Niveau : première enseignement de spécialité	Mouvement et interactions partie 2. Description d'un fluide au repos
Loi fondamentale de la statique des fluides.	Dans le cas d'un fluide incompressible au repos, utiliser la relation fournie exprimant la loi fondamentale de la statique des fluides : $P_2-P_1 = \rho g(z_1-z_2)$. *Tester la loi fondamentale de la statique des fluides.*

Dossier TP Plug'Uino® :
Loi de la statique des fluides
Programmation en langage Python

1. Résumé de l'activité

Cette activité expérimentale a pour but de tester le modèle de la loi fondamentale de la statique des fluides, à partir de mesures de pression obtenues à l'aide d'un capteur de pression différentiel relié à un microcontrôleur.
Programmation et exploitation des résultats avec le langage Microcontrôleur type Arduino et un tableur

2. Thème du programme abordé

Niveau : première enseignement de spécialité	Mouvement et interactions partie 2. Description d'un fluide au repos
Loi fondamentale de la statique des fluides.	Dans le cas d'un fluide incompressible au repos, utiliser la relation fournie exprimant la loi fondamentale de la statique des fluides : $P_2-P_1 = \rho g(z_1-z_2)$. *Tester la loi fondamentale de la statique des fluides.*

3. Matériel mis en œuvre

- Une interface Plug'Uino® Réf. 650 003
- Un capteur de pression différentiel (0 - 50 hPa) Réf. 651 059
- Une éprouvette graduée en plastique TPX 100 mL Réf. 521 018

4. Présentation de l'expérience réalisée

À l'aide d'un microcontrôleur, on mesure la différence de pression entre la surface et différentes profondeurs.
Les mesures successives sont traitées à l'aide d'un tableur-grapheur ou avec un script Python, l'objectif étant de tester la loi de la statique des fluides :

$\Delta P=P_2-P_1 = \rho g(z_1-z_2)$

Le microcontrôleur est particulièrement adapté pour réaliser de nombreuses mesures en un temps limité, ce qui permet d'obtenir une mesure qui moyenne les légères fluctuations des valeurs fournies par le capteur
Branchement du capteur de pression sur la broche analogique : A1
Branchement d'une LED sur la broche digitale D2 (facultatif)

5. Programmation

Pour la mise en place de la connexion du microcontrôleur et la configuration initiale Python, se référer à la fiche : *Python et microcontrôleur.pdf*

Ci-dessous, un script Python fonctionnel, permettant d'effectuer une mesure sur l'entrée analogique A1. Ce script est disponible en téléchargement : *exemple_pyfirmata.py*

```
""" Sciencéthic script exemple de lecture sur une broche analogique d'un microcontrôleur type Arduino
capteur branché sur la broche analogique A1, LED branchée sur la broche digitale D2"""

from pyfirmata import Arduino, util
import time                          #importation des bibliothèques

#------------------------------ paramétrage de la communication-----------------------------------
com=input('Saisir le port de communication COM1, COM2, COM3 .... :')
port = Arduino(com)                  # la variable port contient le port de communication
util.Iterator(port).start()          # itérateur permettant de ne pas saturer la liaison série
A1 = port.get_pin('a:1:o')           # A1 désigne la broche analogique 1 (a:1) utilisée en sortie (:o)
time.sleep(0.5)                       # délai d'attente pour la mise en place de la communication
port.digital[2].write(1)             # écriture sur le port digital 2 (0k pour mesure si LED allumée)

#------------------------------ mesures ----------------------------------------------------------
mesures_capteur=[]                    # liste des mesures
poursuite=True                        # initialisation de la variable poursuite des mesures
while poursuite:                      # tant qu'une mesure est à effectuer
    m=A1.read()                       # lecture de la valeur sur la broche A1
    mesures_capteur.append(m)                 # ajout de la valeur dans la liste
    print('mesure : ',m)              # affichage de la valeur mesurée
    poursuite=input('Déclencher une nouvelle mesure o/n ? ')
    if poursuite=='n'or poursuite=='N':
        poursuite=False
print('liste des mesures : ',mesures_capteur) # affichage de la liste des mesures
port.digital[2].write(0)             # écriture sur le port digital 2 (fin de mesure si LED éteinte)
port.exit()                          # fermeture de la communication sur le port
```

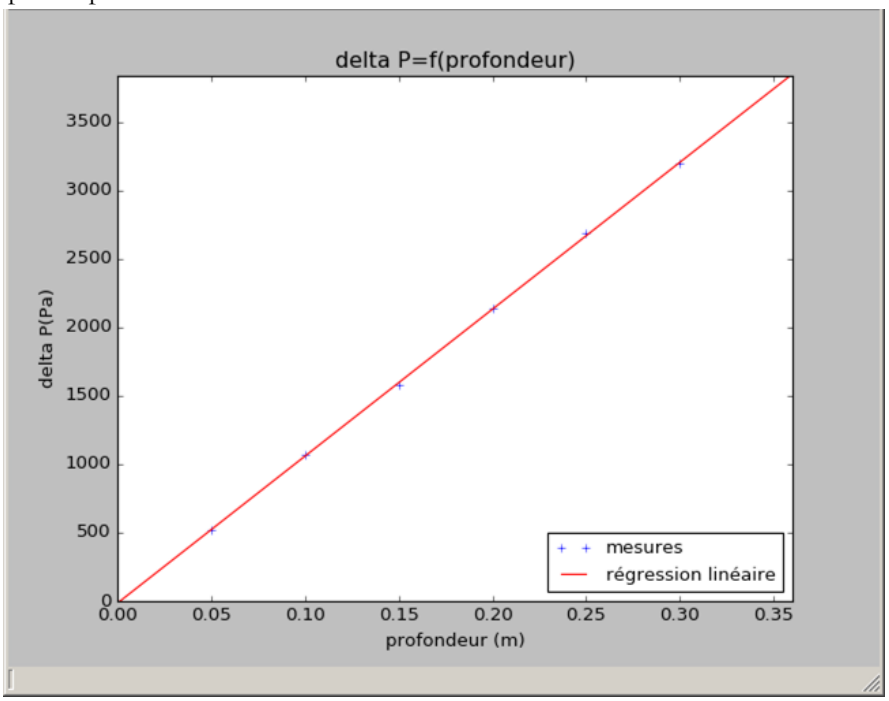
Un script plus complet permettant de saisir la profondeur en parallèle avec les mesures et d'effectuer éventuellement un tracé.
Ce script est disponible en téléchargement : *loi*_hydrostatique_*complet.py*

```
""" Sciencéthic mesures pour loi de l'hydrostatique
lecture sur une broche analogique d'un microcontrôleur type Arduino
capteur branché sur la broche analogique A1, LED branchée sur la broche digitale D2"""
# ------------------------------importation des bibliothèques-----------------------------------------
from pyfirmata import Arduino, util
import time

#------------------------------ paramétrage de la communication-------------------------------------
com=input('Saisir le port de communication COM1, COM2, COM3 .... :')
port = Arduino(com)          # la variable port contient le port de communication
util.Iterator(port).start()    # itérateur permettant de ne pas saturer la liaison série
A1 = port.get_pin('a:1:o')     # A1 désigne la broche analogique 1 (a:1) utilisée en sortie (:o)
time.sleep(0.5)                # délai d'attente pour la mise en place de la communication
port.digital[2].write(1)       # écriture sur le port digital 2 (Ok pour mesure si LED allumée)

#------------------------------ mesures ------------------------------------------------------------
mesure=[]                       # liste des profondeurs
profondeur=[]                   # liste des profondeurs
poursuite=True                  # initialisation de la variable poursuite des mesures
while poursuite:                # tant qu'une mesure est à effectuer
    prof=float(input('profondeur en mètre :'))   # saisie de la profondeur
    profondeur.append(prof)             # enregistrement de la profondeur dans la liste
    a=input('taper ENTER pour effectuer la mesure') # déclenchement de la mesure
    m=A1.read()                         # lecture de la mesure sur la broche
    mesure.append(m)                    # ajout de la mesure dans la liste
    print('mesure : ',m)                # affichage de la mesure
    poursuite=input('nouvelle mesure o/n : ')
    if poursuite!='o' and poursuite!='O':
        poursuite=False

port.digital[2].write(0)  # écriture sur le port digital 2 (test de démarrage si LED sur port digital 2)
port.exit()                         # fermeture de la communication sur le port
print()
print('profondeurs : ',profondeur)   # affichage de la série des profondeurs
print('mesures : ',mesure)           # affichage de la série de mesures
```

6. Résultats obtenus et traitement :

profondeurs : [0.0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3]
mesures : [0.0978, 0.1711, 0.2463, 0.3157, 0.392, 0.4673, 0.5376]

Tracé obtenu après exploitation des mesures :



7. Exploitation

7.1. Programmation

En fonction des capacités des élèves, on peut proposer plusieurs déclinaisons :

- à un niveau **d'initiation**, enlever certains commentaires du programme et demander aux élèves à quoi correspond cette ligne ;
- à un niveau **plus avancé**, effacer certaines lignes et demander aux élèves de programmer eux-mêmes ces lignes (par exemple, la ligne où la mesure est effectuée) ;
- à un niveau **expert**, demander aux élèves de programmer le microcontrôleur : dans un premier temps, pour renvoyer une mesure unique ; dans un second temps, pour acquérir une série de mesures pour le calcul d'une moyenne.

7.2. Résultats des mesures

On peut compléter l'analyse sur les points suivants :

- régression linéaire sur le tracé P=f(z) et détermination de la masse volumique du fluide
- détermination de la relation modélisant la réponse du capteur (étalonnage) en utilisant un pressiomètre.

8. Pistes d'exploration possibles

- Modifier le fluide (huile, éthanol …)

9. Points de vigilance

- Évaluer le volume de gaz contenu dans le tube de raccordement du capteur à la seringue pour l'intégrer dans le volume total de gaz.
- Dans le cas du tracé d'un graphique, le fermer pour relancer l'exécution du script.