     # polyfit est une fonction qui s'applique de la manière suivante :
                                        np.polyfit(x,y,degré du polynôme)
print(modele)          # la fonction renvoie les valeurs des coefficients du polynôme
```

- Pour tracer la courbe :

```
    a,b = modele  # on définit a et b comme étant les coefficients obtenus par modélisation pour un polynôme
                    de degré 1(ajouter des coefficients si polynôme de degré 2, 3…)
    yMod= _____ # introduire une grandeur yMod du type a*x +b si polynôme de degré 1
    plt.plot(x,yMod,'r-', label='modèle') # tracer la courbe yMod = f(x)
```

Document 6 : Pour afficher une courbe en python

Il faut tout d'abord importer le module *pyplot* : *from matplotlib.pyplot as plt*
Instructions pour tracer un graphe y = f(x) :

- pour tracer un graphe avec des valeurs de x et des valeurs de y en reliant les points dans l'ordre de la liste: *plt.plot(x, y)*
- pour afficher une grille de fond : *plt.grid()* ou pour un seul axe *plt.grid(axis='y')*
- on peut indiquer le symbole et la couleur des points du graphe et indiquer une légende : *plt.plot(x, y, 'bo',label= "legende") :* bleu et avec des ronds. Si on veut relier les points 'b-'.
- Pour légender le graphe :
  plt.xlabel("légende")
  plt.legend()
- Pour mettre un titre au graphe : *plt.title (" titre ")*
- *plt.show()* : pour afficher le graphe.