

ET4244

Outcome Based Learning Laboratory 2

Autumn Semester 2020

Report

Name: Wasim G Aswad

Student ID: 17193559

The BSc in Mobile Communications and Security

Group Number: 4

Report Submission Date: 18/12/2020

Abstract

In this Project, the data which have used is collected from sensors gyroscope and accelerometer. We processed the data in usable form and plot the data to get useful insights from the data through different angles.

The dataset is a simple text file which is unprocessed and unknown to a python script or graphs. So first the data is processed and converted into a tabular form and stored in the csv file. What the application do is take the data, convert the data into csv file, store the status of the file in system logs and also write logs on a separate file and store that file in your system directory with the name of log-file and finally plot that cleaned data as per the user choice like which dimension of particular sensor needed to plot.

In this report the first part will contain the table of content for the navigation path throughout the project, list of figures which will depict the graphical interface of the application, list of equations which will contain the whole script used in this project and the list of tables which will displays the module in this project and their functionality. In the second part of the report the work will be discussed on each topic in detail and each topic will be named as chapter.

And in each chapter the tasks will be discussed in detail.

Table of Contents

[Chapter 1: Introduction: 1](#_Toc58448012)

[Chapter 2: Graphical Interface 3](#_Toc58448013)

[Chapter 3: Importing & Pre-processing Data 7](#_Toc58448014)

[Chapter 4: Select & Plot the Data 11](#_Toc58448015)

[Chapter 5: Software used in the project 15](#_Toc58448016)

[Chapter 6: Conclusions 17](#_Toc58448016)

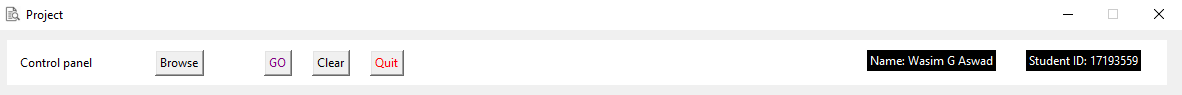
List of Figures

Figure 1

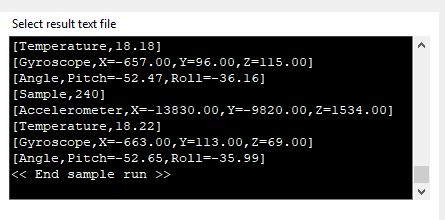


Figure 2

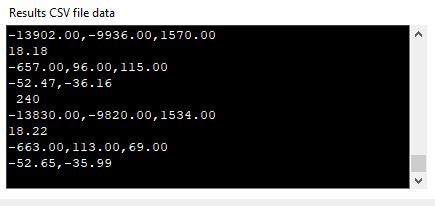


Figure 3

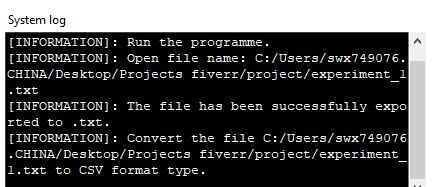


Figure 4



Figure 5

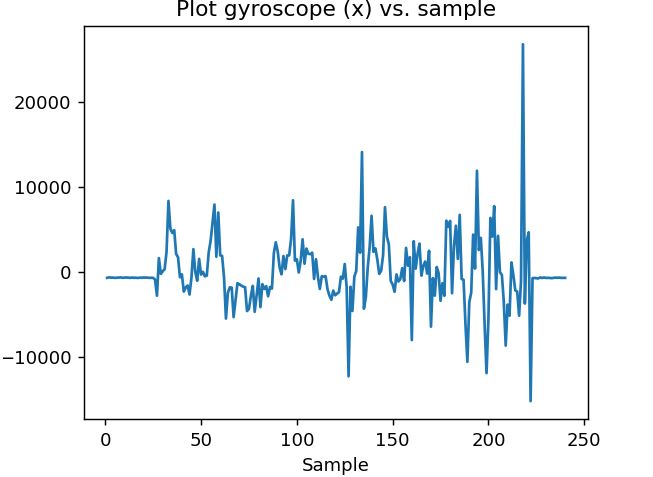


Figure 6

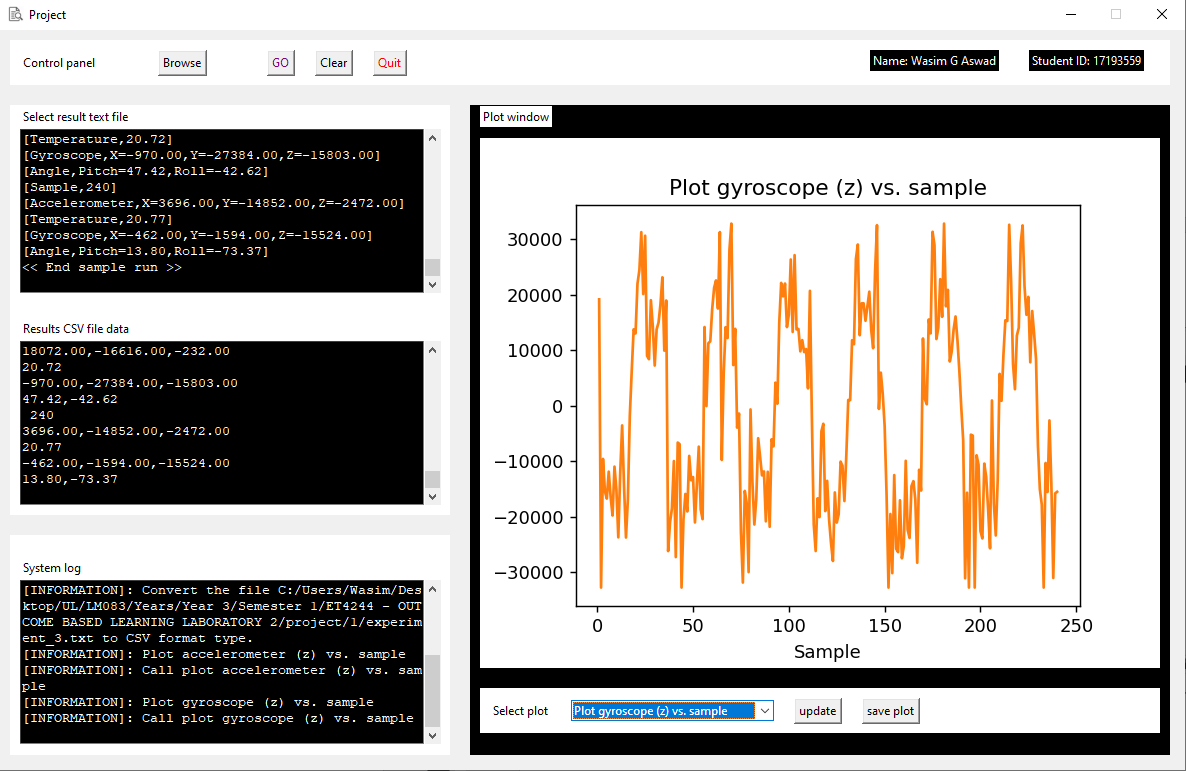


Figure 7

*Figure 2*

List of Tables

|  |  |
| --- | --- |
| Major Functions | Module Used |
| Graphical Interface Design of the Application is designed through a built-in library of Python. | Tkinter and PYQT both are Python module for desktop application development Tkinter is good for simple and small application, That’s why Interface design is achieved through Tkinter. |
| Graphical visualization of the Data is also achieved through a python library. | Matplotlib is a python library for the visulation of data using different charts and graphs. In this report Matplotlib library has been used for the visualization of the data. |
| Read comma separated dataset. | CSV is also python package for importing and reading csv files. And in this app csv is used to import and read the data in a file. |
| os is used to get current working directory so that the location of the dataset can fetch. | OS |
| Numpy is used for fast mathematical calculation. It decrease the processing time. | numpy |
| The [shutil](https://docs.python.org/3/library/shutil.html#module-shutil) module offers a number of high-level operations on files and collections of files. In particular, functions are provided which support file copying and removal. And in application it is used to copy a file. | shutil |
| Python Imaging Library is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats.  In this app it’s used for opening an image and resizing it. | PIL |
| Time is a python package used for time related queries. In our case it’ll be used in storing system logs. | time |
| The sys module provides functions and variables used to manipulate different parts of the Python runtime environment. | sys |
| The doctest module searches for pieces of text that look like interactive Python sessions, and then executes those sessions to verify that they work exactly as shown. | doctest |

## Chapter 1: Introduction:

The application will follow a series of steps while processing. Each step is a separate function; some functions will be called sequentially but the others will be called based on the decision of user. The basic work flow is as follows:

1. First the ***main()*** function will be called. This function is the root of this application program. Compilation will be started from this function.



Figure 3.1

1. This main function will load the graphical interface of the application, it will display the title of the application on the top of the application program using ***title()*** function, It’ll describes the width and height of the interface window using ***geometry()*** function. And the ***resizable()*** function is used to make check on either user can change the size of interface or it’s fixed. In this case it’s fixed because the value of width and height is **False**. We can make it flexible by setting the values of both width and height **True.**
2. Graphical interface will contain multiple actions at the behind of each button.



Figure 4

1. Next function will be called with respect to the button chosen by the user.
2. ***Browse*** button will be used to import the dataset which is in raw text file. And now the data is in processed form we can plot the data
3. ***Go*** button will be used to convert that data file into a processed csv file.
4. ***Save log*** will be used to save the log of the file.
5. ***Clear*** button will clear the screen, imported file, system logs and plotted data if we have already plotted.
6. As Its name shows ***Quit*** button is used to exit the application. It will call the exit script at the backend.

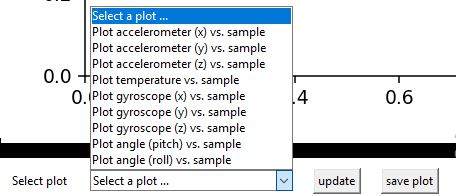


Figure 5

1. For plotting the dataset, a list is available, the desired option will be selected from the list.
2. Update button is used to plot the data in graphical portion & also update the plot portion if there is already plotted data in graph portion.
3. Save plot is used to save the graph in image from.
4. Behind each button there is an action, each button calls a script from implementation.

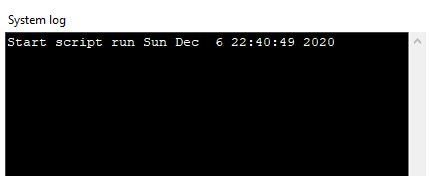


Figure 6

1. System log portion displays the log of the files, it shows dataset names, location and time. On the right upper corner there is also a close button by default which is also used to close the application program.

## Chapter 2: Graphical Interface

In previous chapter, at the last line in *figure 9* ***create\_widgets()***is used to call***create\_widgets()***function. Which displays the graphical interface through which user can interact with the system. **Tkinter** library is used for the graphical interface of the application. This function used different **widgets** imported from tkinter library.

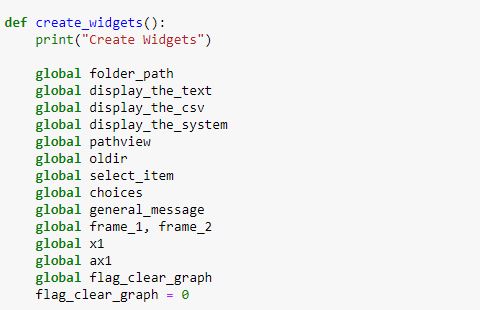


Figure 7

All the variables declared with **global** keyword so that the scope of the variables will remain throughout the execution of the program. Destructor function is used to destroy the variables in the memory. In this case the destructor will be called only with the exit of the application and these variables can be accessed anywhere in the program.



Figure 8

Frames are used for logical division of the page. And one frame can also be divided into multiple sub-frames and each sub-frames can be further divided into rows and columns. Frames are created used ***Frame()*** function which is built-in functionality, ***width*, *height*** and ***background*** are the parameters of the functions which we can set as per our requirements. In *Figure 13* frame1 is created and it is further divided into two sub-frames, ***frame1*** is root frame but ***smallframe1***& ***smallframe2***are derived from *frame1*.

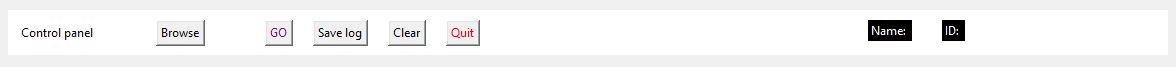


Figure 9

Above mentioned is the full frame, the left side which contains all the buttons is sub-frame2 and the right side which contains ***Name & ID*** is sub-frame one in *Figure 14*. The other frames for the interface is created as *Figure 13.*

After that these frames are converted into rows and columns as per requirements. And we can also set other parameters like padding etc.

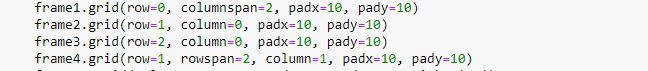


Figure 10

The ***grid()*** function is used for this purpose, *row & column* is used for further logical division and padding is inserted through *padx & pady* parameters. Padx is the padding from x axis and pady is used to insert padding from y axis.

The ***grid\_propagate()*** function is used to actually deploy these grids on the interface. After that Labels, Buttons, checkboxes and graph is drawn with respect to the frames and sub-frames.

Figure 11

***Label()*** is used for drawing a label and Button is used to displaying button on the interface. The text and bg are parameters of Label and smallframe2 is the location of frame where the label will be drawn.

In ***Button()*** command is used for calling the function/Action behind that particular button. In *Figure 16* the *browse\_button* function will be called when the browse will press. Then the ***grid()*** function is used for displaying the label or button and also decide in which particular column or row the button will be displayed and their padding as well. Padding is achieved using two parameters *padx (*padding from x\_axis)& *pady (padding from y\_axis)*.

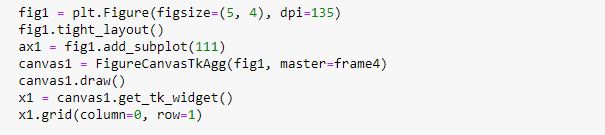


Figure 12

This code will display the graph in the interface at frame4. It’ll display the plot based on selected data set. The selection is provided as a drop down list using the tkinter’s function ***ComboBox().*** And finally bind the selection from the list with the graph which will plot the selected dataset.

## Chapter 3: Importing & Pre-processing Data

The first task is to import the dataset. The browse button is used to fetch the data which is selected by the user of the application program. Browse button is used to call the ***browse\_button()*** function.



Figure 13

Filename is declared as global. Operating system’s function ***getcwd()***is used to get the location of current working directory and stored in a variable which is used to fetch the dataset***. Open()*** function is used to open the file and the parameter ***‘r’*** means open the file in read mode. ***File.readlines()*** is used to read the lines of file and all lines will be stored in lines variable. For loop is used to loop through every single line of the file and then close then file.

The dataset which is imported here is in .txt file. We cannot plot the data in this format, so that’s why another method will be used for the conversion of text file into comma separated file.



Figure 14

Go button on the interface trigger this ***go\_()*** function. This function will call another function export\_to\_csv() which is used to convert the text file into comma separated file. This go function will display the name of the file on the interface in system log option using insert function. Tk is the abbreviation of tkinter module.



Figure 15

The export\_to\_csv() function first get the file path then check whether at this path the file exist or not. In case of not it’ll raise exception but if the file is present at the location it will fetch the path and name of the file and store it into a variable and then use it for later conversion into csv.

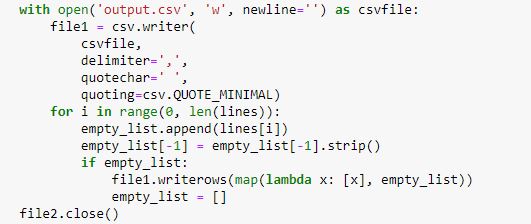


Figure 16

Open function is used to open the file in read and write mode. It opens the file if exist and if file doesn’t exit it then create a file. In this scenario it will create a file with the name of ouput.csv in a write mode. Write mode mean we can write data on the file and read it as well. The delimiter is ‘,’ because the csv files are comma separated files. For loop will read every line of the text file make a separation by commas in every line and then append the new line after the existing line in a sequence. All of these line will be actually written on output.csv file. And at the end the file will be closed using ***close()*** function.

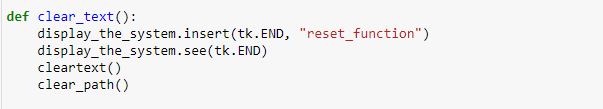


Figure 17

Clear\_text() function is used to clear all the logs and previous processing. In-short It’ll reset the whole application program. It will clear all path which stored in the variable and all the text in the variables.

## Chapter 4: Select & Plot the Data

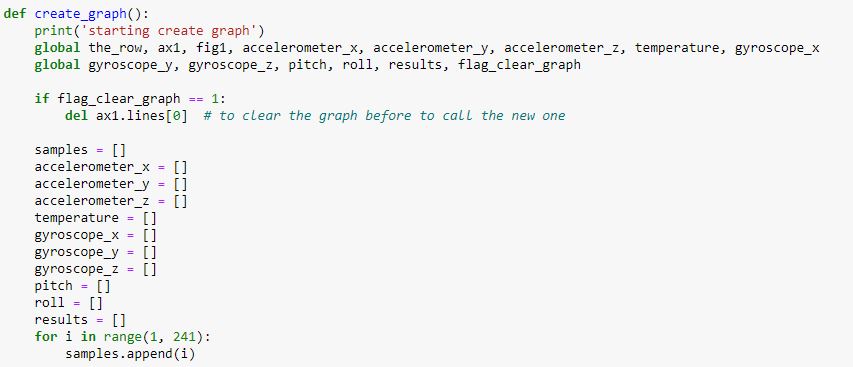
The final step is plotting the processed data. The create\_graph() function will be called for the creation of graphs based on your selected axis.

Figure 18

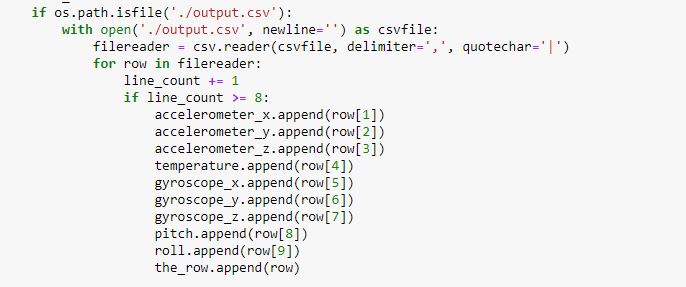
The dataset contains the data of two sensors. Accelerometer data is consisting of three dimensions and same goes for gyroscope, temperature, pitch and roll has atomic column. Sample will be the range of data, it’s like a counter on an axis. In this case the total length of data is 240 so the range in this loop will be from 1 to 241. The sample data will consist of a list of values from 1 to 240 and It’ll loop through all 240 rows of data. In *Figure 23* the empty lists are declared in which the data will be stored from the imported csv file.

Figure 19

In *Figure 24* at first the processed csv file is imported and then in a loop all the rows are storing in required list, each time the loop runs it appends the data in existing list.

Now the data is ready for plot.

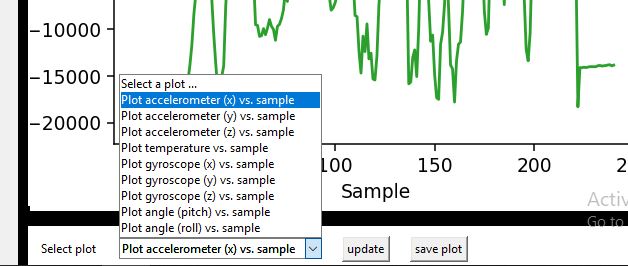


Figure 20

And now the data will be plotted based on the selection of the columns from the list and update button will be used to display the data in graphical format. And save plot option is used to save the plotted data in images form. It’ll through an exception in case if there is no processed dataset (csv file) found.

Gyroscope has the data of there all dimension each named with x, y & z. And same goes for accelerometer. We can choose the one which we wanted to plot vs sample. The graph will display the line graph. Sample will be on x axis always and the data will be on y axis.

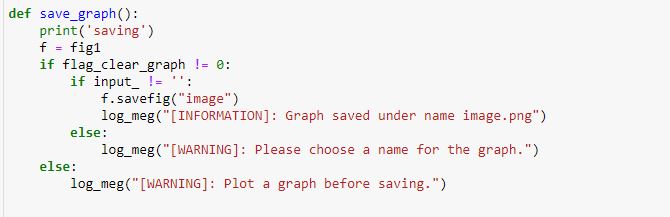


Figure 21

In *Figure 26*, this script is used to save the graph in an image format in a directory. The graph will be automatically saved with the name of *image.png* but if there is not data has plotted and you press the save graph button then the exception will be raised the *plot the graph before saving*. It will save the most recent graph that we have plotted.

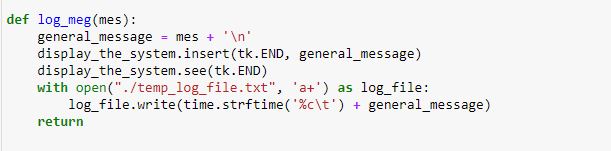


Figure 22

This script is used to store the log related to the state of data on an external file. The file is a text file with a name temp\_log\_file.txt. And *write()* function is used to write the log on that file mentioned in the *open() function.* When *with* keyword is used to open a function it’s mean that the file will be automatically closed after the operation has performed on it. On the other case when we simply use *open()*, the file will be close manually by calling close function.

## Chapter 5: Software used in the project

**Python language**  
Python is a high-level programming language that is considered an object-oriented language and it's a language that attention to whitespace. After using it, can see it is easier than other languages of the same level such as Java or C. Since, it uses terms that are very close to the English language more than other programming languages, which reduces the cost of maintaining the program. Python supports modules and packages, which encourages code reuse and program modularity [1]. In general, Python can be used to create programs for beginners and to complete huge projects at the same time. Often pass a law in the English language. It is often recommended for beginners in the field of programming to learn this language because it is among the fastest-learning programming languages [2].

**PyCharm software**

PyCharm is the most popular IDE used in the Python scripting language. PyCharm offers some of the best features to its users and developers in the following aspects:

1. Write code and check for errors.
2. An advanced debugging proposal.
3. Support for web programming and frameworks like Django and Flask.
4. PyCharm also offers suggestions for code completion more seamlessly, whether it's built-in or for an external package [3].
5. Free program.

## Chapter 6: Conclusions

The project is quite amazing, it collects the raw data, which is meaning less and then make that data worthy and convert it into a meaningful form. After that Analyse the data through different angles, use different parameters to visualize the data. Save the Processing steps of data in the form of logs and save the graphs in pictorial format.

If I attempt this project again I focus on the betterment of the efficiency of code, I tried to make the code shorter using built-in tools and designed in more efficient way so that it takes very less time to execute, I focused on some advance graphical representation for more clarity.

As a personal reflection, I was working on improving the program so that when the user wanted to save an image of the graph, a pop-up window would appear asking the user to enter a name to save the image in it. Accordingly, there was no enough time to help me to do that, so I made it save the pictures with the default name, which is (*image*).

The same thing I can say about saving the log, At the moment the program is based to asks the users when they want to exit from the program if they want to save the log window, and if the users Choose Yes, the log would be saved under the (*log\_file*) name. So the plan was to find a way to allow the users to choose the location and name of the file.

One of the things that didn't work for me was trying to add a splash window. when I add the window it was working but the problem is the programme was working at the same time. that meaning the user can see two windows one is the splash window and behind it have the program.

References

[1] "What is Python? Executive Summary", *Python.org*, 2020. [Online]. Available: https://www.python.org/doc/essays/blurb/. [Accessed: 17- Dec- 2020]

[2]"أنسام الاغا يكتب: 'لغة بايثون (Python) ....' | رقيم", *رقيم*, 2020. [Online]. Available: https://www.rqiim.com/ansamalagha46259/%D9%84%D8%BA%D8%A9-%D8%A8%D8%A7%D9%8A%D8%AB%D9%88%D9%86-python. [Accessed: 17- Dec- 2020]

[3] Tutorialspoint.com. 2020. *Pycharm - Introduction - Tutorialspoint*. [online] Available at: <https://www.tutorialspoint.com/pycharm/pycharm\_introduction.htm> [Accessed 17 December 2020].