```
############################
"""
Mide voltaje a travez de el nanovlotimetro agilent
contola y mide la temperatura a travez de Neocera LTC11
utilizamos una fuente de corriente manejada a travez del puerto
parallelo

se grafican los datos en tiempo real y ademas se guardan en un archivo

"""

import visa
import time
import matplotlib.pyplot as plt
import numpy as np
import parallel


class handler():
    """
                Maneja los eventos de matplotlib y el control de
temperatura
    """
    def __init__(self,line,puerto):
        self.line = line# line es un objeto de matplotlib, es sobre
loo que se grafica
        self.puerto = puerto
        self.setP = 0 #set point
        self.setPex = 0 #punto decimal del set point
        self.new = True # habilita la entrada de un nuevo dato
        self.xs = line.get_xdata()
        self.ys = line.get_ydata()
        self.STOP = False # frena la iteracion
        self.pausa = False
        self.cid =
line.figure.canvas.mpl_connect('close_event',self.stop)
        self.cidkey =
line.figure.canvas.mpl_connect('key_press_event',self.setPoint)

    def stop(self,event):
        """
                Cuando se cierra la ventana se llama a este metodo
para que cambie
                el valor de Stop y se frene la iteracion que esta
corriendo siempre
        """
        self.STOP = True

    def cambiarSetP(self):
        """
                Nos conectamos a traves del puerto serie para cambiar
el setpoint
        """
        print('nuevo set point: ', self.setP)
        self.puerto.write('SMON;') #modo monitor (Esta linea no es
estrictamente necesaria)
```

```
        self.puerto.write('SETP1,'+str(self.setP)+';')# setea el
setPoint a 80
        self.puerto.write('SCONT;') # volvemos a modo control
                #self.puerto.write('SCONT;')# modo control

    def setPoint(self, event):
        """
                Maneja los eventos cuando se aprieta una tecla
                Para cambiar el setPoint escribir un numero con el
teclado(acepta punto decimal)
                No se ve en ninguna parte de la pantalla lo que se
escribe, pero igual esta escribiendo
                Apretar enter para mandar el setpoiunt al equipo
                La tecla borrar borra todo lo que se escribio para
empezar a escribir devuelta
        """
        try:
            n = int(event.key)
            if self.new:
                self.setP = n
                self.setPex = 0
                self.new = False
            elif self.setPex==0 :
                self.setP = self.setP*10 + n
            else:
                self.setP = self.setP + n*(10**self.setPex)
                self.setPex -= 1
        except:
            if event.key == '.':
                self.setPex = -1
            if event.key == 'enter':
                self.new = True
                self.cambiarSetP()
            if event.key == 'backspace':
                self.new = True


class fuente():
        """
        Maneja la fuente de corriente a traves del puerto paralelo
        """
        def __init__(self):
                self.p = parallel.Parallel()  # open LPT1 or
/dev/parport0
                self.curr = 0 # 0-> corriente positiva; 1-> corriente
negativa
                self.medicion = 0 # 0-> tension; 1-> termocupla
                self.p.setData(2*self.medicion+self.curr)
        def set(self):
                self.p.setData(2*self.medicion+self.curr)
        def invertir(self):
                if(self.curr == 1):
                        self.curr = 0
                else:
                        self.curr =1
        def medir(self,entrada):
                if entrada=='t':
```

```
                        self.medicion = 1
                elif entrada=='v':
                        self.medicion = 0



rm = visa.ResourceManager()
rm.list_resources()

tc = rm.open_resource('GPIB::10::INSTR')#a traves de tc nos conectamos
con el tc neocera Ltc 11
#revisar en cada caso el nombre correcto para el equipo
nanovolt = rm.open_resource('GPIB::22::INSTR')# este hay que
habilitarlo desde el equipo

print(nanovolt.query('*IDN?;'))
print(tc.query('*IDN?;'))

 # posibles comandos que quieran utilizar
#tc.write('SMON;')# modo monitor
#tc.write('STUNE1;') # rampa, va desde temperatura actual a 80.0 K a
1.0 K/min

#tc.write('SETP1,10;') # setea el setPoint a 10
#tc.write('SRMP1,1,102.0,5.;') # rampa, va desde temperatura actual a
80.0 K a 1.0 K/min

#tc.write('SCONT;') # modo control

#print(tc.query('QSETP?1;'))# pregunta el setpoint

p = parallel.Parallel()  # open LPT1 or /dev/parport0

p.setData(0x00)
#envia el byte en hexa con los puestos 2-9
# 0xff prende todos
# 0x00 apaga todos
# dan voltaje de 0.5 cuando estan apagados y 3.4 prendidos


file = open('mediciones.txt','a')


t0 = time.time()
t = t0


fig = plt.figure()
#GRafico de temperatura
axTc = fig.add_subplot(311)
axTc.set_xlabel('Tiempo (s)')
axTc.set_ylabel('Temperatura (K)')
lineTc, = axTc.plot([],[])

#Grafico de tension
axNv = fig.add_subplot(312)
```

```
axNv.set_xlabel('Tiempo (s)')
axNv.set_ylabel('Tension 1 (V)')
lineNv, = axNv.plot([],[])

#Grafico de tension
axNt = fig.add_subplot(313)
axNt.set_xlabel('Tiempo (s)')
axNt.set_ylabel('Tension 2 (V)')
lineNt, = axNt.plot([],[])



H = handler(lineTc,tc)



fuente = fuente()
#nanovolt.write("CONF:VOLT:DC 0.001,1E-9") # este comando permite
variar algunas configuraciones del voltimetro


T1 = 0
T2 = 0
file.write("tiempo temperatura V1 V2 T1 T2\n")

while not H.STOP:
    T_aux = tc.query('QSAMP?1;')
    V1 = float(nanovolt.query('READ?;'))
    fuente.invertir()
    fuente.set()
    time.sleep(0.2)
    V2 = float(nanovolt.query('READ?;'))

    t = time.time()

    fuente.medir('t')
    fuente.set()
    time.sleep(0.2)
    T1 = float(nanovolt.query('READ?;'))
    fuente.invertir()
    fuente.set()
    time.sleep(0.2)
    T2 = float(nanovolt.query('READ?;'))

    fuente.medir('v')
    fuente.set()

    T = float(T_aux[0:len(T_aux)-3])


    file.write(str(t-t0)+ " " +str(T)+ " " + str(V1)+ " " + str(V2)+ "
" + str(T1)+ " " + str(T2)+"\n")
```

```
    axTc.plot(t-t0,T,"r*")
    axNv.plot(t-t0,(V1-V2)/2,"b.")
    axNt.plot(t-t0, (T1-T2)/2,"g.")
    plt.pause(0.1) # muestra los datos y espera 0.1 segundos

file.close()
```