

```

# récupération_capteurs.py

001| ### lib thermomètre
002| import os
003| import csv
004| import time
005|
006| ### lib débitmètre
007| import RPi.GPIO as GPIO
008| import time, sys
009|
010| ### lib manomètre
011| import Adafruit_ADS1x15
012|
013|
014|
015| ### thermomètre init
016| file=open('/home/pi/Desktop/TIPE/capteurs/valeurs/thermometre_valeur.txt', 'r')
017|
018| ### graph init
019| file2=open('/home/pi/Desktop/TIPE/capteurs/valeurs/valeurs.txt', 'a')
020| file2.truncate(0)
021| file2.close
022|
023| ### débitmètre init
024| # Initialisation des GPIO
025| FLOW_SENSOR_GPIO = 17
026| GPIO.setmode(GPIO.BCM)
027| GPIO.setup(FLOW_SENSOR_GPIO, GPIO.IN, pull_up_down = GPIO.PUD_UP)
028|
029| global count
030| count = 0
031|
032| ### manomètre init
033| adc = Adafruit_ADS1x15.ADS1115(address=0x48, busnum=1)
034|
035| ### valeur init
036| Time = [0]
037| Debit = [0]
038| Pres = [0]
039| Temp = []
040|
041| global time_init
042| time_init = time.time()
043|
044| rang = 0
045|
046| #température t0
047| Temp.append(file.readlines()[-1])
048|
049| #manomètre t0
050| value = adc.read_adc(0, gain=2/3) # Lecture du port A0 de la carte ADC1115 ||
Gain obtenu par lecture du dossier technique
051| volts = value / 32767 * 6.144 # Ratio of 15 bit value to max volts determines
volts
052| #pa = 2500 * (volts - 0.1015) + 0.5 # Formule obtenu lecture du datasheet et
offset (-0.1015) par test pour la calibration || valeur obtenue en hPa
053| bar = (2500 * (volts - 0.1015) + 0.5)/1000 # Formule obtenue lecture du
datasheet et offset (-0.1015) par test pour la calibration || valeur obtenue en bar
054| Pres_init = round(bar*750,3) #passage de bar à mmHg
055|
056| def countPulse(channel):
057|     global time_init
058|     Time.append(time.time()-time_init)
059|
060|     #compte pulse débitmètre
061|     global count

```

```

062|     if start_counter == 1:
063|         count = count+1
064|
065|     #Récup température
066|     file=open('/home/pi/Desktop/TIPE/capteurs/valeurs/thermometre_valeur.txt',
067| 'r')
068|     Temp.append(file.readlines()[-1]) #récupération de la dernière valeur du
thermomètre
069|     file.close()
070|
071|     #Récup manomètre
072|     value = adc.read_adc(0, gain=2/3)
073|     volts = value / 32767 * 6.144
074|     bar = (2500 * (volts - 0.1015) + 0.5)/1000
075|     Pres.append(round((bar*750) - Pres_init,3))
076|     Debit.append(Debit[-1])
077|
078| GPIO.add_event_detect(FLOW_SENSOR_GPIO, GPIO.FALLING, callback=countPulse)
079|
080| while True:
081|     try:
082|         start_counter = 1
083|         time.sleep(1)
084|         start_counter = 0
085|         # Pulse frequency (Hz) = 7.5Q, Q is flow rate in L/min.
086|         flow = count / 7.5
087|         debit = 1.927*flow-2.156 #formule obtenue par régression
088|         if debit<=0:
089|             Debit.append(0)
090|         else:
091|             Debit.append(debit)
092|         count = 0
093|
094|         fin = min(len(Time), len(Pres), len(Temp), len(Debit))
095|
096|         file2=open('/home/pi/Desktop/TIPE/capteurs/valeurs/valeurs.txt', 'a')
097|         #passage au string parce que les floats ne sont pas "subscriptable"
098|         for i in range(rang, fin):
099|             file2.write(f"{str(Time[i])}\t {str(Debit[i])}\t {str(Pres[i])}\t
{str(Temp[i])}")
100|         file2.close()
101|
102|         rang = fin
103|
104|     except KeyboardInterrupt:
105|         print('\nkeyboard interrupt!') #Ctrl + C
106|         GPIO.cleanup()
107|         sys.exit()

```