Data Log Sorter and Plotter Application

Co-op: Govind Luck

[goviluck@gmail.com](mailto:goviluck@gmail.com)

Manager: Dhruva Acharyya

11/15/2023

Table of Contents

[Purpose 3](#_Toc153546814)

[Architecture and Libraries 3](#_Toc153546815)

[Main code 3](#_Toc153546816)

[Data Log Sorter Class 4](#_Toc153546817)

[Initialization 4](#_Toc153546818)

[Column Adder Methods 4](#_Toc153546819)

[Sort Rest Methods 5](#_Toc153546820)

[Save File 5](#_Toc153546821)

[Unit Updater 5](#_Toc153546822)

[Calculate Error Bar 5](#_Toc153546823)

[Calculate MV FV Data 5](#_Toc153546824)

[Range Pin Divider 6](#_Toc153546825)

[To Excel Sheets 6](#_Toc153546826)

[Collect and Clear Ticks 6](#_Toc153546827)

[Create Plot Data 6](#_Toc153546828)

[Format Save 7](#_Toc153546829)

[Bow Tie Plotter 7](#_Toc153546830)

[Rich Function 7](#_Toc153546831)

[Data Log Sorter Application 7](#_Toc153546832)

[Initialization 7](#_Toc153546833)

[Create Label Entry and Button 8](#_Toc153546834)

[Create Main GUI 8](#_Toc153546835)

[Plotter GUI 8](#_Toc153546836)

[Show Plots GUI 8](#_Toc153546837)

[Sorter GUI 8](#_Toc153546838)

[Conclusion 8](#_Toc153546839)

# Purpose

The purpose of this user application is to ease the sorting of csv files and plotting data. Often users will have a data log that they want to sort by voltage or amp values within the test name. Using excel this task is difficult and extremely time consuming. This application does the sorting process in seconds. By inputting a common text across test names that come either before or after the value to be sorted the value is appended as a column to the csv file. More columns can be added such as the pin number. The order these columns are added to should be in descending importance. Finally, the user can click the button to sort these rows and save the csv to a location.

The other option of this application is to plot force voltage data vs force and measure error data in either volts or least significant bits. This part of the application is specific to VI100 Verify/Force Voltage Measure Voltage tests which contain 3 different types of data, DMM measure, ADC measure, and ADC error measure.

# Architecture and Libraries

Thie python files uses two classes to perform all operations. The first class is a data log sorter class which takes in a csv file to be sorted and has various methods that sort or plot the csv file in python. The second class is the data log sorter app which creates and maintains the GUI.

This project uses many APIS to perform various functions.

* Pandas store csv cells in data frames and perform operations with ease.
* matplotlib.pyplot to plot all relevant data.
* NumPy for math operation.
* Regular expressions for string parsing.
* csv for importing and exporting.
* os for creating folders.
* seaborn for creating more aesthetic plots.
* mpl cursors to view specific data points.
* time to record how long these processes take.
* pickle to save the python figures for future use within the application.
* Tkinter to create a GUI.
* Messagebox from tkinter to display completion messages to the user.
* Filedialog for browsing user folders and files.

# Main code

In order to use this application pandas data frames are used to sort and organize the data and matplotlob.pyplot is used to show the data in a step plot. The user interface is created using the tkinter API.

A screen shot of a computer

Description automatically generated

The main code first ensures that this is the main program being run and if true creates a variable root which root window for the application. It then creates a variable app, which is a data log sorter application class using this main window. Following this the root protocol is set for when the main window is closed. When this occurs, it causes all other windows to be destroyed and the main loop exists. The main loop method simply creates an event loop which detects events and controls the GUI.

# Data Log Sorter Class

Before initializing the object some class constant variables are created that will be used later such as the desired confidence interval, least significant bit mapping, and an empty map for gridline ticks.

## Initialization

When an object of type “data\_log\_sorter” is created, it requires a file name and save folder string input. The file name is parsed to add .csv to the end if needed. The save folder string is also parsed to add a “\” to the end of the name if not added already.

Next the csv file is opened and parsed row by row using csv.reader. For each row the code looks for "Number\_of\_Tests", “Number\_of\_Runs”, and “INDEX”. When the first two are found the code looks at the next value in the csv file to find the number of tests and runs. Finally, when “INDEX” is found the index of the header row is saved to the variable header for creating the pandas data frame.

Next a data frame is created from the csv using the found values and parsing cells after an initial space to ensure there are no errors. The columns of this dataframe are are renamed without any leading or any spaces. Next if any columns contain no data they are removed from the dataframe. Following this any unnamed columns are added to a list and renamed in order according to how many runs there are.

## Column Adder Methods

The first function “force\_voltage\_col\_adder” adds a column of force voltages to the data frame using regular expressions. First a pattern is created which will match all text that starts with “Force” then contains a number and is followed by “V”. The number in the pattern is surrounded by parenthesis which creates a group that is extracted from the matched string. Next a list is created that contains the extracted number for each row. Finally, this list is added to the data frame with column header “Force Vol”.

This process is repeated for pin number, range, and input variables with different matching strings. Sometimes the number cannot be found in the string and thus empty values are filled with 0.0 floating point number.

“var\_col\_adder” has a slightly different pattern variable. This function takes in “var\_sort” which is a text string to be matched to. It also takes in a variable “before\_or\_after” which changes the pattern to be created. If the variable is equal to after it will search for a number before the inputted text. The opposite of this occurs for before.

## Sort Rest Methods

The “sort” methods essentially sort the data frame based give columns and resets the index values so that the new values can be easily accessed.

The first two functions have preset sort columns which are used to sort the data frame so that tests specific to VI100 Verify/Force Voltage are sorted properly for plotting.

The pin and var sort methods instead uses an empty list of sort columns which is created when an object of this type is made. When either of these functions are called, they check if the column header exists in the list (the user inputted text for “var\_sort”), appends it if not and then sorts the data frame by the list of columns. This is important for using the GUI. Creating this list ensures that the order the user decides to create columns in determines the level of importance for sorting.

## Save File

This method creates a save location string using the given save folder upon creation of the class and a new file name which is inputted when one uses the interface. It checks if the file name ends in .csv and adds if needed. It then saves the data frame to a csv file and prints a message to the user.

## Unit Updater

This method first created a unit map with unit prefixes as the keys and their corresponding multipliers as values. It then extracts the unit prefix from the unit column of the data log. Using this prefix each run measurement data is converted to its base power value. Finally, the Unit column is removed from the data frame as it is no longer needed and inaccurate after the conversion.

## Calculate Error Bar

This function calculated the size of the plotted error bars. It takes in a list of values and calculates the error bar by finding the standard deviation, dividing it by the square root of the number of values and multiplying by the constant confidence interval.

## Calculate MV FV Data

This method performs the bulk of the calculations in code. First column headers are created which will contain specific data for each test. More headers are appended to the list for MV, FV, MV error, and FV error for each run.

A for loop is now used to iterate through the data frame incrementing by 3 so that one new row of data can be created for one test which includes DMM measure, ADC measure, and ADC error measure. First the pin number, force voltage, voltage range, test number, and test name are stored in variables for this iteration. The test name saved used regular expressions to capture just the force voltage, range, and pin number to be the new test name.



Next a 3-row data frame is created that contains each test and runs data.

A screenshot of a computer

Description automatically generated

Using this dataframe the means of each row are used to fin the avg measure, force, measure error and force error voltage. The measure voltage is just the average of row 3, the force voltage is already calculated earlier, the avg MV error is the average of row 2 minus row 1, and the FV error is row 1 minus the force voltage. The error bards for MV and FV error are calculated using the calculate\_error\_bar method with each as its input.

Now the data is added to a list which is the added to the data list. This creates a list of lists meaning the added list is a new row of the data frame. After the loop is complete, a new data frame is created using the column headers and new data values.

## Range Pin Divider

This mothod creates a list of data frames for specific ranges and pin numbers. It first creates a list of unique range values by iterating though the new data frame “Range” column. It does the same iterating through the “Pin Number” column. Next an empty list is initialized to store the data frames. Following this the range values list and pin list are iterated through so that if the current range value and pin number are found in the data frame that row is appended to a new data frame. This is done for all combinations of ranges and pins. Each data frame is appended to the list of data frames.

## To Excel Sheets

This method first creates a new excel file in the given save folder. Each data frame in the data frame list is added to a separate sheet in the excel file and labeled appropriately.

## Collect and Clear Ticks

The collect tick’s method takes in a data frame as a variable. It first appends the FV avg, max and min values to its respective lists of the dictionary. It then does the same for the max and min FV and MV error avg to their respective lists of the dictionary.

The clear ticks simply takes in a string of either “FV” or ”MV” and clear the max and min values for FV error if the string is “FV” and clears the all other values in the dictionary if the variable is “MV”

## Create Plot Data

This function creates plot data for each pin number and range. It takes in a data frame, a “FV” or “MV” string, and axis variables. First the pin number of the data frame is saved in a variable. Next the class instance variable “volts\_or\_lsb” is used to enter one of two if statements.

If the variable is equal to “VOLTS” the data frame volt values are for either “FV” or “MV” are updated to be micro volts for better plotting. Next a plot is created with respective labels on the given axis.

If the variable is equal to “LSB” the range value of the given data frame is saved in a variable. This variable is used as a key to access the least significant bit mapping and convert the respective voltage measurements to LSB using constant multipliers. Next a plot is created with respective labels on the given axis.

## Format Save

This method takes in a FV and MV string, range name, axis, and figure variables. It first gets the max and min values of FV and either FV error or MV error. It then creates an appropriate title for the given axis and sets the x axis grid lines to be 11 ticks between the min and max values inclusive. Next it used the class instance variable volts or LSB to create y axis grid lines rounding differently depending on the units. It also creates axis grid lines to be 11 ticks between the min and max values inclusive. After a line a y = 0 is created to easily compare error data to.

Finally, the method checks if there is folder in the save folder where plots are stored. If not a folder is created. In this folder the plot is saved as a .png with an appropriate title and the clear ticks method is called with the input MV or FV. It also save the file as a .pickle file so that if desired the file can be opened again in the GUI even if it was deleted from memory.

## Bow Tie Plotter

This method takes in a string of either “VOTLS” or “LSB”. It first saves this input to a class instance variable. It also creates empty lists of FV and MV axes and figures. The figures lists are class instance variables so they can be accessed later and displayed.

The previously created data frame list is then iterated through using enumerate so that we also record the index of the data frame. First a list index variable is created that is the set using the floor division operator. The current index is divided by the number of pins and rounded down. This ensures the first data frames contain all the pins before a new data frame is accessed.

Using the index, we first check if the remainder of the index plus one divided by the number of pins is equal to one. If this is true it means that all the pins for a range have been plotted and a new graph should be made for a different range. New FV and MV figures and axes are created and added to the lists. Next the create plot data method is called for both FV and MV but the first FV plot is skipped since there is no 1.25V range for this test. Next, we check if the remainder of the index plus one divided by the number of pins is equal to zero. If true it means all the data has been plotted and we can save the figures using format save. Like before the first 1.25V plot is skipped for FV measurements.

Finally, a print statement is used to tell the user where the plots are saved. This print method is called the cyan text method which is used to highlight the save folder.

## Rich Function

This function calls all necessary functions for the plots to be created and saved correctly.

# Data Log Sorter Application

This class creates a general user interface to easily input data log files and perform various operations. This class has three class variables which set the foreground, background, and font of all the labels in the GUI.

## Initialization

When an instance of the data log sorter app is created many class instance variables are created. The class takes in a root input which is saved in the class for creating windows. The variable data is set to type None. Then the entry inputs of the GUI are set to default string values. Next the main GUI method is called which created the main GUI.

## Create Label Entry and Button

These methods take in the current window, text, and other variables to create the GUI. These methods were created to make the code more readable. Using these methods, the GUI can be edited later to include more features or make modifications in a clear and concise way.

## Create Main GUI

This method sets the background, title, and location, of the first Gui that appears. It then creates two buttons which when clicked open either the plotter or sorter GUI.

## Plotter GUI

This method creates another GUI window for plotting. It asks the user to enter a file name and save folder. There is an option to browse the user’s computer for files and folders. It also has two buttons with options to plot the data with a y axis in volts or in LSB. When this button is clicked, and the plots are created another GUI window will open by calling the show plots GUI method.

## Show Plots GUI

This method creates a GUI to allow the user to show plots. It allows the user to click on buttons to open any of the created graphs or all at once. It used a labels dictionary to iterate through range values. Each button is created so that the show figure method is called when the button is pressed.

This method takes in the row index and column index. Using this index, the pickle file is accessed and opened using the pickle API. The plot is then shown with cursors. The show all method has a similar functionality but instead shows all the plots at once. When the created window is closed all plots are destroyed and the data sorter class object is deleted.

## Sorter GUI

This method created the data log sorter GUI. It asks the user for a file named and save folder with the same browse functionality as the previous GUI. It also asks the user for a text string to search for in the test name column. Using this string the user has the option to find the number that comes before or after the string and create a column in the csv file. The user also has the option to create a column of pin numbers using various methods.

The user is then asked for a new file name and can save the new file. This file is also sorted based on the buttons clicked and text entries. The order of the button clicks determines the importance of sorting by with the first button being clicked the most important. The columns to be sorted is displayed from most to least important on the GUI using the update sort columns and reset methods.

# Conclusion

The goal of this project was to create an easier process for plotting and sorting data to save hours of time which could be spent instead analyzing this data. This goal was met with the addition of including an extensive GUI which can be used by anyone.

In the future some improvements can be made to make the code easy to manipulate if changes are needed. The purpose of this documentation is to explain all functions in the code so if errors occur the user can look in the correct locations.