# -\*- coding: utf-8 -\*-

"""

Created on Thu Feb 1 19:10:17 2018

@author: ferchi

"""

import math as m

from matplotlib import pyplot as plt

import numpy as np

from scipy.optimize import curve\_fit

from array import array

import os

#cargo los txt

tiempo=np.loadtxt('C:/Users/ferchi/.spyder-py3/labo/Mediciones/difusion/TIME2.txt', delimiter = ',')

datos=np.loadtxt('C:/Users/ferchi/.spyder-py3/labo/Mediciones/difusion/DATA2.txt', delimiter = ',')

tiempo2=np.loadtxt('C:/Users/ferchi/.spyder-py3/labo/Mediciones/difusion/tiempos3.txt', delimiter = ',')

datos2=np.loadtxt('C:/Users/ferchi/.spyder-py3/labo/Mediciones/difusion/datos3.txt', delimiter = ',')

tiempo\_fin1=12\*3600+32\*60+42

tiempo\_inicio1=tiempo\_fin1-tiempo[5,-1]

tiempo\_fin2=12\*3600+59\*60+60

tiempo\_inicio2=tiempo\_fin2-tiempo[5,-1]

tiempo2=tiempo2+tiempo\_inicio2-tiempo\_inicio1

#¢junto las matrices de ambas mediciones

tiempo=np.c\_[tiempo, tiempo2]

datos=np.c\_[datos, datos2]

#cada fila de "datos" corresponde a un canal

#limpio los puntos que se alejan bastante del anterior

d=len(datos[0,:])-1

i=0

while i<d:

if abs(datos[1,i]-datos[1,i+1])>1:

datos[1,i+1]=datos[1,i]#reemplazo el punto por el anterior (podia borrar el punto pero era mas lio)

i=i+1

i=0

while i<d:

if abs(datos[2,i]-datos[2,i+1])>1:

datos[2,i+1]=datos[2,i]

i=i+1

i=0

while i<d:

if abs(datos[3,i]-datos[3,i+1])>1:

datos[3,i+1]=datos[3,i]

i=i+1

i=0

while i<d:

if abs(datos[4,i]-datos[4,i+1])>1:

datos[4,i+1]=datos[4,i]

i=i+1

i=0

while i<d:

if abs(datos[5,i]-datos[5,i+1])>1:

datos[5,i+1]=datos[5,i]

i=i+1

i=0

while i<d:

if abs(datos[0,i]-datos[0,i+1])>1:

datos[6,i+1]=datos[6,i]

i=i+1

'''

plt.plot(tiempo[0,:],datos[0,:],'r.', label = 'CH6')

plt.plot(tiempo[1,:],datos[1,:],'b.', label = 'CH5')#por si pinta graficar con puntos

plt.plot(tiempo[2,:],datos[2,:],'g.', label = 'CH4')

plt.plot(tiempo[3,:],datos[3,:],'r.', label = 'CH3')

plt.plot(tiempo[4,:],datos[4,:],'b.', label = 'CH2')

plt.plot(tiempo[5,:],datos[5,:],'g.', label = 'CH1')

'''

#ploteo

plt.plot(tiempo[0,:],datos[0,:],'r-', label = 'CH6')

plt.plot(tiempo[1,:],datos[1,:],'b-', label = 'CH5')

plt.plot(tiempo[2,:],datos[2,:],'g-', label = 'CH4')

plt.plot(tiempo[3,:],datos[3,:],'r-', label = 'CH3')

plt.plot(tiempo[4,:],datos[4,:],'b-', label = 'CH2')

plt.plot(tiempo[5,:],datos[5,:],'g-', label = 'CH1')

plt.grid(True) # Para que quede en hoja cuadriculada

plt.title('Temperatura en funcion del tiempo')

plt.xlabel('tiempo (s)')

plt.ylabel('Temnperatura (ºC)')

plt.legend(loc = 'best')