

COMMENT RÉALISER UN CAPTEUR DE CO₂ AFIN DE VENTILER LA SALLE DE CLASSE DE TECHNOLOGIE ?

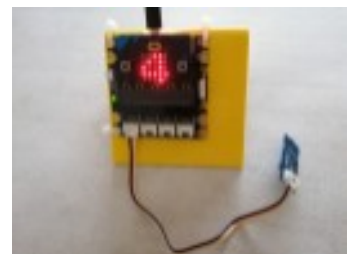
Pour éviter la propagation des virus par voie aérienne comme celle de la COVID-19, il est conseillé de renouveler l'air dans les lieux clos afin de diminuer le taux de particules en suspension dans l'air.

Problématique :

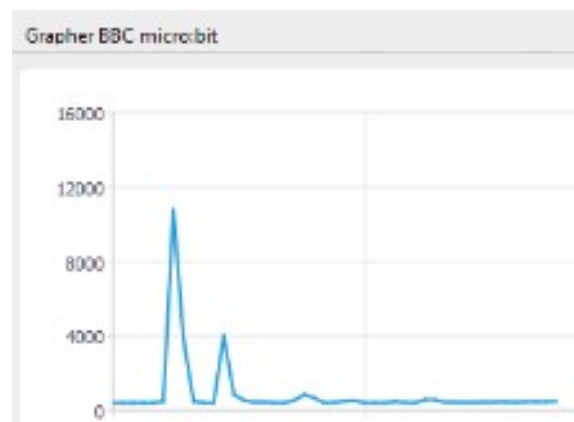
Comment savoir si une salle de classe est correctement aérée ?

```
vitta science  
no name  
1 from microbit import *  
2 from sgp30 import *  
3  
4 sgp30 = SGP30()  
5  
6 while True:  
7     display.scroll(str(sgp30.eCO2()))  
8  
9
```

```
Me 113beta0 - sgp30 avec mu editor  
Mode Nouveau Charger Enregistrer Rechercher  
sgp30 avec mu editor et graphique.py  
1 from microbit import *  
2 from sgp30 import *  
3  
4 sgp30 = SGP30()  
5  
6 while True:  
7     display.scroll(str(sgp30.eCO2()))
```



```
Mode Nouveau Charger Enregistrer Flasher  
sgp30 avec mu editor et graphique.py  
1 from microbit import *  
2 from sgp30 import *  
3  
4 sgp30 = SGP30()  
5  
6 while True:  
7     mesure = (sgp30.eCO2())  
8     sleep(1000)  
9     display.scroll(str(mesure))  
10    print((mesure,))  
11    sleep(1000)
```



Compétences à valider

CT 4.2 - Appliquer les principes élémentaires de l'algorithmique et du codage à la résolution d'un problème simple.

CS 5.7 - Analyser le comportement attendu d'un système réel et décomposer le problème posé en sous-problèmes afin de structurer un programme de commande.

CT 5.5 - Modifier ou paramétrer le fonctionnement d'un objet communicant.



COMMENT RÉALISER UN CAPTEUR DE CO₂ POUR LA CLASSE DE TECHNOLOGIE ?

SÉQUENCE 8 – ACTIVITÉ 4

Par quoi et comment programmer un objet technique ?



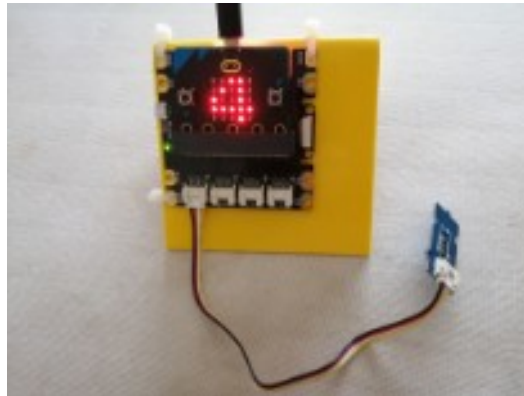
Activités à réaliser en îlot:

Temps alloué : 1h20 minutes

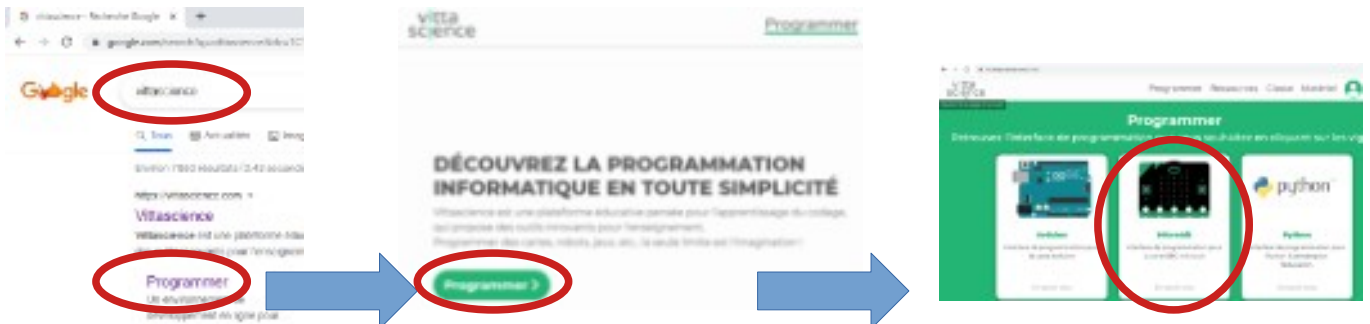
Problème à résoudre : Dans le cadre du cours de technologie, vous allez découvrir comment programmer un capteur de Co₂ afin de savoir si la classe est bien aérée.

Nous allons maintenant noter les mesures effectuées par notre maquette de capteur de CO₂ et les analyser

Utiliser la maquette microbit avec le capteur de CO₂ - branchez la en USB ou rajouter les batteries afin d'être autonome :



TRAVAIL 1 : Se connecter sur le site de VITTASCIENCE AVEC LE NAVIGATEUR CHROME :



Réaliser le programme ci-dessous en mode python :

```

vitta
science

no name
+ [lock] [file] [undo] [redo] [refresh]

1 from microbit import *
2 from sgp30 import *
3
4 sgp30 = SGP30()
5
6 while True:
7     display.scroll(str(sgp30.eCO2()))
8

```



COMMENT RÉALISER UN CAPTEUR DE CO2 POUR LA CLASSE DE TECHNOLOGIE ?

SÉQUENCE 8 – ACTIVITÉ 4

Par quoi et comment programmer un objet technique ?



FICHER SGP30.PY :

```
from microbit import i2c,sleep
class SGP30:
    def __init__(self):
        self.serial=self.read([0x36, 0x82],10,3)
        if self.read([0x20,0x2f],0.01,1)[0]!=0x0020:raise RuntimeError('SGP30 Not detected')
        self.iaq_init()
    def TVOC(self):return self.iaq_measure()[1]
    def baseline_TVOC(self):return self.get_iaq_baseline()[1]
    def eCO2(self):return self.iaq_measure()[0]
    def baseline_eCO2(self):return self.get_iaq_baseline()[0]
    def iaq_init(self):self.run(['iaq_init',[0x20,0x03],0,10])
    def iaq_measure(self):return self.run(['iaq_measure',[0x20,0x08],2,50])
    def get_iaq_baseline(self):return self.run(['iaq_get_baseline',[0x20,0x15],2,10])
    def set_iaq_baseline(self,eCO2,TVOC):
        if eCO2==0 and TVOC==0:raise RuntimeError('Invalid baseline')
        b=[]
        for i in [TVOC,eCO2]:
            a=[i>>8,i&0xFF]
            a.append(self.g_crc(a))
            b+=a
        self.run(['iaq_set_baseline',[0x20,0x1e]+b,0,10])
    def set_iaq_humidity(self,PM3):
        b=[]
        for i in [int(PM3*256)]:
            a=[i>>8,i&0xFF]
            a.append(self.g_crc(a))
            b+=a
        self.run(['iaq_set_humidity',[0x20,0x61]+b,0,10])
    def run(self,profile):
        n,cmd,s,d=profile
        return self.read(cmd,d,s)
    def read(self,cmd,d,rs):
        i2c.write(0x58,bytearray(cmd))
        sleep(d)
        if not rs:return None
        cr=i2c.read(0x58,rs*3)
        o=[]
        for i in range(rs):
            w=[cr[3*i],cr[3*i+1]]
            c=cr[3*i+2]
            if self.g_crc(w)!=c:raise RuntimeError('CRC Error')
            o.append(w[0]<<8|w[1])
        return o
    def g_crc(self,data):
        c=0xFF
        for byte in data:
            c^=byte
            for _ in range(8):
                if c&0x80:c=(c<<1)^0x31
                else:c<<=1
            return c&0xFF
```



COMMENT RÉALISER UN CAPTEUR DE CO2 POUR LA CLASSE DE TECHNOLOGIE ?

SÉQUENCE 8 – ACTIVITÉ 4

Par quoi et comment programmer un objet technique ?



TRAVAIL 2 : Enregistrer le programme en python :

The screenshot shows the Vitta Science web interface. On the left, a code editor contains the following Python code:

```

1 from microbit import *
2 from sgp30 import *
3
4 sgp30 = SGP30()
5
6 while True:
7     display.scroll(str(sgp30.eCO2()))
8

```

On the right, a modal window titled "Sauvegarder le projet" (Save the project) is displayed. It contains the text: "Pour enregistrer votre projet, veuillez vous connecter. Je me connecte ou Je m'inscris !". Below this text are three buttons: "Se connecter", "Créer une compte", and "Partager". At the bottom of the modal, there is a green button labeled "Télécharger le programme en Python".

Below the code editor, a warning message states: "Ce type de fichier risque d'endommager votre ordinateur. Voulez-vous vraiment enregistrer main.py ?". To the right of this message are two buttons: "Enregistrer" and "Supprimer".

Blue arrows indicate the flow: from the code editor to the save modal, and from the warning message to the "main.py" file icon.

TRAVAIL 3 : Enregistrer le programme .hex :

The screenshot shows the Vitta Science web interface. The code editor contains the same Python code as in the previous step:

```

1 from microbit import *
2 from sgp30 import *
3
4 sgp30 = SGP30()
5
6 while True:
7     display.scroll(str(sgp30.eCO2()))
8

```

On the right side of the interface, there is a button labeled "Télécharger .hex". A blue arrow points from this button to a file icon labeled "microbit_1635696....hex".

TRAVAIL 4 : Transférer le programme .hex sur la carte microbit :

The screenshot shows a Windows File Explorer window. The left sidebar shows the navigation pane with "Ce PC" expanded. The "Disque local (C:)" and "MICROBIT (D:)" drives are visible. A blue arrow points from the "microbit_1635696....hex" file icon to the "MICROBIT (D:)" drive.

On the right side, there is a photograph of a yellow Microbit board with a red LED lit up. A blue USB cable is plugged into the board.



COMMENT RÉALISER UN CAPTEUR DE CO₂ POUR LA CLASSE DE TECHNOLOGIE ?

SÉQUENCE 8 – ACTIVITÉ 4

Par quoi et comment programmer un objet technique ?



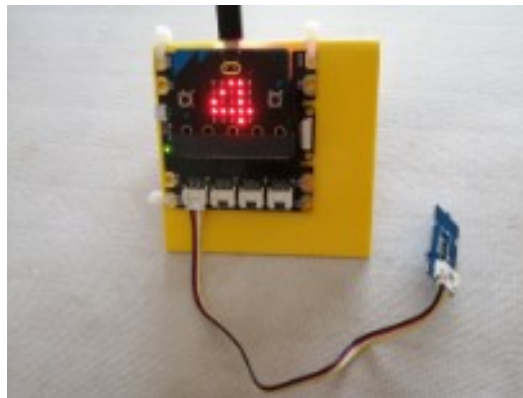
Activités à réaliser en îlot:

Temps alloué : 1h20 minutes

Problème à résoudre : Dans le cadre du cours de technologie, vous allez découvrir comment programmer un capteur de Co₂ afin de savoir si la classe est bien aérée.

Nous allons maintenant noter les mesures effectuées par notre maquette de capteur de CO₂ et les analyser

Utiliser la maquette microbit avec le capteur de CO₂ - branchez la en USB ou rajouter les batteries afin d'être autonome :



TRAVAIL 1 : Lancer le programme MU EDITOR :



Réaliser le programme ci-dessous en mode python :

```
sgp30 avec mu editor.py  X
1 from microbit import *
2 from sgp30 import *
3
4 sgp30 = SGP30()
5
6 while True:
7     display.scroll(str(sgp30.eCO2()))
```

```
from microbit import *
from sgp30 import *
```

```
sgp30 = SGP30()
```

```
while True:
    display.scroll(str(sgp30.eCO2()))
```



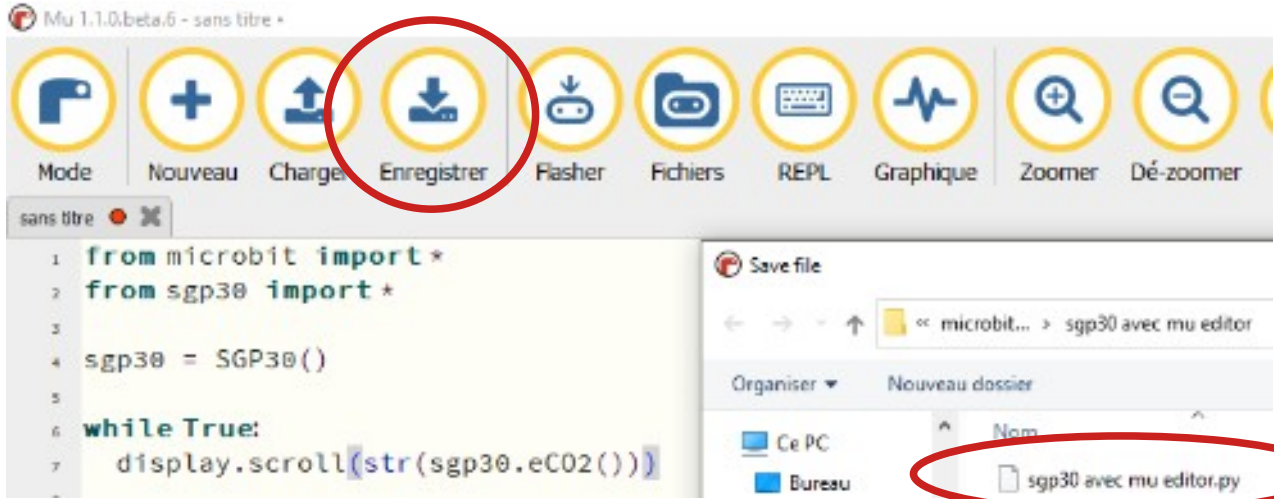

COMMENT RÉALISER UN CAPTEUR DE CO2 POUR LA CLASSE DE TECHNOLOGIE ?

SÉQUENCE 8 – ACTIVITÉ 4

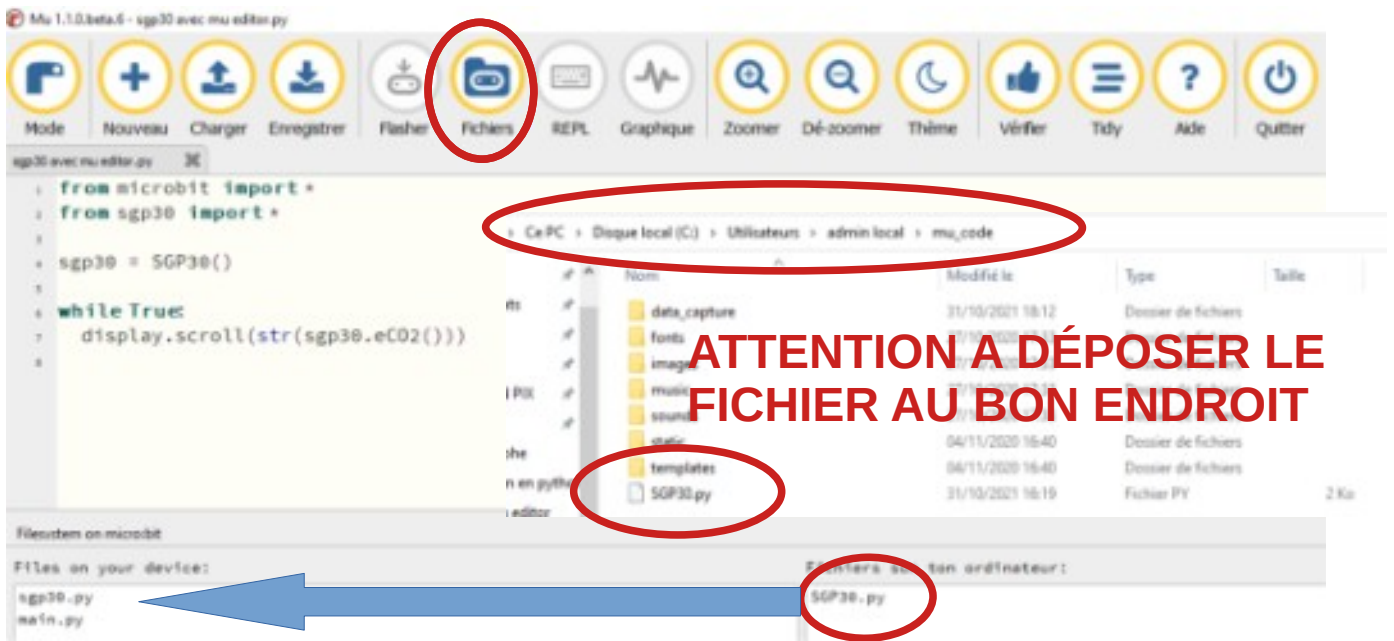
Par quoi et comment programmer un objet technique ?



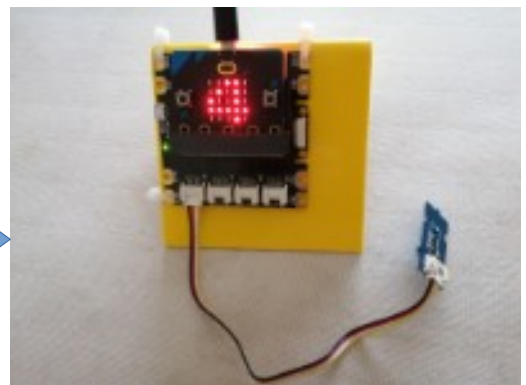
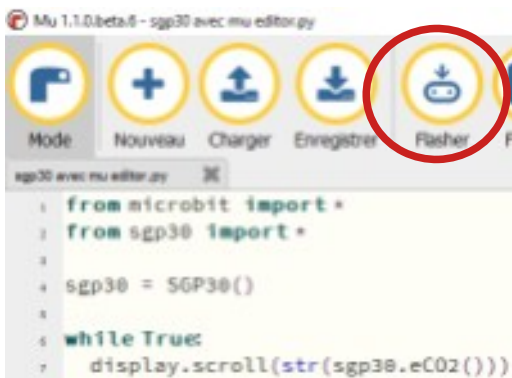
TRAVAIL 2 : Enregistrer le programme en python :



TRAVAIL 3 : Transférer le fichier bibliothèque SGP30.py dans la carte microbit :



TRAVAIL 4 : Transférer le programme sur la carte microbit :





COMMENT RÉALISER UN CAPTEUR DE CO2 POUR LA CLASSE DE TECHNOLOGIE ?

SÉQUENCE 8 – ACTIVITÉ 4

Par quoi et comment programmer un objet technique ?



TRAVAIL 5 : Modifier le programme en python afin d'afficher un graphique :

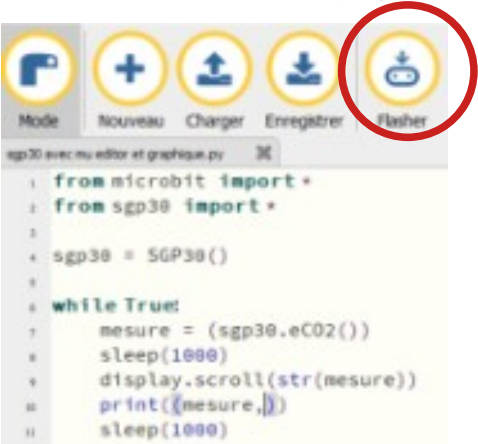
```
sgp30 avec mu editor et graphique.py
1 from microbit import *
2 from sgp30 import *
3
4 sgp30 = SGP30()
5
6 while True:
7     mesure = (sgp30.eCO2())
8     sleep(1000)
9     display.scroll(str(mesure))
10    print((mesure,))
11    sleep(1000)
```

```
from microbit import *
from sgp30 import *
```

```
sgp30 = SGP30()
```

```
while True:
    mesure = (sgp30.eCO2())
    sleep(500)
    display.scroll(str(mesure))
    print((mesure, ))
    sleep(500)
```

TRAVAIL 6 : Transférer le programme sur la carte microbit :



TRAVAIL 7 : Afficher le REPL et le graphique :

