```
# -*- coding: utf-8 -*-
"""
Created on Fri Nov 20 11:39:23 2020

@author: benw2
"""
import microspec as usp
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import tkinter as tk
# Turn on LED
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(18, GPIO.OUT)
GPIO.output(18, True)

#Read file for getting wavlgth map
df=pd.read_csv("1308-36_map_noheader.txt",sep='\t',index_col=False)
pix=df.iloc[:,0].tolist()
df=pd.read_csv("1308-36_map_noheader.txt",sep='\t')
wvlgths=df.iloc[:,0].tolist()

#open window and put header in
window = tk.Tk()
greeting = tk.Label(text="Chromation Spectrometer GUI")
greeting.pack()

kit = usp.Devkit() #class, opens up serial comm
kit.setAutoExposeConfig(start_pixel=min(pix),stop_pixel=max(pix),max_exposure=10000
)
kit.serial.close()

#Open Serial
def s_open():
    kit.serial.open() #class, opens up serial comm
    kit_open= tk.Label(text="Serial Open")
    kit_open.pack()

btn_s_open=tk.Button(text="Open Serial",command=s_open)
btn_s_open.pack()

#Exposure Button
def exps():
    #set exposure/autoexpose
    kit.autoExposure()
    num_exps=kit.getExposure()
    example=tk.Label(text=num_exps)
    example.pack()
```

```
btn_exps=tk.Button(text="Auto Expose",command=exps)
btn_exps.pack()

#Number of averages entry window
na_lab = tk.Label(text="Enter Number of Averages")
na_lab.pack()
entry_avgs=tk.Entry(width=10)
entry_avgs.pack()


#Start button and def
def start():
    started = tk.Label(text="Started")
    started.pack()
    #start plot
    num_avg=range(int(entry_avgs.get())) #seto array up to num_avg from text box
    #

    #while loop to get data and plot
    global run
    run=1
    global avg
    #while run==1:
    avg_counts=[]
    for x in num_avg:
        reply=kit.captureFrame()
        counts = ([reply.frame[pixel] for pixel in reply.frame if min(pix)<=pixel
<=max(pix)])
        avg_counts.append(counts)

    avg_counts=np.array(avg_counts)
    avg=np.mean(avg_counts,axis=0)
    plt.plot(wvlgths,avg)
    plt.xlabel("Wavelength (nm)")
    plt.ylabel("Counts")
    plt.show()


btn_start=tk.Button(text="Start",command=start)
btn_start.pack()

#Close Serial
def s_close():
    kit.serial.close()
    kit_close= tk.Label(text="Serial close")
    kit_close.pack()

btn_s_close=tk.Button(text="Close Serial",command=s_close)
btn_s_close.pack()
```

```
OUTPUT = tk.Label(text="OUTPUT")
OUTPUT.pack()

window.title("Chromation GUI")
window.mainloop()
GPIO.output(18, False)
```