```
#!/usr/bin/env  python
# Sample Python/Pygame Programs
# Simpson College Computer Science
# http://programarcadegames.com/
# http://simpson.edu/computer-science/
# modif pour trraceur courbe temps reel 17/6/2013
# utilisable aussi  pour visu enregistreur Nokia,chargeur batterie
#
#
 # resion 18/6/2013 menu in separate window
 # revision 25/6/2013 different color for different curves
 # revision 2/7/2013 divers
 #revision 4/7/2013 bug corrected in graphe_tout (y_to_plot)
 # revision 5/7  scale lin or log for y
 # revision 7/7 double scale for log for low value
 # graph10_5.py 7/7/2013 :make option of loading a file to analyse not yet
working
 # graph_10_7 8/7/2013 :loading file and visu working some bugs to remove
 # graph_10_9 bugs removed, window size adjusable
 # graph_10_12 16 juillet 2013:rs232 flag put in place to solve some
problem when downloading
# graph 11_test_3  4aout 2013:adjsutable backgroud color
# graph_12_test_1  22/8/2013 option to download and lot data from gas
flowmeter
# graph_13.py : 22/8/2013 ok version debuggee
# graph15.py :elimine bug sur echelle log
#graph16 21/12/2013 sauve auto ors dechargement recorder
#-----------------------------------------------------------------------
# ----------------------------general description :
# show data coming from RS232 port having the following sructure:
#  *,x,y1,y2,y3,....CR/LF
# values of y1, y2,y3... are ploted against x
# the * caracter marque the start of data , each field is comma separated
from the others
# in case of downloading from a portable instrument the end of download is
marqued by # which
# close the serial port
#
# Case of gas flowmeter : data structure coming from recorder: *,t1 CR LF
where t1 is time between 2
# decompression in second .Before the first valid record . the
decompression volume is send as  v,50
# (unit for v  is Ncm3/10)
# example of valid file :
#      bla bla bla CR LF
#      v,50
#      *,12 CR LF
#      *,14 CR LF
#-----------------------------------------------------------------
import pygame
import serial
import math
import tkFileDialog as Selector
yes=1
no=0
# ----- default value ----------------------
largeur_win=600
hauteur_win=300
baud_rate=4800  # download speed
v=50   # volume par decompression cas lecture gas flowmeter
echelle_lin=yes  # 1 for linear scale 0 for log scale
```

```
x0=0   # valeur x mini
xm=1000  # val x maxi
y0=1   # val y mini
ym=1000 # val y maxi
nx=10   # nombre graduation sur axe x
ny=10  # nombre graduation sur Y
end_of_download='#'   # caracter used in the file send o rs232 to mark end
of data
y_to_plot=['1']  # liste contenant les param y a tracer 1,3,5 par defaut 1
por y1
data_=[]   # liste qui stocke x,y1,y2,...
record_number=0  # number of record read
number_of_parameter=0  # number of parameter
color=[0]  # list containing color of curves
# Define the colors we will use in RGB format
black = [  0,  0,  0]
grey=[125,125,125]
light_grey=[200,200,200]
white = [255,255,255]
blue =  [  0,  0,255]
green = [  0,255,  0]
red =   [255,  0,  0]

background_graph=light_grey  # background of graph can be change by the
menu
color=[black,red,blue,green]  # 4 colors used to draw differents curves
rs232_open=False
dummy=0
affiche=' '
# ------------------------------------------
# command code for vt102 tty terminal
reset_origine=chr(27)+"[?6l"
cursor_home=chr(27)+"[H"
reverse_video=chr(27)+'[7m' # video inverse
under_line=chr(27)+'[4m'  # souligne
bold=chr(27)+"[1m"  # ecriture en gras
reset_graph_mode=chr(27)+"[m"  # revient au regalge affichage par defaut
reset=chr(27)+'c'   # efface ecran  curseur en haut et revient au reglage
par default

#-------------------------------------------------
def open_serial():
         global ser,rs232_open
         ser=serial.Serial('/dev/ttyUSB0',baud_rate,timeout=1)
         #ser=serial.Serial(0)
         rs232_open=True
         return


#-----------------------------------------------------
def graphe(y_a_tracer,curve_color):
        global x0,xm,y0,ym,val
        global echelle_lin
        global nx,ny,data_,number_of_parameter


        xori=0.06*largeur_win
        xabci=0.9*largeur_win
        yori=0.9*hauteur_win
        yordo=0.03*hauteur_win
        yabci=yori
```

```
        xordo=xori
        x=0
        y=0
        ximpri=0
        yimpri=0
        dx=xm-x0
        if echelle_lin==no:
                y0=y0+1
                dy=math.log(ym)-math.log(y0)
        else:
                dy=ym-y0
        # trace axes
        pygame.draw.line(screen,black,[xori,yori],[xabci,yabci],3)
        pygame.draw.line(screen,black,[xori,yori],[xordo,yordo],3)

        font = pygame.font.Font(None, 20)
        # -----------------trace reticule--------------------------
        y=y0
        pas_y=(ym-y0)/ny # pas pour tracer le reticule suivant Y
        while y<=ym+1:
        #for y in range(y0,ym+1,dy/ny):
            if echelle_lin==yes:
                yimpri=yori-(y-y0)*(yori-yordo)/dy

pygame.draw.line(screen,light_grey,[xori,yimpri],[xabci,yimpri],1)
                text = font.render(str(y),True,black)
                y=y+pas_y
            else:
                yimpri=yori-(math.log(y)-math.log(y0))*(yori-yordo)/dy

pygame.draw.line(screen,light_grey,[xori,yimpri],[xabci,yimpri],1)
                text = font.render(str(y),True,black)
                if y<y0+pas_y:
                   y=y+pas_y/ny
                else:
                   y=y+pas_y

            screen.blit(text, [xori-30,yimpri])


        for x in range(x0,xm+1,dx/nx):
            ximpri=xori+(x-x0)*((xabci-xori)/dx)

pygame.draw.line(screen,light_grey,[ximpri,yori],[ximpri,yordo],1)
            text = font.render(str(x),True,black)
            screen.blit(text, [ximpri,yori+10])



        n_param=number_of_parameter   # len(val)
        n_record=len(data_)/n_param
        #print data_
        #y_a_tracer=1  # pour test preliminaire
        #print "nparam=",n_param,"  nrecord=",n_record
        first=1
        for i in range(0,len(data_),n_param):
                try:
                        x=int(data_[i])   # sort x et y du tableau data_
                        y=int(data_[i+y_a_tracer])
                        #print "i= ",i,"  x=",x,"  y=",y,"x+y",x+y
```

```
                        ximpri=xori+(x-x0)*(xabci-xori)/dx
                        if echelle_lin==yes:
                                yimpri=(y-y0)*(yori-yordo)/dy
                        else:
                                yimpri=(math.log(y)-math.log(y0))*(yori-
yordo)/dy
                        yimpri=yori-yimpri
                        if first==1:

        pygame.draw.rect(screen,curve_color,[ximpri,yimpri,2,2],2)
                                xold=ximpri
                                yold=yimpri
                                first=0

                        else:

        pygame.draw.line(screen,curve_color,[xold,yold],[ximpri,yimpri],2)
                                xold=ximpri
                                yold=yimpri
                except:
                        print "erreur fichier"
                        pass
        return
#---------------------------------------------------------------------
def graphe_tout():
        global y_to_plot,color
        #print "valeurs en memoire"
        jj=0
        for i in range(0,len(y_to_plot)):
                #print "valeurs i",i," y_to_plot ",y_to_plot[i],"y_to_plot
",y_to_plot
                y_a_tracer=int(y_to_plot[i])
                couleur=color[jj]
                graphe(y_a_tracer,couleur)
                jj=jj+1
                if jj>3:#  4 valeurs pour la couler des courbes
                        jj=0
        #dummy=raw_input( "return pour suite")

#---------------------------------------------------------------------------
def get_command():
        global x0,xm,y0,ym
        global baud_rate
        global y_to_plot,number_of_parameter
        global data_,record_number,affiche
        global largeur_win,hauteur_win
        global background_graph

        print reverse_video
        print"---------Available commands ------"
        print reset_graph_mode
        print " 0   : change download baud rate  (",baud_rate,")"
        print " 1   : change Xmin i  Xmaxi"
        print " 2   : change Ymini  Y maxi"
        print " 3   : define data_Y to plot vs X"
        print " 4   : change x y graduations "
        print " 5   : save data read"
        print " 6   : load a file of data "
        print " 7   : change Y scale type (lin or log)"
        print " 8   : change graph background color"
        print " 9   : change window size "
```

```
print
print " 10  : start real time downloading "
print " 11  : start recorder download "
print " 12  : start gas_flowmeter download"
print reverse_video,
dummy=raw_input("Your choice please ? ")
print reset_graph_mode

if dummy=='0':
        baud_rate=int(raw_input("New baud rate "))

        return 0
if dummy=='1':
        x0=int(raw_input("X mini "))
        xm=int(raw_input("X maxi "))
        #try:
        update_graph()
        #except:
        #       print "erreur"
        #       pass

        return 1


if dummy=='2':
        y0=int(raw_input("Y mini "))
        ym=int(raw_input("Y maxi "))
        try:
                update_graph()
        except:
                print "erreur"
                pass
        return 2

if dummy=='3':
        global y_to_plot
        print "enter index of various Y to plot   eg 1+3+4"
        s=raw_input("  ")
        y_to_plot=s.split('+')
        print y_to_plot
        try:
                update_graph()
        except:
                print "erreur"
                pass
        return 3

if dummy=='4':
        global nx,ny
        print" nombre. grad. sur x et y actuelles :",nx,"/",ny
        nx=int(raw_input("Nouvelle val. de nx "))
        ny=int(raw_input("Nouvelle val. de ny "))
        try:
                update_graph()
        except:
                print "erreur"
                pass
        return 4

if dummy=='5':
```

```
                descr=Selector.asksaveasfilename(title='Fichier contenant
les donnee',defaultextension=" *.dat")
                print "nom fichier ",descr
                print "number of record:",record_number
                print "number of parameters ",number_of_parameter
                #print data_[0],",",data_[1],",",data_[2],",",data_[3]
                #zz=raw_input(" suite enter ")
                ii=0
                try:
                        file=open(descr,'w+')
                        while ii<len(data_):
                                jj=0
                                while jj<number_of_parameter:
                                        print >>file,(data_[ii+jj]+","),

                                        jj=jj+1
                                print >> file

                                ii=ii+number_of_parameter
                        file.close()
                        print "download is done "
                except:
                        print "erreur"
                        dummy=raw_input("return ")
                        pass
                return 5
        if dummy=='6':

                data_=[]
                descr=Selector.askopenfilename(defaultextension=".dat")
                print "nom fichier ",descr
                file_descri=descr
                # file strucure :   x,y1,y2,y3,     example for 3 Y as of X,
                try:
                        file=open(descr,'r')
                        record_number=0
                        for lu in file:
                                print lu,
                                lu=lu.strip()   #strip RC et NL
                                lu=lu.strip(',')# strip , terminal
                                #print "lu apres strip",lu
                                val=lu.split(",")
                                #print "apres lu.split",val
                                #number_of_parameter=len(val)

                                for item in val:
                                        data_.append(item)
                                record_number=record_number+1
                                number_of_parameter=number_of_parameter+1
                        number_of_parameter=len(val)
                        print "nombre de paramametre",len(val)
                        print " number of record :",record_number
                        file_descri=file_descri+"/ Parameters
:"+str(number_of_parameter)
                        file_descri=file_descri+"/
Records="+str(record_number)
                        affiche=file_descri  # to show on graph
                        file.close()
                        prepare_graph()
                        update_graph()
                        #get_command()
```

```
                except:
                        print "erreur"
                        pass
                return 6
        if dummy=='7':
                global echelle_lin
                echelle_lin=int(raw_input("choix type echelle :1= lineaire
0 =log  "))
                try:
                        update_graph()
                        pygame.display.flip()  # test alas
                except:
                        print "erreur"
                        pass
                return  7
        if dummy=='8':
                print "enter the backgroud color for graph
present=",background_graph
                xx=int(raw_input("value between 0 (black) and 255 (white)"))
                background_graph=[xx,xx,xx]
                update_graph()
                print "ok"
                return 8
        if dummy=='10':
                # start downloading
                data_=[]

                return 10
        if dummy=='9':
                print "present window widh ",largeur_win
                largeur_win=int(raw_input(" enter new width "))
                print " present window height ",hauteur_win
                hauteur_win=int(raw_input(" Enter new height "))
                try:
                        prepare_graph()
                        update_graph()
                except:
                        print "erreur"
                        pass
                return  9


        if dummy=='11':
                total_download_rs232()

                save_data()
                #affiche='Parameter:'+ 'Records:'
                #prepare_graph()
                #affiche='Parameters:'+str(number_of_parameter)+'   Number
Records:'+str(record_number)
                #pygame.display.set_caption(affiche)
                #graphe_tout()
                #pygame.display.flip()
                #et_command()
                return 11
        if dummy=='12':
                total_download_rs232_flowmeter()

                prepare_graph()
                affiche='Parameters:'+str(number_of_parameter)+'   Number
Records:'+str(record_number)
```

```
                pygame.display.set_caption(affiche)
                graphe_tout()
                pygame.display.flip()
                return 12
        return -1


def save_data():
                descr=Selector.asksaveasfilename(title='Fichier contenant
les donnee',defaultextension=" *.dat")
                print "nom fichier ",descr
                print "number of record:",record_number
                print "number of parameters ",number_of_parameter
                #print data_[0],",",data_[1],",",data_[2],",",data_[3]
                #zz=raw_input(" suite enter ")
                ii=0
                try:
                        file=open(descr,'w+')
                        while ii<len(data_):
                                jj=0
                                while jj<number_of_parameter:
                                        print >>file,(data_[ii+jj]+","),

                                        jj=jj+1
                                print >> file

                                ii=ii+number_of_parameter
                        file.close()
                        print "download is done "
                except:
                        print "erreur"
                        dummy=raw_input("return ")
                        pass


#-----------------------------------------------------------------
def update_graph():
        screen.fill(background_graph)
        font = pygame.font.Font(None, 25)
        text = font.render(affiche,True,black)
        # Put the image of the text on the screen at 60*3
        screen.blit(text, [60,1])
        graphe_tout()
        pygame.display.flip()
        return

# -----------------------------------------------------------------

#-----------------------------------------------------------------------
def download_one_record_rs232_flowmeter():
         # load data_[] with the various parameters
    global data_,record_number,val,number_of_parameter,affiche
    global done
    global total_time,total_volume
    global v
    # structure  *,t1 ended with CR,NL
    global ser



    c='a'
```

```
    while c!='*':  # attente  carac *
           c=ser.read()
           print c,
           if c=='v':  # detecte valeur volume decompression
                               print"v detecte"
                               lu=ser.readline()
                               lu=lu.strip(',')
                               v=int(lu)
           if c==end_of_download:
                               print" End of data "
                               ser.close()


                               return -1


    lu=ser.readline()

    #print "lu ser.readline", lu
    lu=lu.strip()   #strip RC et NL
    lu=lu.strip(',')
    #print "lu apres strip",lu
    val=lu.split(",")
    #print "apres val=lu.split",val
    x=int(val[0])
    #print "int(val)=", x

    total_time=total_time+x
    debit=(3600*v/x)/10  # debit en Ncm3/h
    total_volume=total_volume+v/10 # volume total en Ncm3
    number_of_parameter=3
    record_number=record_number+1
    print record_number,":",
    print "Temps:",total_time,"(sec)  Debit:",debit," (Ncm3/h)  Volu
degaze:",total_volume," (Ncm3)"
    data_.append(total_time)
    data_.append(debit)
    data_.append(total_volume)
    #print data_
    #variable_names=('temps(sec)=',' debit=',' volume')
    return record_number
#-----------------------------------------------
def total_download_rs232_flowmeter():
        global total_time,total_volume
        total_time=0
        total_volume=0
        dummy=0
        open_serial()
        while download_one_record_rs232_flowmeter()!=-1:
                dummy=dummy
        ser.close()
        #prepare_graph()
        #graphe_tout()
        #pygame.display.flip()

        return
```

```
#-----------------------------------------------------------------
def total_download_rs232():
        dummy=0
        open_serial()
        while download_one_record_rs232()!=-1:
                dummy=dummy
        ser.close()
        #prepare_graph()
        #graphe_tout()
        #pygame.display.flip()

        return
#-----------------------------------------------------------------
def download_one_record_rs232():
         # load data_[] with the various parameters
    global data_,record_number,val,number_of_parameter,affiche
    global done
    # structure  *,x1,y1,y2,.....ended with CR,NL
    global ser
    c='a'
    while c!='*':  # attente  carac *
           c=ser.read()
           print c,
           if c==end_of_download:
                           print" End of data "
                           ser.close()


                           return -1


    lu=ser.readline()

    #print "lu ser.readline", lu
    lu=lu.strip()   #strip RC et NL
    lu=lu.strip(',')
    #print "lu apres strip",lu
    val=lu.split(",")
    #print "apres lu.split",val
    number_of_parameter=len(val)
    #print "nombre de paramametre",len(val)
    record_number=record_number+1
    print record_number,":",
    #print val
    ii=0
    variable_names=('X=',' y1=',' y2=','  y3=',' y4=','  y5=')

    for item in val:
               print variable_names[ii],val[ii],
               data_.append(item)
               ii=ii+1



    print ""


    return record_number

#-----------------------------------------------------------------
def listen_rs232():
```

```
# return 0 if nothing read or number_of_parameter
# load data_[] with the various parameters
global data_,record_number,val,number_of_parameter,affiche
global done
# structure  *,x1,y1,y2,.....ended with CR,NL
if ser.isOpen()==False:
           return 0
if ser.inWaiting()==0:
           return 0

c='a'
while ser.inWaiting()>=0 and c!='*':  # attente  carac *
       c=ser.read()
       print c,
       if c==end_of_download:
                           print" End of data "
                           ser.close()
                           rs232_open=False
                           get_command()
                           return 0


lu=ser.readline()

#print "lu ser.readline", lu
lu=lu.strip()   #strip RC et NL
lu=lu.strip(',')
#print "lu apres strip",lu
val=lu.split(",")
#print "apres lu.split",val
number_of_parameter=len(val)
#print "nombre de paramametre",len(val)
record_number=record_number+1
print record_number,":",
#print val
ii=0
variable_names=('X=',' y1=',' y2=','  y3=',' y4=','  y5=')

for item in val:
           print variable_names[ii],val[ii],
           data_.append(item)
           ii=ii+1



print ""

#print data_x

# Select the font to use. Default font, 25 pt size.
font = pygame.font.Font(None, 25)

# Render the text. "True" means anti-aliased text.
# Black is the color. This creates an image of the
# letters, but does not put it on the screen
affiche=' ' # 'X='+val[0]+'  y1='+val[1]
ii=0
for item in val:
           affiche=affiche+variable_names[ii]+val[ii]
           ii=ii+1
```

```
    #affiche='X='+val[0]+'  y1='+val[1]
    text = font.render(affiche,True,black)
     # Put the image of the text on the screen at 60*1
    screen.blit(text, [60,1])
    return number_of_parameter

#------------------------------------------------------------
def prepare_graph():
        global screen ,largeur_win,hauteur_win
        pygame.init()
        size = [largeur_win,hauteur_win]
        screen = pygame.display.set_mode(size)
        screen.fill(background_graph)

#----------------------PROGRAMME PRINCIPAL--------------------------------
--------

encore=True
while encore==True:
        c=get_command()
        #print "val de c =",c
        if c==10:
                encore=False
# --- zone de telechargement -------------
prepare_graph()
pygame.display.set_caption("Type Esc key for menu ")
open_serial()
#Loop until the user clicks the close button.
done = False
clock = pygame.time.Clock()

while done == False:

    # This limits the while loop to a max of 10 times per second.
    # Leave this out and we will use all CPU we can.
    #clock.tick(1)

    for event in pygame.event.get(): # User did something
        if event.type == pygame.QUIT: # If user clicked close
            done=True # Flag that we are done so we exit this loop
        if event.type== pygame.KEYDOWN and event.key == pygame.K_ESCAPE:
#escape K
            #ser.flush()
            #ser.close()
            get_command()
            open_serial()
            ser.flush()


    # Clear the screen and set the screen background

    screen.fill(background_graph)

    if rs232_open==True:
        if listen_rs232()>0:
            graphe_tout()
            pygame.display.flip()

# Be IDLE friendly
pygame.quit()
ser.close()
```