

# Projet FORMATION DIU

## Objet : station météo

### Sujet du projet

#### **Création d'une mini station météo**

- Les concepts techniques et aspects transverses

A partir d'une mini station de météo et de son capteur physique BMP280, en s'aidant d'une arduino, le projet consiste à générer un fichier csv. Ce fichier devra être traité en Python.

A partir de ce traitement, le compte rendu des résultats se fera avec une page HTML comportant du CSS.

- Les prérequis

Pour mener ce TP, les élèves auront déjà :

- fait un TP avec de l'arduino ;
- utiliser un capteur .

Ils auront reçu un enseignement Python assez complet. Ils devront avoir une maîtrise : - des boucles ;

- de la notion de tableau ;
- des fichiers en lecture, en écriture ;
- de l'utilisation de fonctions ;
- d'appel aux bibliothèques.

L'enseignement des généralités d'HTML et de CSS devra avoir démarré pour les lancer sur la construction de leur page de rendu.

- Nombre d'élèves

Il est prévu que les élèves soient par groupe de 4.

- Volume horaire (heures classe + heures maison)

Il est prévu 6 séances de 2 heures en classe avec du travail à terminer à la maison en fin de chaque séance.

Au sein du groupe, l'organisation se fera selon trois modalités :

- un élève travaillera sur le capteur, l'Arduino avec en finalité la création d'un fichier brut csv.
- deux élèves s'occuperont, en Python, de la lecture du fichier csv, devront créer un tableau, traiter les données et générer des courbes représentatives des informations recueillies. On utilisera le format PNG.
- Un élève travaillera sur l'élaboration de la page en HTML ainsi que sur la forme à l'aide du CSS.

Le premier travail que devront effectuer les groupes sera de définir les formats d'échanges entre chaque découpage du projet. Les spécifications devront être suffisamment précises pour éviter les erreurs. Elles concerneront, par exemple, les données du capteur, le fichier csv, les données transmises en Python, le format HTML etc...

- Modalités de rendu et d'évaluation

Le groupe devra soutenir son projet lors d'une présentation orale de dix minutes en s'appuyant sur un outil numérique. Cette présentation sera suivie d'une démonstration fonctionnelle du projet. Associé à cette présentation, un compte rendu écrit de suivi, de projet au format pdf ( 4 à 6 pages), retraçant les étapes du projet et mettant en évidence les choix, les difficultés rencontrées, devra être transmis.

- Déroulement d'une séance de projet en classe

Au début de chaque séance, les élèves devront évaluer la réussite de leurs objectifs. Ils devront travailler en autonomie, rechercher des informations, prendre des décisions et avancer dans leurs projets...

Ils devront compléter un cahier de suivi et définir les objectifs à atteindre pour la prochaine séance ; bien évidemment en fonction des difficultés rencontrées et de l'avancement du projet. Ils devront terminer la séance en ayant défini le travail maison et sa répartition.

Au cours de la séance, ils devront rendre compte de l'avancée du projet, des difficultés rencontrées avec le professeur qui servira de chef de projet.

- Eléments de corrections

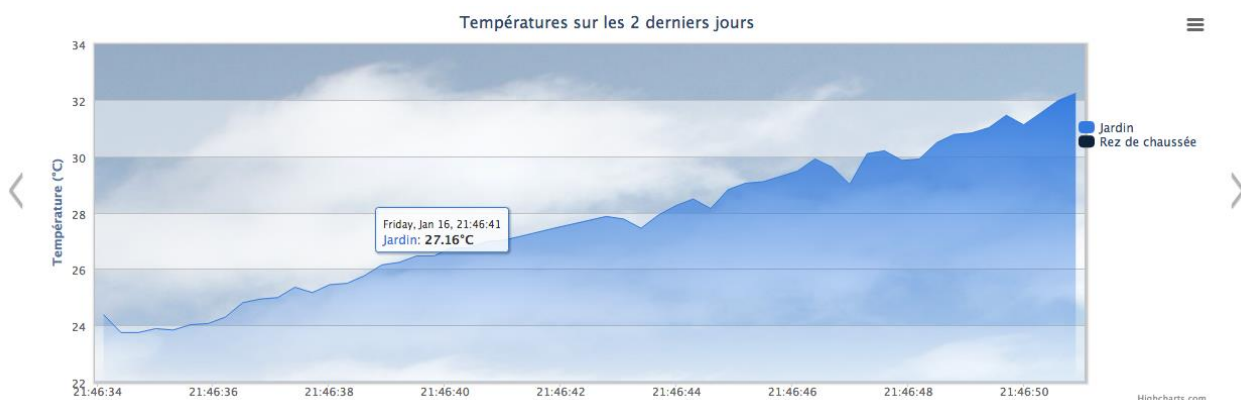
Le groupe 2 a comme première tâche de créer un fichier test.csv, ce qui le rend indépendant des progrès du groupe 1. Il convient donc que l'enseignant dispose d'un fichier test.csv qu'il peut fournir le cas échéant.

Le groupe 3 est indépendant pour sa première tâche, mais doit disposer de données brutes pour réaliser la seconde partie.

L'enseignant dispose d'un fichier Python contenant les données brutes stockées dans le fichier, pour éviter de transmettre la solution que doit fournir le groupe 2.

Améliorations possibles selon le niveau des élèves :

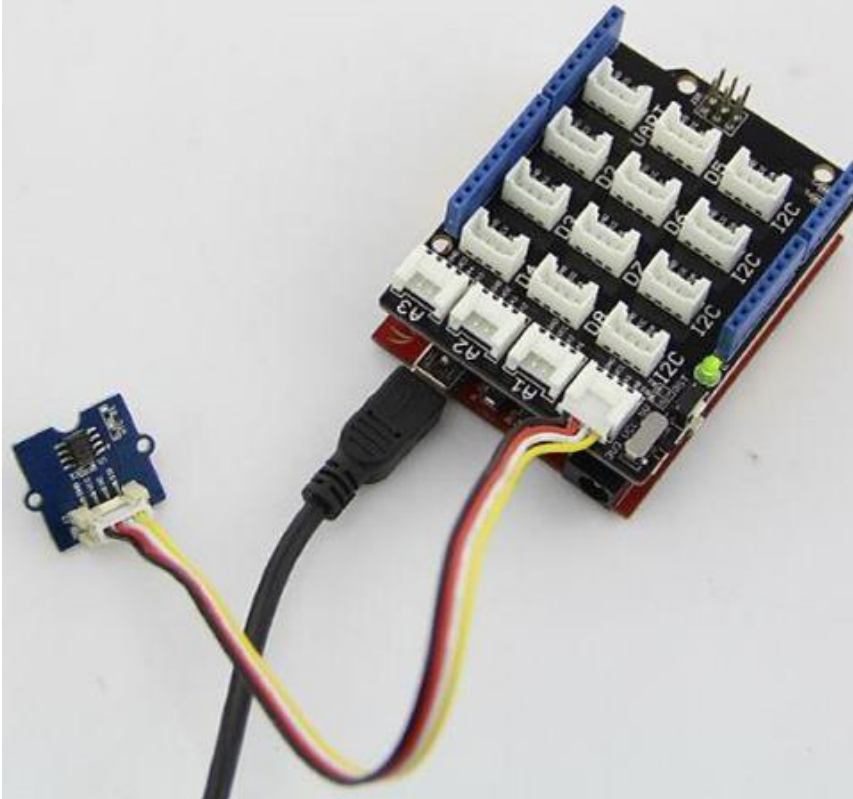
- Réaliser une transmission sans fil à partir du module arduino, dans le but d'obtenir un module déporté.
- Réaliser un réveil à intervalle régulier du module arduino pour économiser la batterie.
- Changer le style de graphique
- lire seulement les données de la dernière heure



<https://www.pihomeserver.fr/2013/07/24/station-meteo-raspberry-pi-home-server-une-station-meteo-avec-une-interface-web/>

- Exemples de projet et difficultés rencontrées

version capteur température CTN (BELKACEM)



Transfert des informations météorologiques entre l'Arduino et le PC par trame au format CSV émise toute les secondes

**Temp;Pression;Hydro\r\n**

Comme je n'avais qu'un capteur de température, j'ai simulé la pression et l'hydrométrie par une valeur arbitraire que j'auto-incrémente à chaque nouvelle trame.

L'horodatage des trames se fait sur le PC par le programme producteur.py.

La création du fichier html se fait par le programme lecteur.py.

On pourra prévoir une version :

- avec trame CSV horodatée par l'Arduino en lui ajoutant un module RTC (Real Time Clock).
- communication sans fil par Bluetooth ou Wifi

- Réflexions sur le projet

Un environnement identique entre le lycée et la maison implique un mode d'emploi détaillé sur la version Python à installer, le mode opératoire pour installer des modules ("pip install pyserial" par exemple). L'utilisation de la ligne de commande va en freiner plus d'un.

L'ensemble peut être un peu pénible du côté « Install ».

Est ce qu'il peut être envisagé d'avoir un PC dédié à ce projet ? L'environnement windows/linux peut causer des soucis de compatibilité.

Doit-on envisager de parler d'émulateur ?

## Annexe ( A distribuer aux élèves)

fiche d'activité : **station météo**

**Modalités** L'activité est à faire par groupe de 4 personnes. La restitution de l'activité se fera :

- à l'oral, le groupe devra soutenir son projet lors d'une présentation orale de dix minutes en s'appuyant sur un outil numérique. Cette présentation sera suivie d'une démonstration fonctionnelle du projet.
- à l'écrit, un compte rendu écrit de suivi, de projet au format pdf ( 4 à 6 pages), retraçant les étapes du projet et mettant en évidence les choix, les difficultés rencontrées, devra être transmis.

### **Activité : Conception d'un projet**

A partir d'une mini station de météo et de son capteur physique BMP280, en s'aidant d'une arduino, vous allez générer un fichier csv. Ce fichier devra être traité en python.

A partir de ce traitement, vous rendrez compte de vos résultats avec une page HTML comportant du CSS.

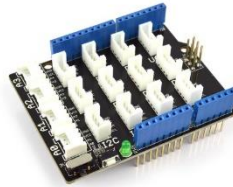
# Fiche Arduino

## Matériel nécessaire :

Arduino UNO



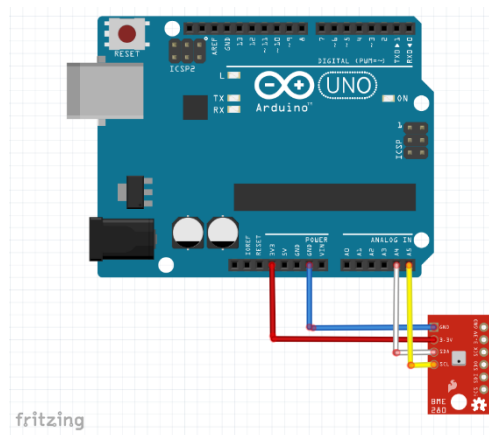
Shield grove



Capteur BME280



## On modélise le circuit par :



- Installer la bibliothèque correspondant au capteur sur le site du fabricant.
- Obtenir le résultat suivant :

```
COM4 (Arduino/Genuino Uno)

19:08:33.068 -> Pressure: 100996.00Pa
Humidity: 47%

19:08:38.042 -> Temp: 23.61C
19:08:38.077 -> Pressure: 101000.00Pa
19:08:38.077 -> Humidity: 47%

19:08:43.064 -> Temp: 23.60C
19:08:43.064 -> Pressure: 100996.00Pa
19:08:43.098 -> Humidity: 47%

19:08:48.044 -> Temp: 23.61C
19:08:48.079 -> Pressure: 101000.00Pa
Humidity: 47%

19:08:53.050 -> Temp: 23.60C
19:08:53.085 -> Pressure: 100999.00Pa
19:08:53.085 -> Humidity: 47%
```

- Modifier le programme arduino en fonction des paramètres vu avec le groupe 2
- Installer le module pyserial : <https://pypi.org/project/pyserial/>
- Vérifier la communication entre le programme Python et l'arduino

- Réaliser l'écriture du fichier CSV en Python



# Fiche Décodage CSV, création des graphiques


- Convenir avec le groupe 1 du format des données à échanger.
- Créer un fichier test.csv en cohérence avec les règles au-dessus avec un tableur, et l'exporter au format CSV.
- En utilisant la documentation : <https://docs.python.org/fr/3/library/csv.html>,

Créer un programme qui lit et affiche les informations contenues dans le fichier test.csv

- Stocker ces informations dans une liste dont le nom est convenu avec le groupe 3.
- Convenir avec le groupe 3 des noms des graphiques.
- Installer les modules numpy et matplotlib
- En utilisant la documentation : <https://matplotlib.org/>

Créer les graphiques correspondants à l'évolution de la température, pression et hygrométrie.

# Fiche Génération page HTML et CSS

- En utilisant un simple éditeur, créer une page web contenant, le titre, un tableau avec les mesures (température, pression et hygrométrie) en fonction du temps, ainsi que 3 images qui seront fournies par le groupe 2.
- En utilisant Python  créer un programme qui va réaliser la construction de la page HTML, en prenant en compte les données fournies par le groupe 2.
- Utiliser la feuille de style pour personnaliser l’affichage.

# Proposition de code du projet

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Jun 13 20:37:11 2019

@author: brembilla
"""
#projet météo

import csv
import numpy as np
import matplotlib.pyplot as plt
import serial
import datetime

def lectureFichierCsv(url):
    # PRE url valide
    # POST retourne un tableau contenant les données CSV

    tableau=[]
    with open(url, newline='') as csvfile:
        lecteur = csv.reader(csvfile, delimiter=',', quotechar='"')
        for ligne in lecteur:
            tableau.append(ligne)

    return tableau

def generationImageTableau(tableau):
    # PRE tableau des données déjà créé
    # POST création de 3 fichiers images (T=f(T), P=f(T) et Hygro=f(T))

    date=[]
    heure=[]
    temperature=[]
    pression=[]
    hygrometrie=[]
    for i in range(1,len(tableau)):
        date.append(tableau[i][0])
        heure.append(tableau[i][1])
        temperature.append( float(tableau[i][2]))          # il faut redéfinir le format
    flottant.
        pression.append( float(tableau[i][3]))             # idem
        hygrometrie.append(float(tableau[i][4]))            # idem

    x = np.array(heure)
    yt = np.array(temperature)
    yp = np.array(pression)
    yh = np.array(hygrometrie)

    creationImage(x,yt,["relevé de température","Température"])
    creationImage(x,yp,["relevé de pression","Pression"])
    creationImage(x,yh,["relevé d'hygrométrie","Hygrométrie"])

def creationImage(x,y,noms):
    # PRE x contient une liste de temps valide, y contient une liste valide de la
    grandeur dépendante du temps G
    # POST crée le graphique G=f(t) et le sauvegarde dans le fichier nom.png

    plt.title(noms[0])
    plt.ylabel(noms[1])
    plt.xlabel('Heure')
    plt.grid(True)
```



```

plt.plot(x, y)
plt.xticks(x, rotation=60) # rotation pour une meilleure
lecture du graphique
plt.subplots_adjust(bottom=0.3)
plt.savefig("graphique"+noms[1]+".png")
plt.close()

```

```

def generationPageHtml(tableau):
    # PRE    tableau des données déjà créé, les 3 images au format png sont déjà créées
    # POST   génère le fichier html "page.html" contenant un tableau avec les valeurs
brutes T=f(T), P=f(T) et Hygro=f(T) et les 3 graphiques correspondants

```

```

    fichier = open("page.html", "w") # "w" pour write (ecrase le fichier si il existe)
ou "a" pour append (ecrit à la suite du fichier)
    fichier.write("<!DOCTYPE html>\n")
    fichier.write('<span lang="fr"><p>\n')
    fichier.write('<head>\n')
    fichier.write('<meta charset="utf-8">\n')
    fichier.write('<title>Ma page météo</title>\n')
    fichier.write('\n')
    fichier.write('') # x un seconde et y l'url de la page pas nécessaire si même page
    fichier.write('</head>\n')
    fichier.write('<table><tr><td><p>\n')
    generationTableauHtml(tableau, fichier)
    fichier.write('\n')
    fichier.write('\n')
    fichier.write('\n')
    fichier.write('</td></tr></table>\n')
    fichier.write('</span>')
    fichier.close()

```

```

def generationTableauHtml(tableau, fichier):
    # PRE    tableau des données déjà créé
    # POST   écrit le tableau des données dans la page html

```

```

    fichier.write('<table>')
    fichier.write('<caption>Données Brutes</caption>')
    fichier.write('<tr>')
    for item in tableau[0]:
        fichier.write('<th>'+item+'</th>')
    fichier.write('</tr>\n')

```

```

for i in range(1, len(tableau)):

```

```

    fichier.write('<tr>')
    for item in tableau[i]:
        fichier.write('<td>'+item+'</td>')
    fichier.write('</tr>\n')

```

```

    fichier.write('</table>\n')

```

```

url='donnees.csv'

```

```

ser = serial.Serial('/dev/ttyACM1', 9600) # mettre le nom du port sur lequel est
branché votre arduino, à lire dans arduino IDE. Attention, ce port peut changer d'un
branchement au branchement suivant !
fichierCsv = open("donnees.csv", "w") # crée le fichier donnees.csv
fichierCsv.write("Date,heure,Temperature,Pression,Hygrométrie\n") # on passe les
entêtes du fichier csv et \n pour indiquer la fin de ligne (on passe à la ligne
suivante dans le fichier csv)
fichierCsv.close()

```

```

while True: # indéfiniment, le programme reçoit les
données, les écrit dans le fichier, génère la page html
    fichierCsv = open("donnees.csv", "a") # ajoute des lignes dans le fichier
donnees.csv

```

```

line = ser.readline()
newString = line.decode(encoding='UTF-8')
date = datetime.datetime.now() #
l'heure est prise sur l'ordinateur
fichierCsv.write(str(date.day)+"/"+str(date.month)+"/"+str(date.year)+",") #
format de la date au format français, suivi du séparateur ,
fichierCsv.write(str(date.hour)+":"+str(date.minute)+":"+str(date.second)+",") #
format de l'heure suivi du séparateur ,
fichierCsv.write(newString) #
newString contient les données transmises par l'arduino branché en USB (ici, temp,
pression et hygrom)
fichierCsv.close()

tableau=lectureFichierCsv(url)
generationImageTableau(tableau)
generationPageHtml(tableau)

```