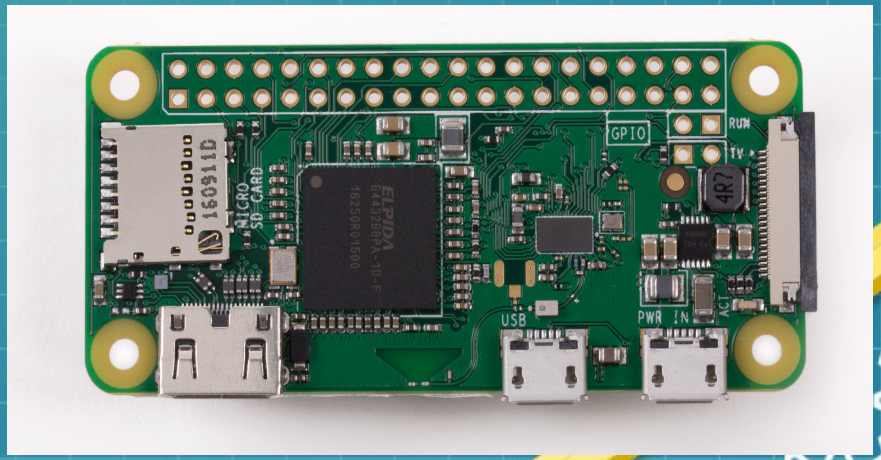
Test de la sonde à CO2 MHZ-19 sur Raspberry pi Zero W



- Présentation MHZ19 et Rasp pi Zero W.
- Constitution et fonctionnement.
- Montage
- Logiciel sous python3 : CO2.py
- Prix et fournisseur.
- Version UART/USB pour PC.

Présentation MHZ19 et Rasp pi Zero W.

Constitution et fonctionnement de la sonde CO2

Elle est constituée d'une chambre de mesure (boîtier) contenant une cellule émettant un rayon infrarouge (source lumineuse) et d'un récepteur ainsi qu'un filtre pour les interférences. Le filtre placé devant le détecteur empêche les autres ondes que celle utilisées pour la mesure d'atteindre le détecteur.
Ce récepteur infrarouge mesure l'intensité du flux après son absorption par le C02.
La poussière, la vapeur d'eau sont normalement sans effet sur la précision de la mesure.
La plage de mesure des sondes est de l'ordre de 0 à 5000 ppm* pour le modèle MHZ19 B*.
Le signal de sortie suivant les modèles va de 0-10 V ou 0-20 mA, ce signal est proportionnel à la concentration mesurée.

***ppm=parties par million**
MHZ19 A = 0 à 2000 ppm

Nota : Cette sonde relève aussi la température.

Le dioxyde de carbone et d'autres
gaz composés d'au moins deux atomes différents absorbent le rayonnement infrarouge (RI) de manière spécifique et unique. Ces
gaz sont détectables à l'aide de techniques par RI. La vapeur d'eau,
le méthane, le dioxyde de carbone
et le monoxyde de carbone sont des exemples de gaz mesurables
par un capteur à RI.

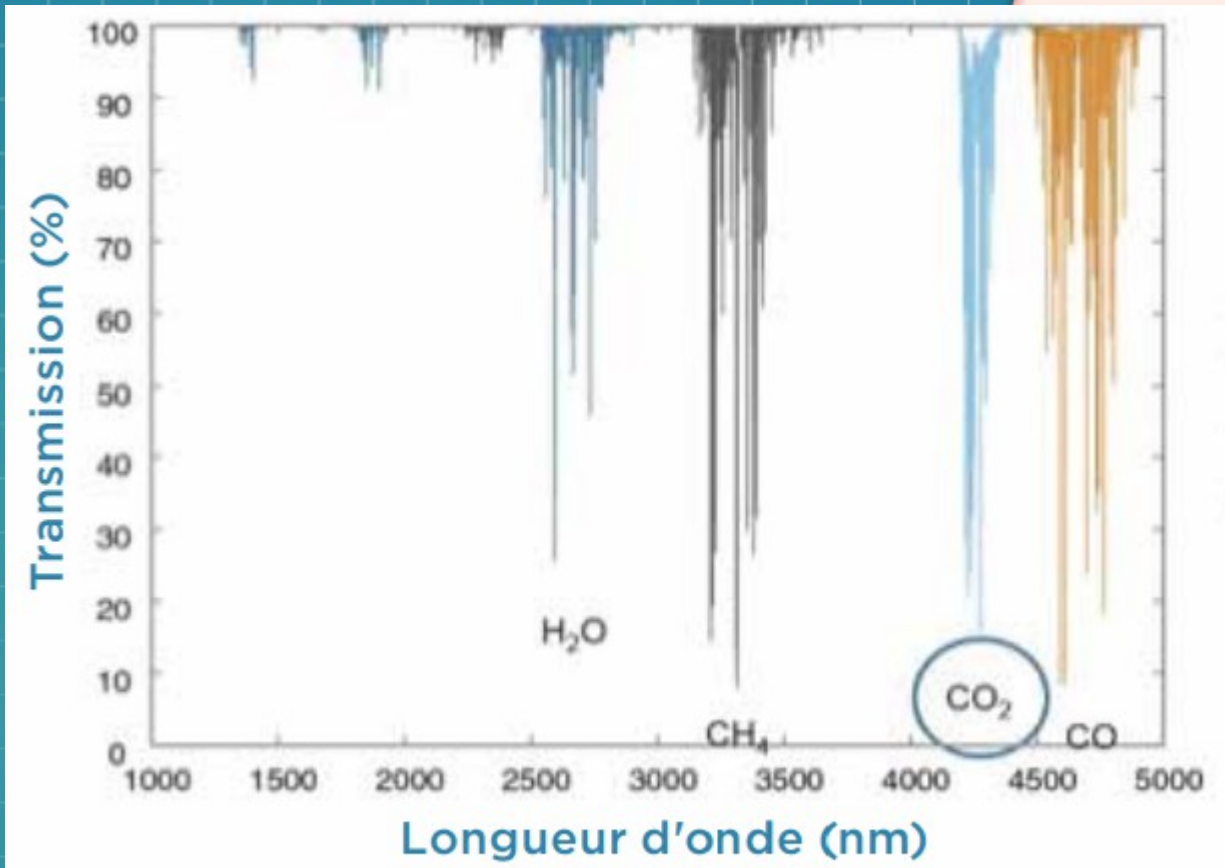


Figure 1. Absorption par RI du CO_2 et d'autres gaz.

Schéma illustrant le principe de fonctionnement d'un capteur à CO2

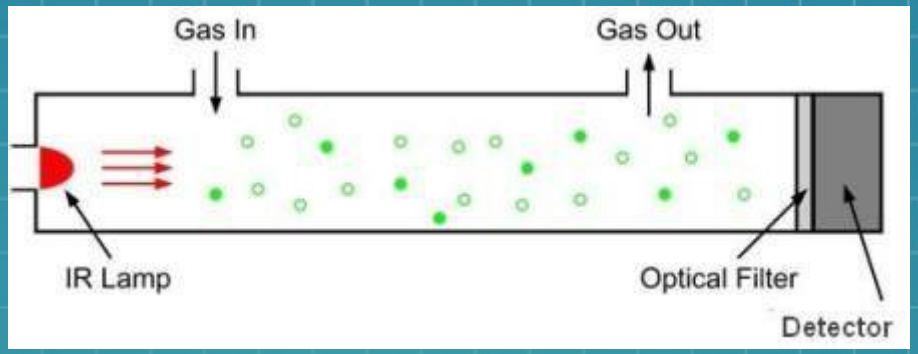
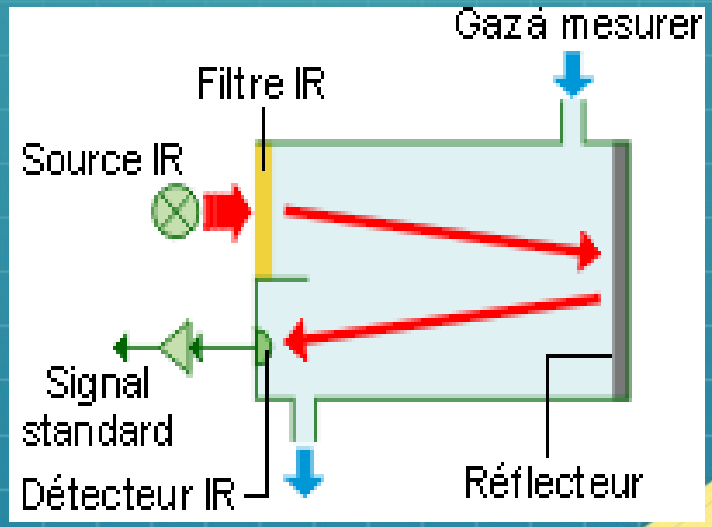
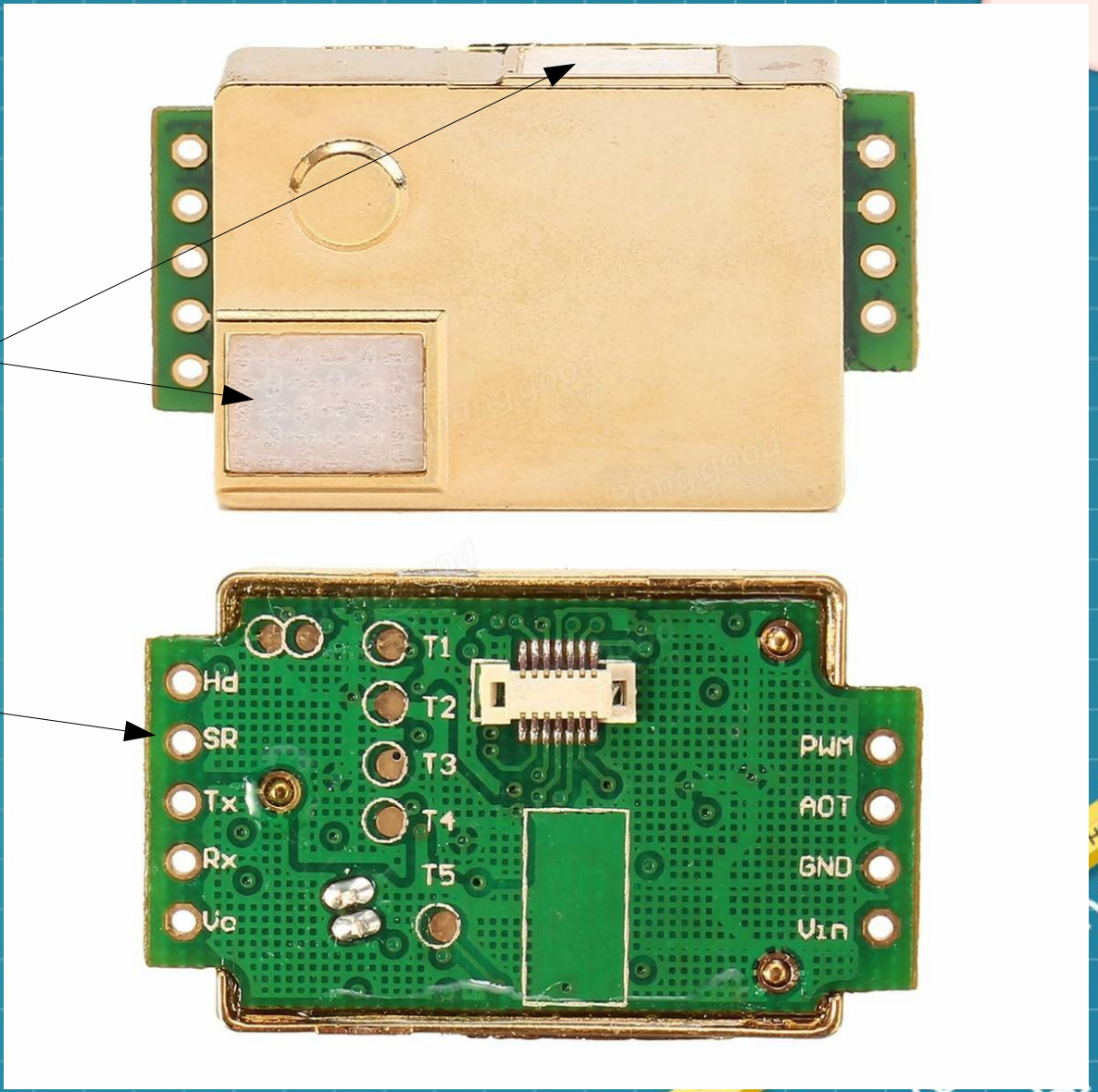


Schéma illustrant le principe de fonctionnement d'un capteur MH-Z19



Entrée / Sortie des gaz

Connections GPIO vers GPIO Raspberry



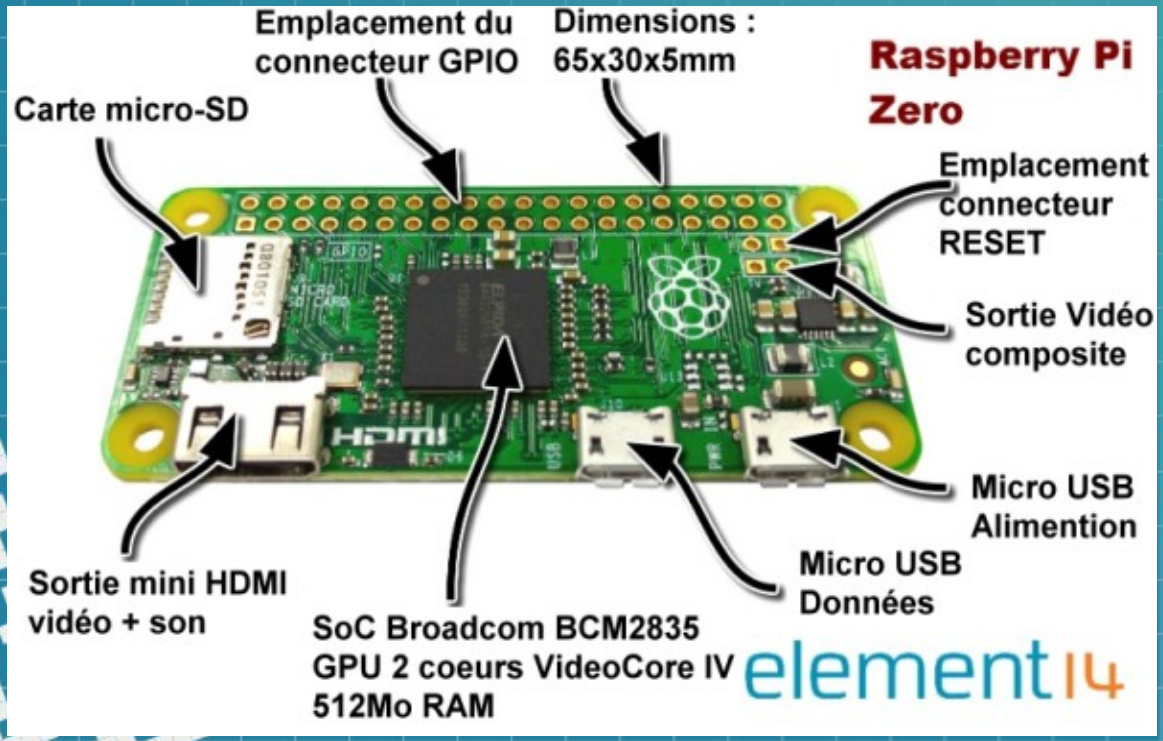
Sonde MHZ19
Recto-Verso

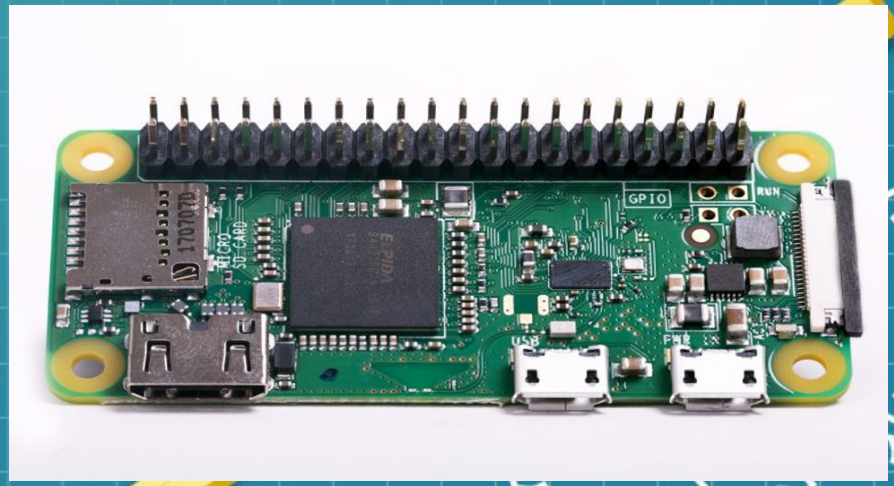
Technical Specifications MH-Z19	
Target gas	Carbon Dioxide CO2
Operating Voltage	3.6 to 5.5 Vdc
Operating current	< 18mA average
Interface levels	3.3 Vdc
Output signal format	UART or PWM
Preheat time	3 min
Response time	<60 s
Accuracy	± (50 ppm+5% reading value)
Measuring range	0 to 5000 ppm
Operating temperature range	0 to + 50°C
Dimensions	33mm×20mm×9mm(L×W×H)

Mini-PC : Raspberry pi zero W

Un des plus petits ordinateurs actuels. Le disque dur est une carte micro-sd contenant l'environnement de type Linux de préférence (Xubuntu, Debian Jessie...). Il possède une connexion wifi, bluetooth, deux ports micro-usb (un d'alimentation, l'autre pour les périphériques externes comme souris, clavier ou clefs usb), une sortie micro-HDMI et des ports GPIO*. Ces broches métalliques sont des ports d'entrée/sortie et permettent la communication avec des périphériques comme les sondes.

***General Purpose Input/Output, littéralement Entrée/Sortie pour un Usage Général**



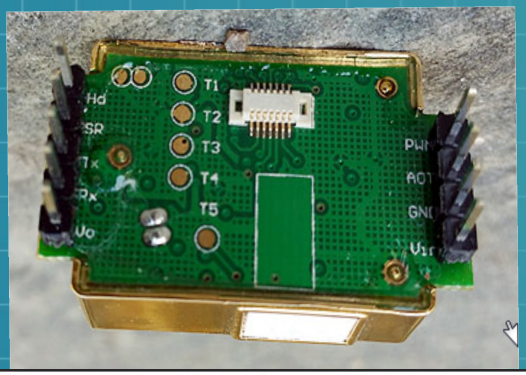


Montage.

Il faut relier les pins de chaque GPIO selon les correspondances suivantes :

Function	Raspi pin	MH-Z19 pin
Vcc +5V	2 +5V	6 Vin
GND	6 GND	7 GND
UART	8 TXD0	2 RXD
UART	10 RXD0	3 TXD

MHZ-19



PIN	Description
Pin 6	Vin (voltage input)
Pin 7	GND
Pin 1	Vout (output voltage 3.3V, output current lower than 10mA)
Pin 9	PWM
Pin 5	HD (zero calibration, low level above 7 seconds) (Factory Reserved)
Pin 2	UART (RXD) 0~3.3V digital input
Pin 3	UART (TXD) 0~3.3V digital output
Pin 4	SR (Factory Reserved)
Pin 8	AOT (Factory Reserved)

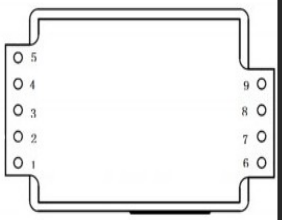
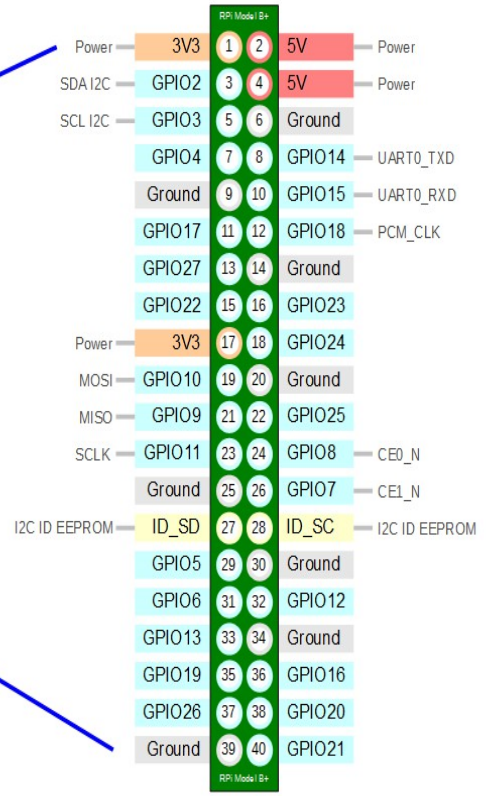
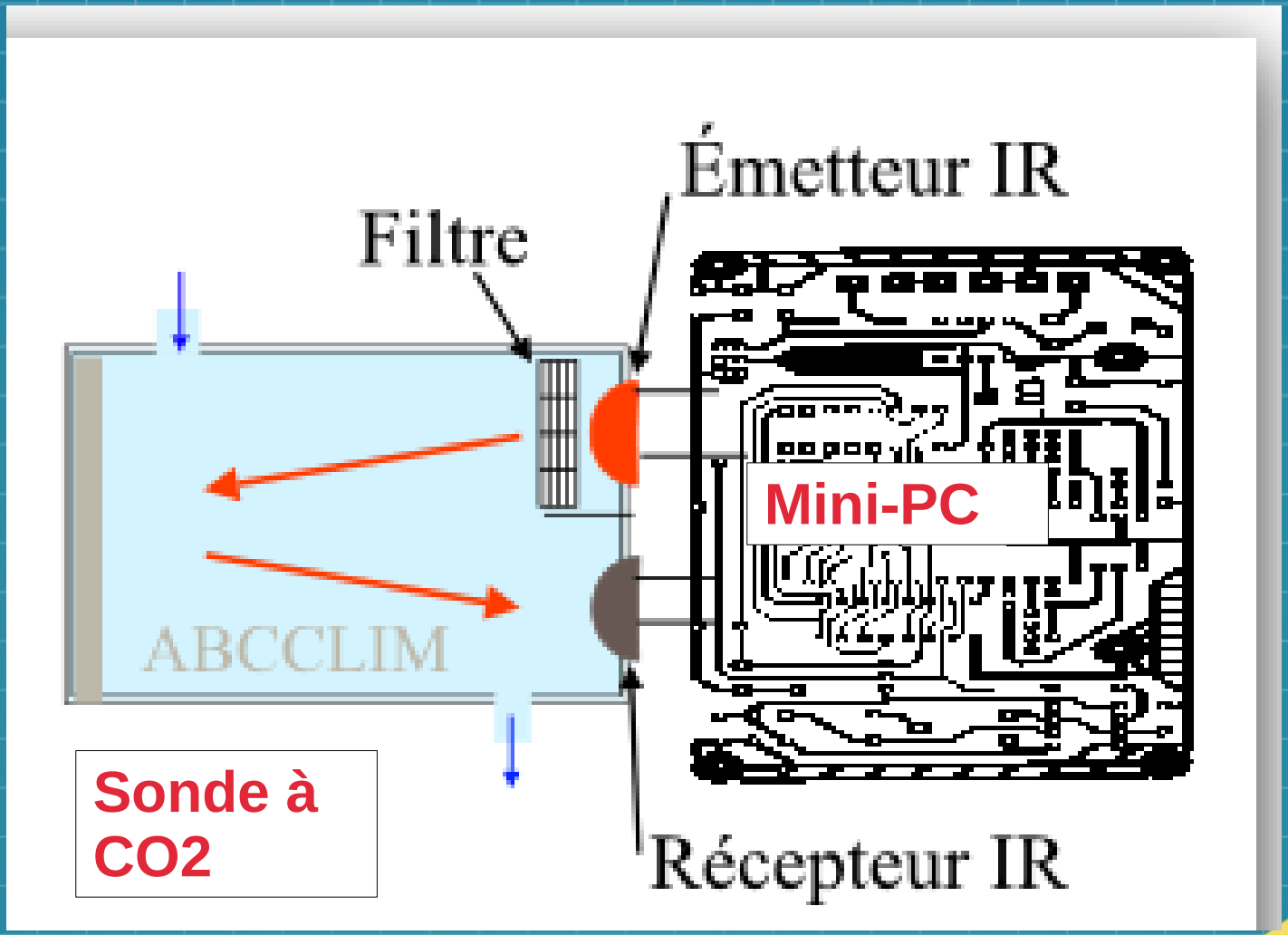


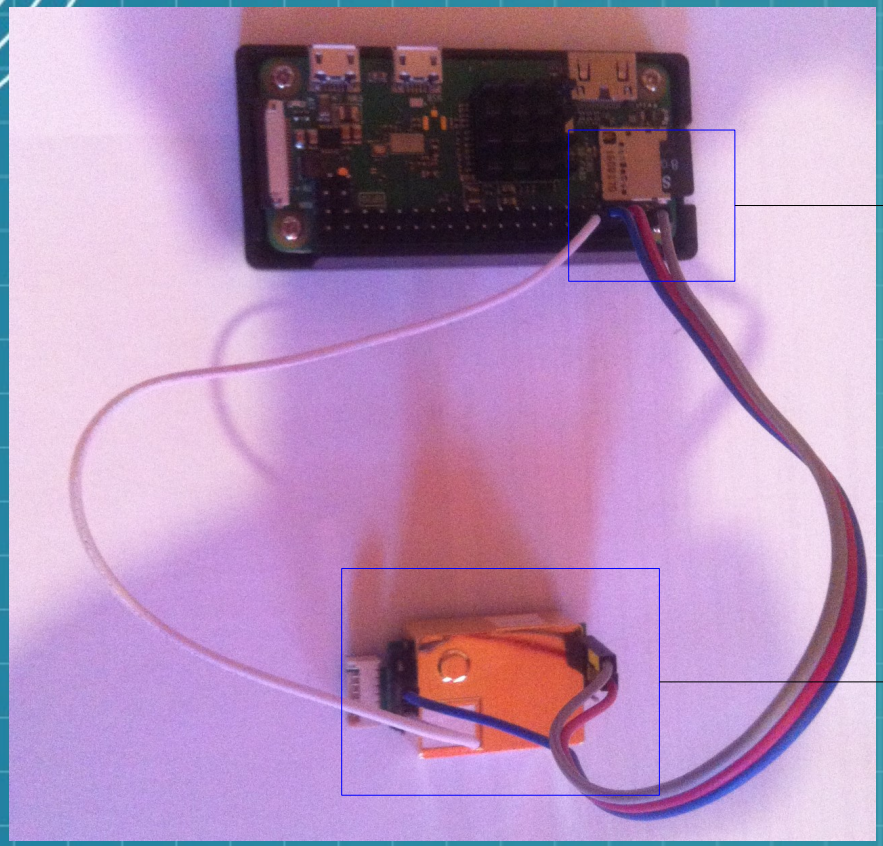
Fig.6. MH-Z19 pin definitions.

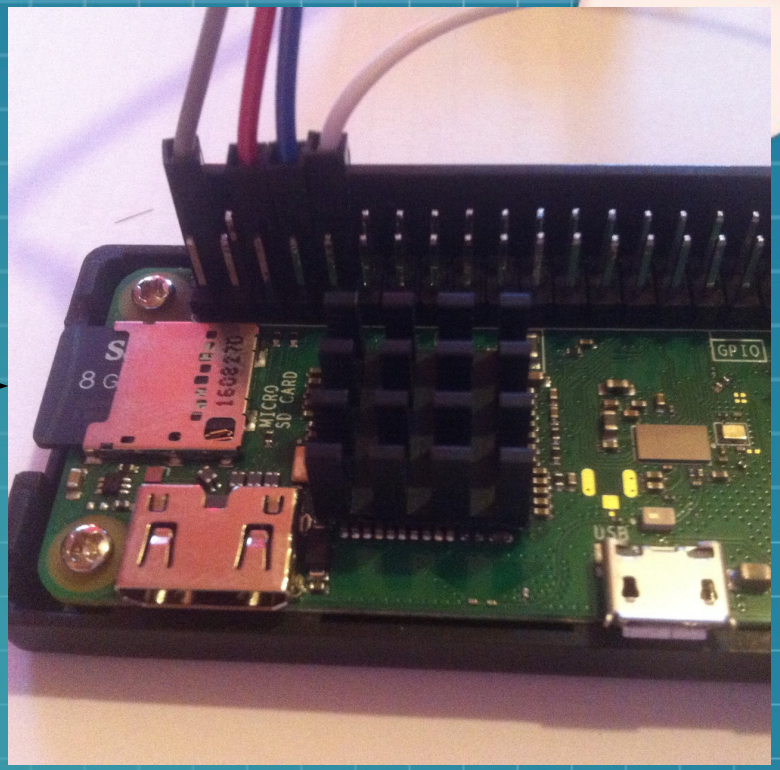
Raspberry pi zero W





Sonde CO2 émetteur et récepteur reliée à un mini-PC




GPIO
MICRO SD CARD
USB


MH-Z19B
0-5000ppm
20180404

Logiciel sous python3 : CO2.py

Vérifier que co2.py (ou une autre version) est présent dans */home/pi*.

Ouvrir votre terminal en mode administrateur (*sudo*).

Tapez *ls* puis entrée

Vous voyez apparaître co2.py dans la liste des éléments présents dans home.

Tapez : *sudo python3 co2.py*
Entrez votre mot de passe administrateur.

Les données récupérées par la sonde commencent à apparaître dans le terminal et seront enregistrées dans un fichier co2.csv dans /home/pi. Elles auront la forme : date – heure – CO2 ppm – Température en °C.

En parallèle une fenêtre affiche un graphique d'animation immédiate comportant une courbe de température (verte) et une pour la concentration en CO2 (rouge). Cette image peut être enregistrée à tout moment ou le sera automatiquement en fin d'expérience.

Choix dans co2.py

Vous pouvez activer certaines options en rajoutant à la fin de la ligne de commande *sudo python3 co2.py* différentes lettres :

python3 co2.py -h fera apparaître l'aide et les différentes options de démarrage.

python3 co2.py -o permet de choisir le dossier de destination du fichier .csv

python3 co2.py -i permet de choisir l'intervalle de temps entre chaque mesure (2 sec. Minimum)

python3 co2.py -t permet de définir la durée de l'expérience (par défaut 600 sec.)

python3 co2.py -f permet de choisir le dossier de destination du fichier .jpg

python3 co2.py -p permet de choisir le port GPIO si l'on change de mini-PC

python3 co2.py -l permet de choisir le type de tracé des courbes (-/--/-.)

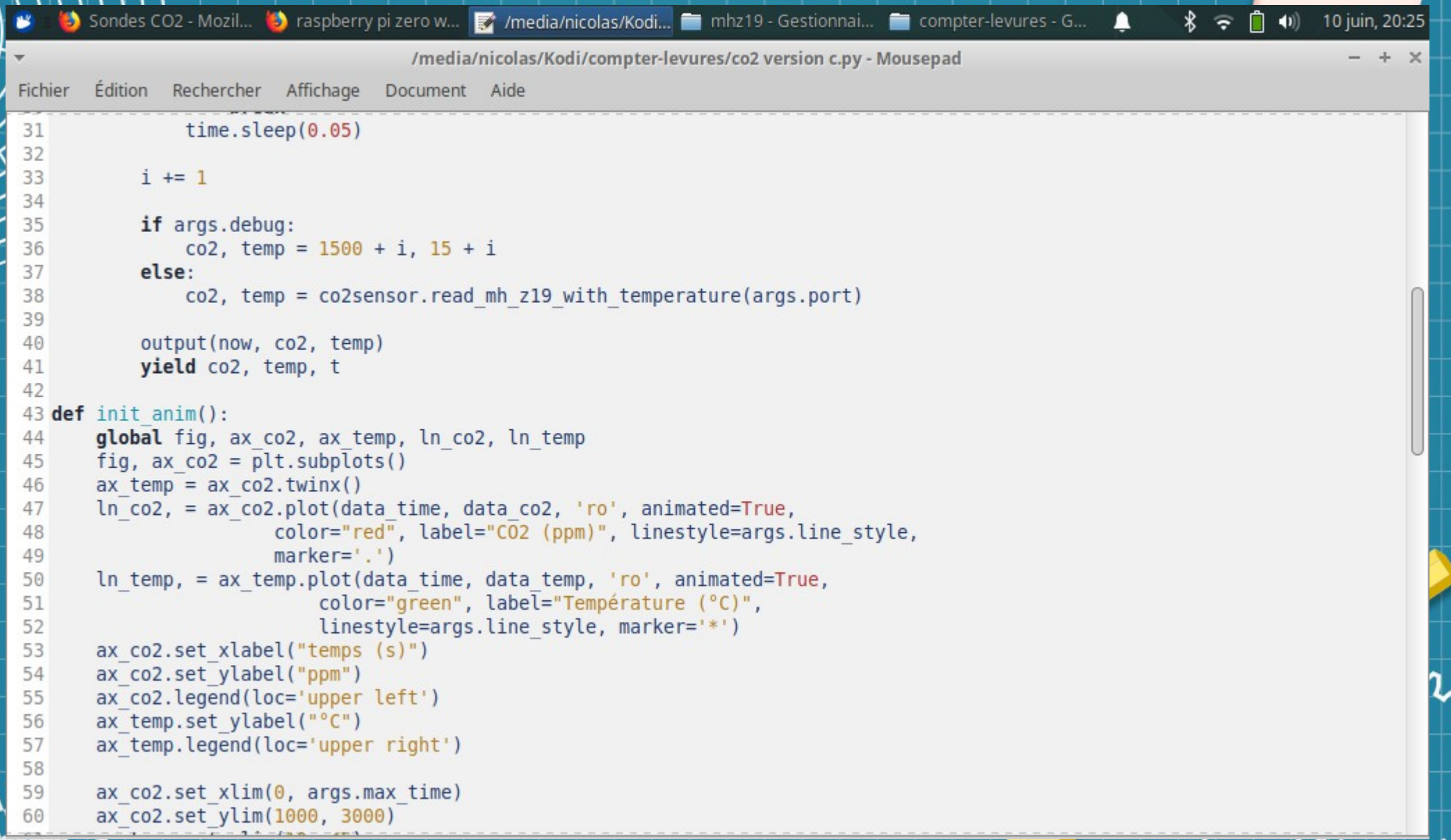
Programme

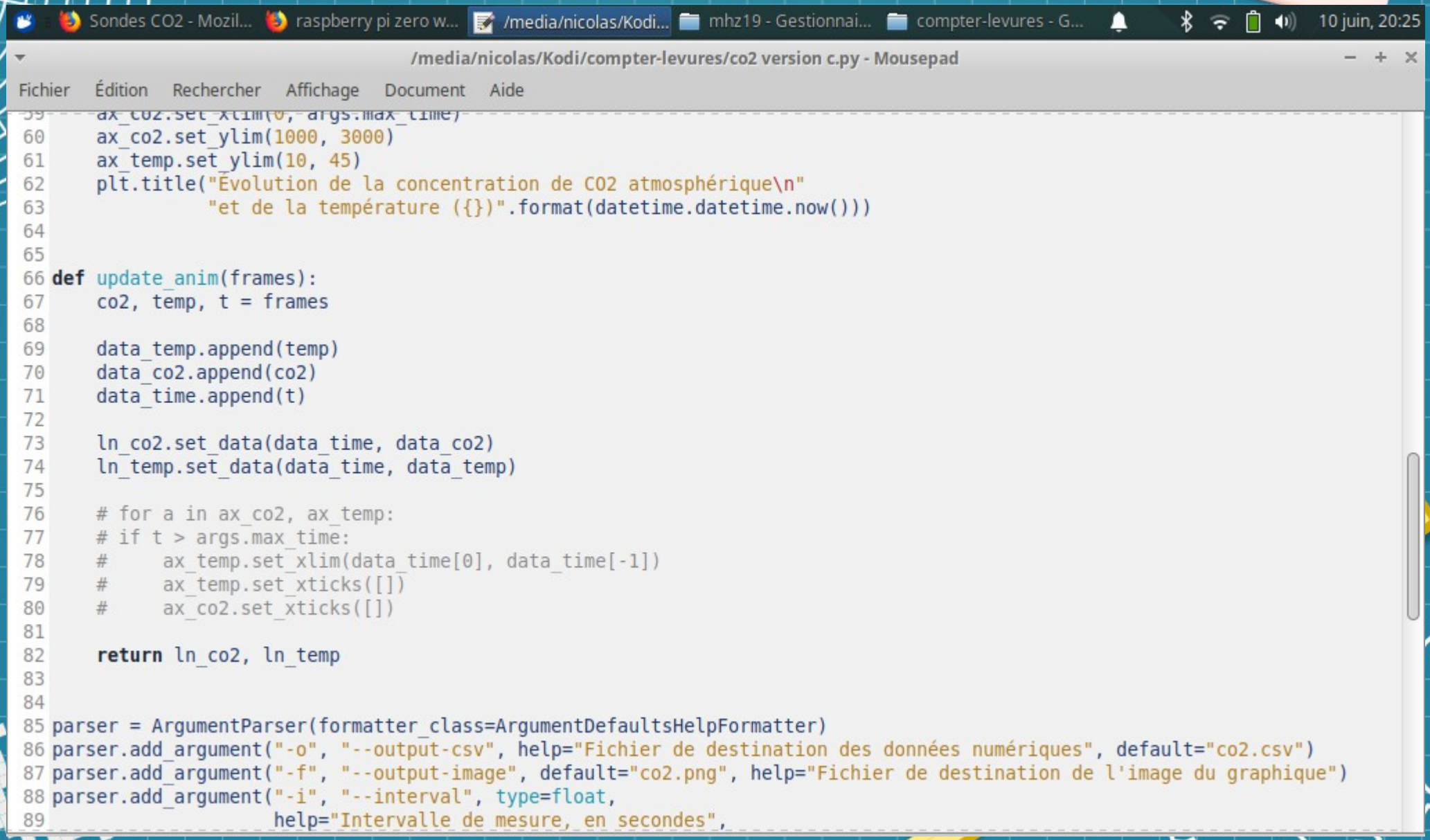
Sondes CO2 - Mozil... raspberry pi zero w... /media/nicolas/Kodi... mhz19 - Gestionnai... compter-levures - G... 10 juin, 20:23

/media/nicolas/Kodi/compter-levures/co2 version c.py - Mousepad

Fichier Édition Rechercher Affichage Document Aide

```
'''
@author: Nicolas Cedilnik & Nicolas Marangé
'''
import datetime
import time
from argparse import ArgumentParser, ArgumentDefaultsHelpFormatter

from matplotlib import pyplot as plt
from matplotlib.animation import FuncAnimation
from pmsensor import co2sensor

def output(time, concentration, temperature):
    """
    La fonction qui affiche et éventuellement sauvegarde les mesures
    """
    print("{}: {} ppm - {} °C".format(time, concentration, temperature))
    with open(output_file, 'a') as file:
        print("{},{},{}".format(time, concentration, temperature),
              file=file)

def get_co2_and_temp():
    i = 1
    start = datetime.datetime.now()
    t = 0
    while t < args.max_time:
        while True:
            now = datetime.datetime.now()
            t = (now - start).total_seconds()
            if t > i * args.interval:
                break
            time.sleep(0.05)
```


Sondes CO2 - Mozil...
raspberry pi zero w...
/media/nicolas/Kodi...
mhz19 - Gestionnai...
compter-levures - G...
10 juin, 20:25
/media/nicolas/Kodi/compter-levures/co2 version c.py - Mousepad
Fichier Édition Rechercher Affichage Document Aide
31 time.sleep(0.05)
32
33 i += 1
34
35 if args.debug:
36 co2, temp = 1500 + i, 15 + i
37 else:
38 co2, temp = co2sensor.read_mh_z19_with_temperature(args.port)
39
40 output(now, co2, temp)
41 yield co2, temp, t
42
43 def init_anim():
44 global fig, ax_co2, ax_temp, ln_co2, ln_temp
45 fig, ax_co2 = plt.subplots()
46 ax_temp = ax_co2.twinx()
47 ln_co2, = ax_co2.plot(data_time, data_co2, 'ro', animated=True,
48 color="red", label="CO2 (ppm)", linestyle=args.line_style,
49 marker='.')
50 ln_temp, = ax_temp.plot(data_time, data_temp, 'ro', animated=True,
51 color="green", label="Température (°C)",
52 linestyle=args.line_style, marker='*')
53 ax_co2.set_xlabel("temps (s)")
54 ax_co2.set_ylabel("ppm")
55 ax_co2.legend(loc='upper left')
56 ax_temp.set_ylabel("°C")
57 ax_temp.legend(loc='upper right')
58
59 ax_co2.set_xlim(0, args.max_time)
60 ax_co2.set_ylim(1000, 3000)


Sondes CO2 - Mozil...
raspberry pi zero w...
/media/nicolas/Kodi...
mhz19 - Gestionnai...
compter-levures - G...
10 juin, 20:25
/media/nicolas/Kodi/compter-levures/co2 version c.py - Mousepad
Fichier Édition Rechercher Affichage Document Aide
ax_co2.set_xlim(0, args.max_time)
ax_co2.set_ylim(1000, 3000)
ax_temp.set_ylim(10, 45)
plt.title("Évolution de la concentration de CO2 atmosphérique\n"
"et de la température ({})".format(datetime.datetime.now()))
def update_anim(frames):
co2, temp, t = frames
data_temp.append(temp)
data_co2.append(co2)
data_time.append(t)
ln_co2.set_data(data_time, data_co2)
ln_temp.set_data(data_time, data_temp)
# for a in ax_co2, ax_temp:
# if t > args.max_time:
# ax_temp.set_xlim(data_time[0], data_time[-1])
# ax_temp.set_xticks([])
# ax_co2.set_xticks([])
return ln_co2, ln_temp
parser = ArgumentParser(formatter_class=ArgumentDefaultsHelpFormatter)
parser.add_argument("-o", "--output-csv", help="Fichier de destination des données numériques", default="co2.csv")
parser.add_argument("-f", "--output-image", default="co2.png", help="Fichier de destination de l'image du graphique")
parser.add_argument("-i", "--interval", type=float,
help="Intervalle de mesure, en secondes",

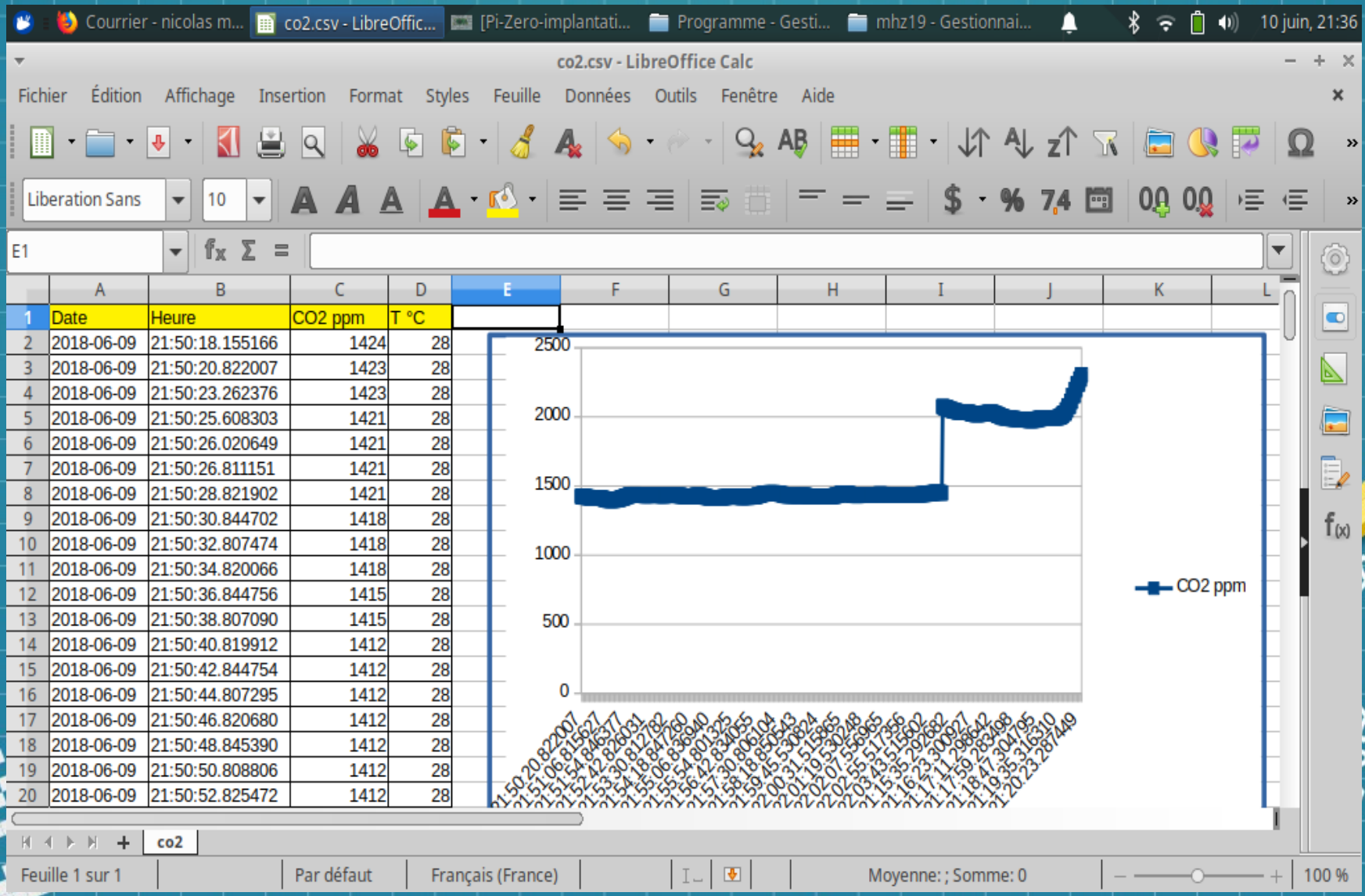
/media/nicolas/Kodi/compter-levures/co2 version c.py - Mousepad

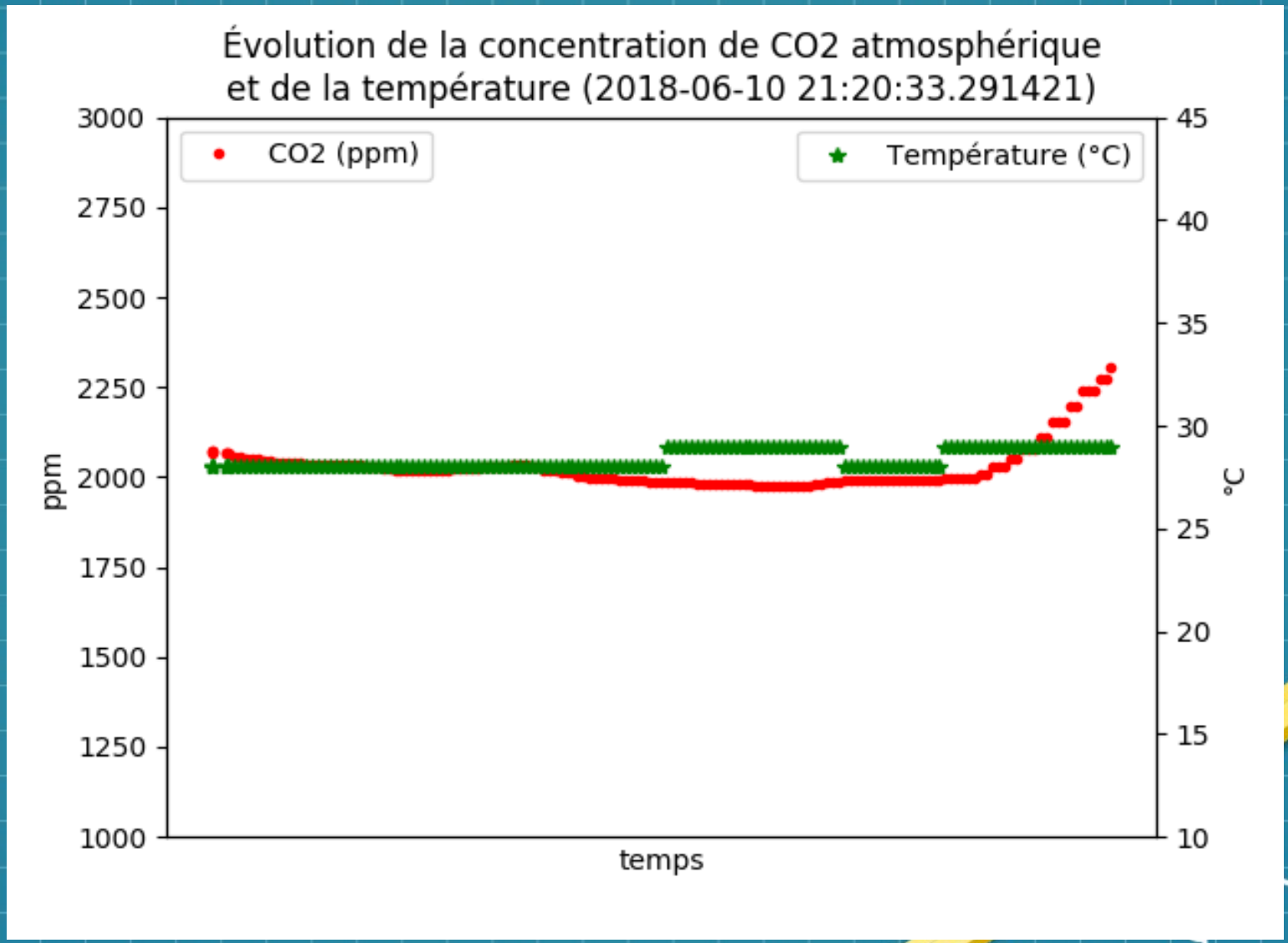
Fichier Édition Rechercher Affichage Document Aide

```
                    default=2)
parser.add_argument("-p", "--port", default="/dev/serial0", help="Le port sur lequel est branché la sonde")
parser.add_argument("-d", "--debug", default=False, action="store_true",
                    help="Ne pas lire les données de la sonde, mais utiliser des valeurs factices.")
parser.add_argument("-l", "--line-style", default='',
                    help="Le style de line affiché peut être '', :', '-', '--' ou '-.'")
parser.add_argument("-t", "--max-time", default=600.0, type=float,
                    help="Le maximum initial de l'axe des abscisses")
args = parser.parse_args()

output_file = args.output_csv

print("On écrit dans {}".format(output_file))

# mesure qui sert à rien
if args.debug:
    co2, temp = 1500, 10
else:
    co2, temp = co2sensor.read_mh_z19_with_temperature(args.port)

data_co2 = []
data_temp = []
data_time = []
init_anim()

print("Appuyez sur ALT+F4 pour quitter")
ani = FuncAnimation(fig, update_anim,
                    frames=get_co2_and_temp, blit=True, repeat=False)
plt.show()

# ax_co2.set_xticks([0, data_time[-1] / 2, data_time[-1]])
# ax_co2.set_ylim(min(data_co2), max(data_co2))
# ax_temp.set_ylim(min(data_temp), max(data_temp))
fig.savefig(args.output_image)
print("Graphique enregistré dans {}".format(args.output_image))

```

Résultats

Fichier .csv ouvert avec un tableur , ici LibreOffice Calc



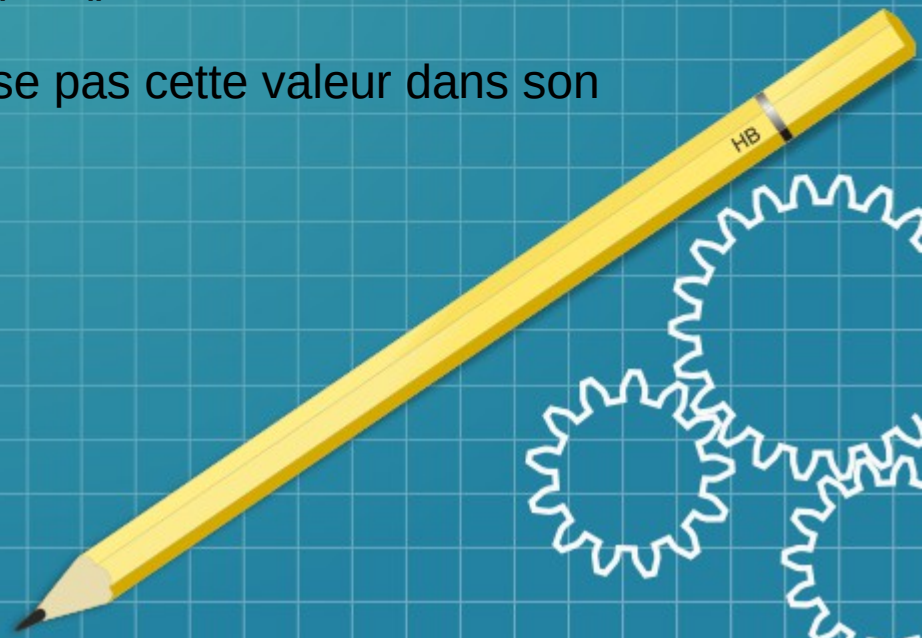

Évolution de la concentration de CO2 atmosphérique
et de la température (2018-06-10 21:20:33.291421)
CO2 (ppm)
Température (°C)
ppm
°C
temps
3000
2750
2500
2250
2000
1750
1500
1250
1000
45
40
35
30
25
20
15
10

Avantages

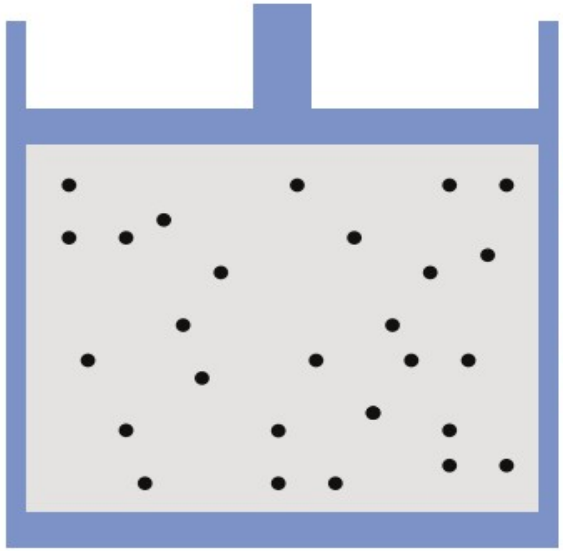
Les capteurs à RI présentent plusieurs avantages en comparaison des capteurs électrochimiques Ils sont stables et hautement sélectifs par rapport au gaz mesuré. Ils sont plus endurants et, puisque le gaz mesuré n'interagit pas directement avec le capteur, les capteurs à RI peuvent supporter des niveaux élevés d'humidité, de poussière, d'encrassement...

Inconvénients

Ces capteurs ne tiennent pas compte de la pression atmosphérique du moment et du lieu. Ils ont été calibrés à 1500 ppm (pression au niveau zero marin d'après le constructeur).
Ce détecteur affiche la température mais n'utilise pas cette valeur dans son calcul.

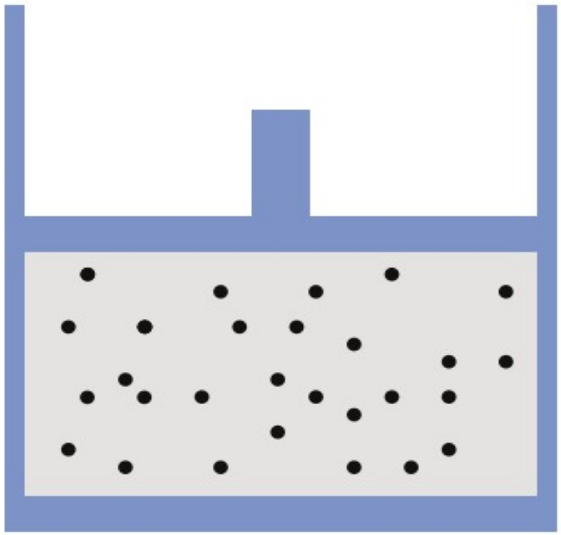


Augmentation de la pression à température constante

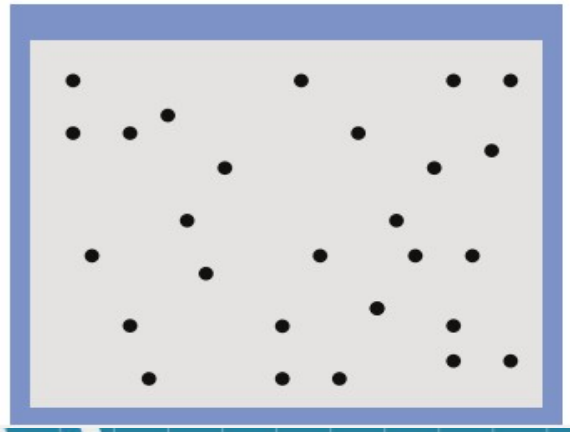


La pression augmente
à température constante.

Le capteur à RI détecte
les molécules de CO_2.

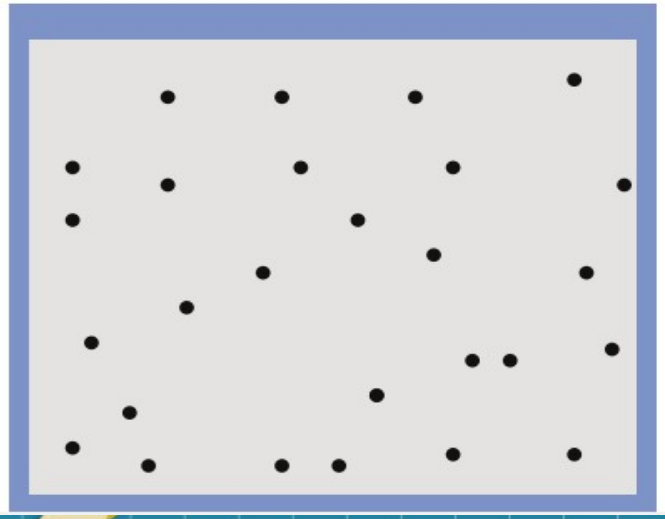


Augmentation de la température à pression constante



La température augmente
à pression constante

Le capteur à RI détecte
moins les molécules
de CO_2.



Il faudrait plutôt employer la formule suivante:

$$\rho(t,p) = \rho(25°C,1013hPa) \times \frac{p}{1013} \times \frac{298}{(273+t)}$$

où :

ρ = concentration du volume de gaz [ppm ou %]
p = pression ambiante [hPa]
t = température ambiante [°C]

Équation 1. Calcul de la concentration de gaz à une température et une pression donnée.

Prix et fournisseur

16€08 livré chez aliexpress

https://fr.aliexpress.com/item/10PCS-LOT-MH-Z19-infrared-co2-sensor-for-co

20€ chez banggood

https://www.banggood.com/fr/MH-Z19-0-5000PPM-Infrared-CO2-Sensor-For-

Les autres sites sont plus chers.

Pour les moins bricoleurs:

36€ tout monté
https://fr.aliexpress.com/item/Free-ship-by-quick-Singapore-post-5pcs-lot-Ra

Version de luxe
https://fr.aliexpress.com/item/Free-shipping-Original-10000ppm-S8-CO2-Sen

Version UART/USB pour PC

Évolution possible : branchement uart sur les pins du capteur puis branchement en usb sur PC avec révision du logiciel.

0,87€ frais de port inclus

Sources :

https://www.vaisala.com/sites/default/files/documents/CEN-TIA-Parameter-How-to-measu

https://www.circuits.dk/testing-mh-z19-ndir-co2-sensor-module/

https://projetsdiy.fr/projet-micropython-publier-taux-co2-capteur-mhz19-domoticz/

https://www.home-assistant.io/components/sensor.mhz19/

https://github.com/open-homeautomation/pmsensor