

**Detecting and Characterizing Voltage Sags and Swells
Using RMS Variation Calculations**

by

Erin Nicole Griffith

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
December 13, 2014

Keywords: Voltage disturbance, RMS detection method

Copyright 2014 by Erin Nicole Griffith

Approved by

S. Mark Halpin, Chair, Professor, Electrical and Computer Engineering
R. Mark Nelms, Professor and Chair, Electrical and Computer Engineering
Charles A. Gross, Professor Emeritus, Electrical and Computer Engineering

Abstract

Power quality is of increasing interest in the field of power engineering. Analyzing variations in power quality is essential to determining the causes of fluctuations in the power system. Sifting through data to find disturbances among thousands of instantaneous voltage readings is an extremely time-consuming process. The program developed for this work automates the process of isolating disturbances and provides metrics related to voltage sags and swells. These results can then be examined to determine which events require further inspection. The program also provides a plotting function, allowing the user to view waveforms for further analysis.

Acknowledgments

I would like to thank Dr. Halpin and Dr. Nelms for their guidance in my graduate studies. They have helped me to realize what I am capable of achieving. To Dr. Gross, Dodie, Dr. Roppel, Tammy, and Sara, I wish to express my gratitude for being true friends. Finally, I would like to thank my mother, for being the embodiment of *unconditional* love.

Table of Contents

Abstract	ii
Acknowledgments	iii
List of Tables	vi
List of Figures	vii
Chapter 1: Introduction	1
1.1 Causes of Voltage Sags and Swells	1
1.2 Effects of Voltage Sags and Swells	1
Chapter 2: Disturbance Detection Using RMS Voltage	3
Chapter 3: Visual Basic [®] for Applications and Microsoft Excel [®]	6
Chapter 4: Program Implementation	8
4.1 Changing Program Defaults	8
4.2 Importing the Data File(s)	10
4.3 RMS Calculations	11
4.4 Comparing Nominal and Single-Cycle RMS Values	12
4.5 Disturbance Summary Report	12
Chapter 5: Disturbance Plotting	14
5.1 User Interface for Plotting	14
5.2 Plotting Program	16

Chapter 6: Other Program Features	19
6.1 Viewing Imported Data Files	19
6.2 Clearing the Workbook of Previous Results	20
6.3 The “Hidden” Worksheet	20
6.4 Workbook Protection	21
Chapter 7: Results	22
7.1 Testing	22
7.2 Program Limitations	23
Chapter 8: Conclusion	25
References	26
Appendix A: Program Flow Charts	27
Appendix B: VBA Code	31

List of Tables

Table 1. Categories and Typical Characteristics of Power System Electromagnetic Phenomena	4
Table 2. Disturbance Summary Report	13
Table 3. Data File Characteristics.....	23

List of Figures

Figure 1. “Home” Screen User Interface	9
Figure 2. Optional User Inputs Dialogue Box	10
Figure 3. Select Data File(s) Dialogue Box	11
Figure 4. User Interface for Plotting Parameters	15
Figure 5. Plot Duration Not Defined Error Message	16
Figure 6. Time Domain Plot of Voltage and Current	17
Figure 7. RMS Values Plot of Voltage	18
Figure 8. User Interface for Viewing Raw Data Files	19

Chapter 1

Introduction

The types of equipment connected to the electrical grid have changed dramatically in recent decades [1]. Not only are these new devices susceptible to variations in power quality, but they can also be the cause of fluctuations. Thus, the interest in power quality has increased in recent years. The ability to classify the different types of disturbances is an indispensable step in determining the cause of a disturbance, with the goal of mitigating or eliminating its effects on equipment tied to the system [2]. The focus of this work is automating the process of isolating and characterizing two such disturbances: voltage sags and voltage swells.

1.1 Causes of Voltage Sags and Swells

Faults are a very common cause of voltage sags and can cause voltage swells on unfaulted phases. Their origins are unpredictable and can include lightning, tree limbs coming into contact with conductors, small animals entering equipment, ice loading, and countless others [3]. Other causes for sags are more predictable. The starting of large motors (or other large loads) or switching a capacitor bank off can result in the voltage at some points in the system dropping below nominal equipment ratings [2, 4]. Turning off a large load or switching a capacitor bank on can lead to voltage swells [2].

1.2 Effects of Voltage Sags and Swells

The consequences of these phenomena can be as simple as the momentary flicker of a light bulb [1]. While inconvenient, the cost associated with monitoring and

mitigation may not be justified by the public's desire to alleviate such effects.

Overvoltage conditions can degrade insulation, decreasing the lifespan of expensive equipment such as motors, transformers, PTs, CTs, cable, and switchgear. Electronics can be shut down or damaged by a variety of power quality issues [2]. This is of special concern for industrial plants, which rely heavily on computers and programmable logic controllers that are highly susceptible to disturbances in supply voltage [4]. If the electronics in manufacturing facilities are shut down even momentarily, the risk of revenue loss is high due to production delays (which can be hours for plants with long start-up times) or faulty products which must be discarded. If equipment is damaged, the time and money needed to repair it are also considerations. While disruptions in the power supply can be a source of aggravation to an individual, economic ramifications are the most important reasons to monitor power quality [2].

By establishing screening criteria, the ability to isolate voltage sags and swells is easily realized with a computer program that uses root-mean-square (RMS) calculations. The RMS values can also be used to calculate the duration and magnitude of the sags and swells. With these metrics, a user can determine which events should be studied further. Examining a plot of an event can also provide clues as to its cause and how to prevent it from having any effects on equipment and the end user. The program developed for this thesis imports data files from monitoring devices and scans them for disturbances that meet criteria related to voltage sags and swells. Information about the disturbances found by the program – including location in the data file, duration, magnitude, and categorization (based on RMS magnitudes) – is displayed in a summary report. The program can also plot the waveforms for further examination.

Chapter 2

Disturbance Detection Using RMS Voltage

The process for detecting disturbances can be divided into three phases: data acquisition, disturbance isolation, and disturbance characterization [5]. The equipment used for data acquisition stores six instantaneous readings for each sampling instant, one for each phase of both current and voltage. If a disturbance that meets some pre-defined criteria occurs, the device stores the current and voltage readings during the disturbance for later analysis (along with some data both before and after the event). Thus, all data files contain a disturbance somewhere in the file, but it must be further isolated and evaluated once the data has been downloaded.

Several techniques have been devised for detecting voltage disturbances. The calculation and comparison of RMS voltages are used in the computer program developed in this work. This method is best suited for detecting short- and long-duration RMS variations, as defined in IEEE Standard 1159. Such disturbances are characterized by their duration and RMS magnitude, and their most distinctive feature is “the change of the voltage signal’s RMS value during the disturbance” (Table 1) [2, 5]. Another advantage of using RMS voltage for detection is that a disturbance containing several conditions (e.g., one-half cycle undervoltage followed by one cycle of overvoltage) will be reported as one event [4]. Some transient events may be detected by the program, but many may not be identified because the RMS voltage may not change significantly [2, 5]. Additionally, the metrics in the summary report may not be useful for such disturbances.

Table 1. Categories and Typical Characteristics of Power System Electromagnetic Phenomena [2]

Categories	Typical duration	Typical voltage magnitude
1.0 Transients		
1.1 Impulsive		
1.1.1 Nanosecond	< 50 ns	
1.1.2 Microsecond	50 ns – 1 ms	
1.1.3 Millisecond	> 1 ms	
1.2 Oscillatory		
1.2.1 Low frequency	0.3 – 50 ms	0 – 4 pu ^a
1.2.2 Medium frequency	20 μs	0 – 8 pu
1.2.3 High frequency	5 μs	0 – 4 pu
2.0 Short-duration root-mean-square (rms) variations		
2.1 Instantaneous		
2.1.1 Sag	0.5 – 30 cycles	0.1 – 0.9 pu
2.1.2 Swell	0.5 – 30 cycles	1.1 – 1.8 pu
2.2 Momentary		
2.2.1 Interruption	0.5 cycles – 3 s	< 0.1 pu
2.2.2 Sag	30 cycles – 3 s	0.1 – 0.9 pu
2.2.3 Swell	30 cycles – 3 s	1.1 – 1.4 pu
2.3 Temporary		
2.3.1 Interruption	> 3 s – 1 min	< 0.1 pu
2.3.2 Sag	> 3 s – 1 min	0.1 – 0.9 pu
2.3.3 Swell	> 3 s – 1 min	1.1 – 1.2 pu
3.0 Long duration rms variations		
3.1 Interruption, sustained	> 1 min	0.0 pu
3.2 Undervoltages	> 1 min	0.8 – 0.9 pu
3.3 Overvoltages	> 1 min	1.1 – 1.2 pu
3.4 Current overload	> 1 min	
4.0 Imbalance		
4.1 Voltage	steady state	0.5 – 2%
4.2 Current	steady state	1.0 – 30%
5.0 Waveform distortion		
5.1 DC offset	steady state	0 – 0.1%
5.2 Harmonics	steady state	0 – 20%
5.3 Interharmonics	steady state	0 – 2%
5.4 Notching	steady state	
5.5 Noise	steady state	0 – 1%
6.0 Voltage fluctuations	intermittent	0.1 – 7% 0.2 – 2 P _{st} ^b
7.0 Power frequency variations	< 10 s	± 0.10 Hz

^a The quantity *pu* refers to *per unit*, which is dimensionless. The quantity 1.0 pu corresponds to 100%. The nominal condition is often considered to be 1.0 pu. In this table, the nominal peak value is used as the base for transients and the nominal rms value is used as the base for rms variations.

^b Flicker severity index P_{st} as defined in IEC 61000-4-15:2003 and IEEE Std 1453TM-2004.

The RMS value for each phase (V_{RMS}) is calculated by the program for the j^{th} cycle as shown in (1) [5].

$$V_{RMS}(j) = \sqrt{\frac{1}{N} \sum_{k=jN}^{(j+1)N-1} v^2(k)} \quad (1)$$

j = period number (integer)

N = number of samples per cycle (integer)

v = value of voltage or current at sample k

The window of N samples is then shifted forward in time by one-half cycle and the RMS value is again calculated for one cycle. Shifting the window one-half cycle at a time provides “a convenient measure of the magnitude evolution” [2, 4].

An acceptable voltage range is calculated based on the nominal voltage and specified tolerances. If a single-cycle RMS voltage value does not fall within the acceptable range, it is categorized as part of a disturbance. The next RMS value, the samples for which were taken after a one-half cycle time shift, is then compared to the acceptable voltage range. Once an RMS value within tolerances is found, the event is deemed to be over.

The duration of the event (t_{cyc}) is calculated by determining the number of cycles between the cycle in which the event begins (row k_1 of the file) and the cycle in which the voltage is again within acceptable limits (row k_2 of the file).

$$t_{cyc} = \frac{\Delta k}{N} \quad (2)$$

Because the RMS values are calculated every half cycle, t_{cyc} can only be calculated to the nearest half-cycle. The duration of an event is calculated for each phase of the voltage.

Chapter 3

Visual Basic[®] for Applications and Microsoft Excel[®]

Even with a calculator, the computation of an RMS value for a single cycle of data would be tedious. When multiple calculations are needed, automating the process is essential to ensure accuracy and save time. However, the calculation of RMS values is not the only step in the process of categorizing disturbances that lends itself to automation.

To analyze a file from a data recorder, a user might have to import the file for viewing and formatting. Then, the disturbance would have to be found in the midst of thousands (or tens of thousands) of values which also contain some data for normal system operation. This might be done in a spreadsheet by finding minimum and maximum values or by plotting all data and zooming in on the disturbance. After the disturbance data had been isolated, formulas within the spreadsheet could be used to quantify the event and determine its duration. This would be a lengthy and laborious process, made even worse if multiple files were to be reviewed.

Even without Visual Basic for Applications (VBA), Excel has the ability to perform all the tasks necessary to analyze the data: importing, sorting, calculating, and plotting. With VBA, the process cannot only be automated, but its built-in dialogue boxes and ability to create UserForms (a type of customizable dialogue box containing a myriad of controls to guide the user through a set of selections) can make using a program written in VBA extremely user-friendly [6]. Excel's spreadsheet environment also

provides a way to present results to the user for further examination and analysis. Even the ability to analyze multiples files at once can be handled with a program written in VBA.

With its ability to handle the necessary tasks, perhaps what makes the solution of VBA in Excel most desirable is that it is so readily available to so many users: another expensive program need not be purchased. Additionally, users will likely be familiar with how to use Excel. Even if unacquainted with Excel, VBA and its dialogue boxes can be used to guide most any user familiar with Microsoft Windows[®] through the process of opening, analyzing, and plotting data files.

Chapter 4

Program Implementation

The following are the general steps used to process the data downloaded from the monitoring device. (See also Appendix A.)

1. Change default data file characteristics and disturbance tolerances (optional)
2. Import data file(s)
3. Calculate nominal RMS voltage and current for each phase
4. Calculate single-cycle RMS values, updated every half cycle, for each phase of voltage and current
5. Compare single-cycle RMS values to nominal values with applied tolerances
6. Write disturbance data to summary table
7. Plot disturbance(s) (optional)
8. View data file(s) for disturbance(s) (optional)

When the Excel workbook containing the VBA program is opened, the user can select from six routines accessed by the buttons in the “Home” worksheet (Fig. 1). (See Appendix B for complete program code.)

4.1 Changing Program Defaults

The user must know two characteristics of the data file: the number of rows at the beginning of the file which do not contain voltage and current readings (header rows), and the sampling frequency of the monitoring device. Defaults of three header rows and a sampling frequency of 32 samples/cycle are assumed, but can be changed by the user

with the “Edit Default Data File Characteristics (Optional)” button. Also available for modification are the voltage levels used to classify a signal as a disturbance. The defaults are set to correspond to the definitions for “sag” and “swell” as set forth in IEEE Standard 1159. A sag is said to have an RMS value less than or equal to 0.9 pu, while a swell has a value greater than or equal to 1.1 pu. The duration of both is at least 0.5 cycles [2]. Clicking the button to change the defaults produces four successive input boxes. The input box for header rows is shown in Figure 2.

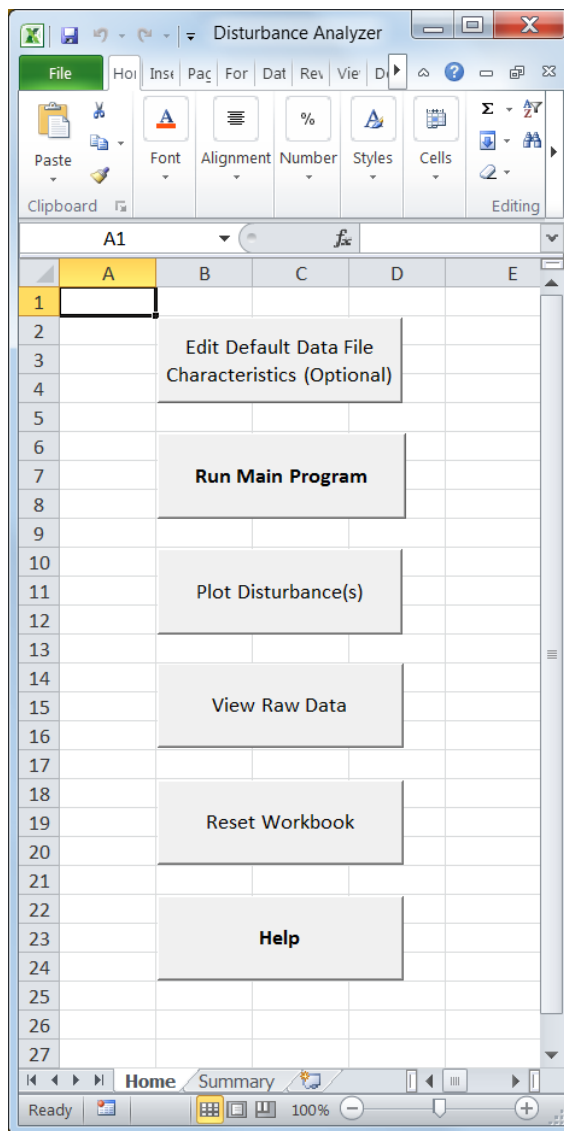


Figure 1. “Home” Screen User Interface

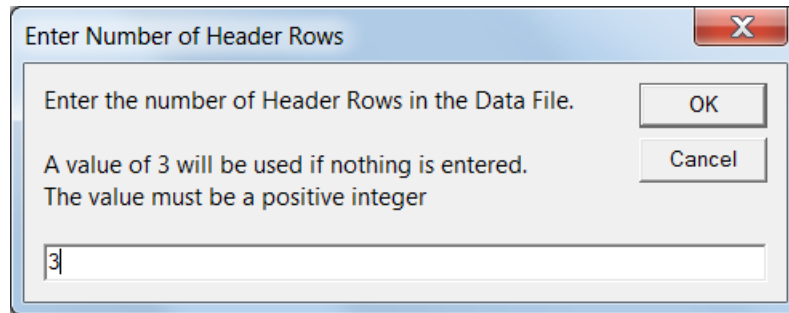


Figure 2. Optional User Inputs Dialogue Box

If any values entered are not usable by the program (e.g., the user enters a letter or negative number), a dialogue box will appear to let the user know what type of information is required.

4.2 Importing the Data File(s)

The program is designed to handle data files that may come from other monitoring devices, as long as the data files have the following characteristics.

- The data is in six columns, with the following order: V_a , V_b , V_c , I_a , I_b , I_c .
- The sampling frequency is an even number of samples per cycle.
- With the exception of header rows, all other rows in the file are data samples.
- The disturbance does not occur in the first six cycles of data.
- The data is in a text file with a “.txt” extension.

Clicking the “Run Main Program” button prompts the user to select data files with a dialogue box just like those used for opening files in other programs run in Microsoft Windows (Fig. 3). Any number of files can be selected. A new worksheet is created in which the voltage and current data is stored, and the file path for that data file is written to the first row in the worksheet. Each worksheet is named “RawData” followed by a number. (The data files are numbered sequentially as they are opened.)

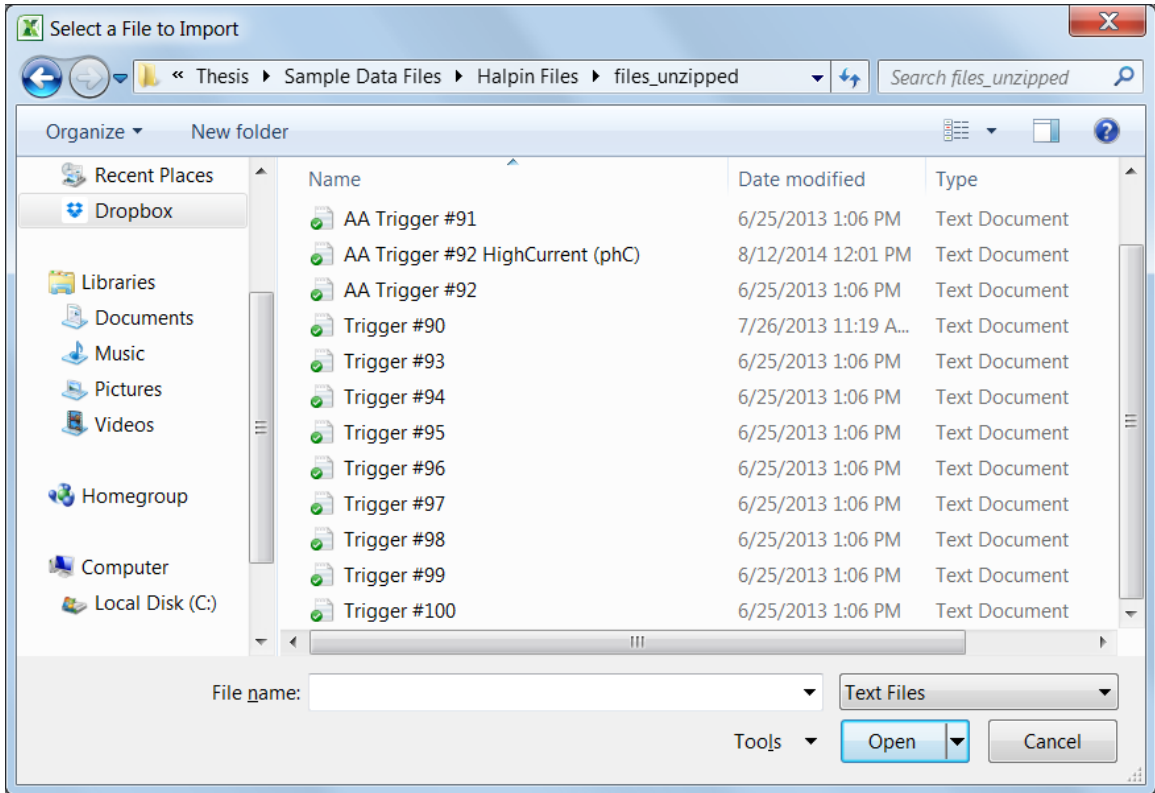


Figure 3. Select Data File(s) Dialogue Box

4.3 RMS Calculations

For each phase, the first six cycles (0.1 seconds for a 60 Hz system) are used to establish nominal RMS voltages and currents (V_{RMS_nom}) against which to compare single-cycle RMS values. Using the data at the beginning of the file means the user does not need to know the rated system voltage and allows for any tolerances or scaling particular to the measuring equipment.

$$V_{RMS_nom} = \sqrt{\frac{1}{6N} \sum_{k=h+1}^{6N+h+1} v^2(k)} \quad (3)$$

N = number of samples per cycle

h = number of header rows

v = value of voltage or current in row k of spreadsheet

Single-cycle RMS values are calculated using (1) and are stored in the corresponding “RawData” worksheet, along with the row number containing the cycle’s first data point. This data will be used for plotting.

4.4 Comparing Nominal and Single-Cycle RMS Values

Each single-cycle voltage value is compared to that phase’s nominal RMS value. If the single-cycle value is within tolerances, the program moves to the next RMS value; if the value is equal to or outside of the specified tolerances, the program recognizes a disturbance and begins printing data to the “Summary” sheet, including the file name, the rows in the spreadsheet in which the disturbance occurs, duration, voltage and current magnitudes, and a categorization of the disturbance. The program considers a disturbance to have ended when three successive comparisons are within tolerance levels (i.e., when the voltage in each phase is within acceptable limits). It then continues to examine the file for more disturbances until all cycles have been checked.

4.5 Disturbance Summary Report

The “Summary” sheet (Table 2) provides enough data for the user to determine which disturbances, if any, warrant further examination. The inclusion of the start and end rows in the summary allows the user to find the beginning of a disturbance in the imported data. The start row does not indicate the exact row in which the disturbance began. Rather, it indicates the first row of the cycle for which the single-cycle RMS value was not within the prescribed tolerances. Similarly, the last row recorded in the report is the first row of the last cycle in which the disturbance was detected. For files containing more than one disturbance, the row numbers can be used to calculate how many cycles separate the disturbances. Because the window for single-cycle RMS values is shifted

one-half cycle before the next calculation, the duration of the event is reported to the nearest half cycle. The voltage and current magnitudes are reported as a percentage of their nominal values. This information is used to categorize the disturbance as an “Overvoltage,” “Undervoltage,” or “Undervoltage due to Overcurrent” event. The third category is assigned when the single-cycle RMS current value is more than double the nominal value during an undervoltage event in the corresponding phase voltage. Such an event is likely the result of a downstream fault. All of this information is given on a per-phase basis, allowing the user to see which phases were affected.

Table 2. Disturbance Summary Report

Disturbance Summary						
Folder:	C:\Users\Owner\Dropbox\Thesis\Sample Data Files\Halpin Files\files_unzipped					
File:	2 Disturbances (AA91).txt					
			Duration	Voltage	Current	
Disturbance 1	Start Row	Last Row	(cycles)	(% of Normal)	(% of Normal)	Condition
Va	2901	3013	4	62	116	Undervoltage
Vb	2917	3013	3.5	63	125	Undervoltage
Vc	2901	3013	4	62	116	Undervoltage
Disturbance 2						
Va	8533	8645	4	62	114	Undervoltage
Vb	8549	8645	3.5	63	123	Undervoltage
Vc	8533	8645	4	61	117	Undervoltage
File:	AA Trigger #92 HighCurrent (phC).txt					
			Duration	Voltage	Current	
Disturbance 1	Start Row	Last Row	(cycles)	(% of Normal)	(% of Normal)	Condition
Va	2741	2837	3.5	77	109	Undervoltage
Vb	2741	2837	3.5	76	108	Undervoltage
Vc	2741	2837	3.5	77	220	Undervoltage due to Overcurrent
File:	Trigger #90.txt					
			Duration	Voltage	Current	
Disturbance 1	Start Row	Last Row	(cycles)	(% of Normal)	(% of Normal)	Condition
Va	-	-	-	-	-	-
Vb	2469	2485	1	164	101	Overvoltage
Vc	-	-	-	-	-	-
File:	Trigger #93.txt					
			Duration	Voltage	Current	
Disturbance 1	Start Row	Last Row	(cycles)	(% of Normal)	(% of Normal)	Condition
Va	2037	2053	1	114	100	Overvoltage
Vb	2037	2053	1	120	100	Overvoltage
Vc	-	-	-	-	-	-

Chapter 5

Disturbance Plotting

Many electromagnetic phenomena have signatures that can be easily recognized when viewed graphically [2]. While the main program places an event into one of three categories, the cause of a disturbance may be narrowed further by an individual who can recognize a particular signature in a plot of the event. With that in mind, an automated plotting routine has been developed, allowing the user to skip the step of using the row numbers in the summary table to plot the disturbance by finding those rows in the raw data.

5.1 User Interface for Plotting

The user interface allows an individual to customize the plot(s) by making the following choices (Fig. 4).

- Which disturbance(s) should be plotted?
- Should the raw data or the single-cycle RMS values be plotted?
- Should the voltage, current, or voltage and current be plotted?
- How many cycles should be plotted before and after the event?

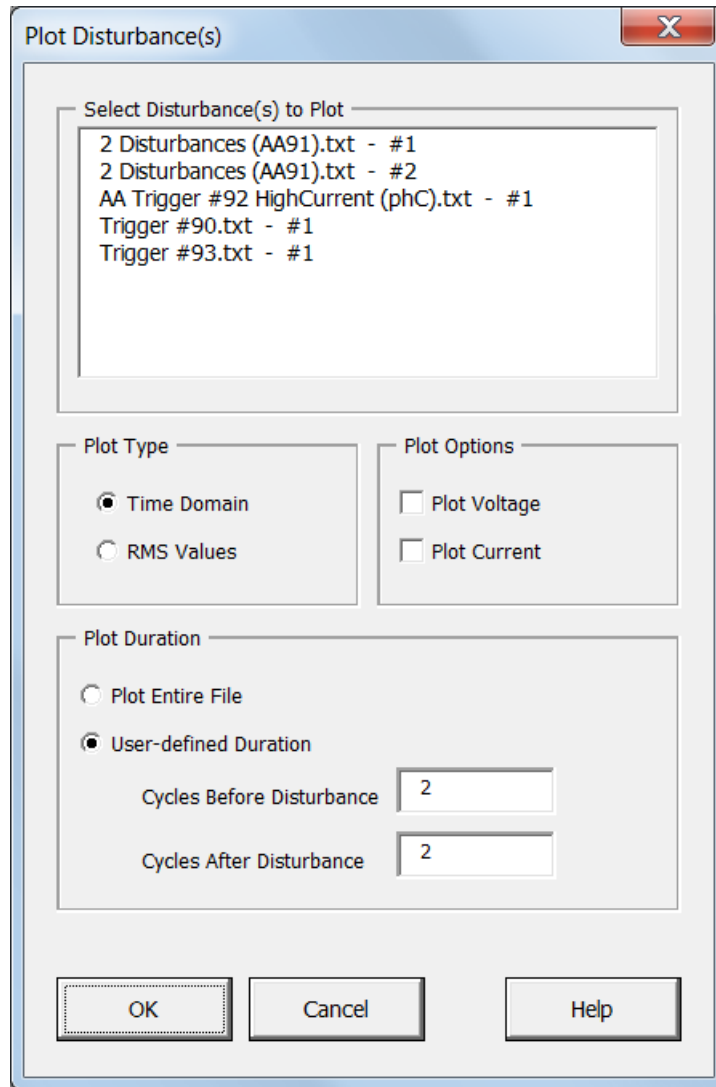


Figure 4. User Interface for Plotting Parameters

More than one disturbance may be selected for plotting. If the Time Domain Plot Type option is selected, the program plots the raw data imported from the data file. The RMS Values option plots the single-cycle RMS values calculated by the main program. Only the voltage is examined when the search for a disturbance is being conducted, but the current may also experience abnormal fluctuations. The ability to view the current and voltage simultaneously may be needed for the user to classify an event, so the option to view both on the same plot is available. Viewing the waveforms before the disturbance

will provide a frame of reference when viewing the event. Additionally, the main program deems a disturbance to be over once the voltage is again within tolerance levels. However, full voltage recovery may not occur for several cycles. Thus, the ability to select how many cycles are to be plotted before and after an event is needed. The entire file may also be plotted.

5.2 Plotting Program

When the “Plot Disturbance(s)” button on the “Home” screen is clicked, the UserForm in Fig. 4 is initialized by adding a list of all disturbances available for plotting to a VBA ListBox which then appears to the user. After all selections have been made and the OK button has been clicked, the plotting program checks that the user has made all necessary selections. If an item has been overlooked, a dialogue box appears informing the individual that not all selections have been made (Fig. 5).

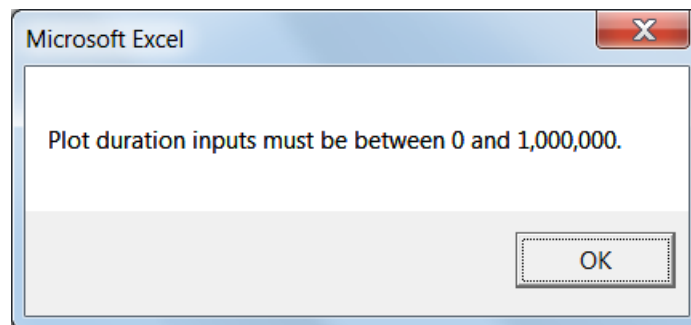


Figure 5. Plot Duration Not Defined Error Message

The program takes the user’s inputs and carries out calculations to determine the rows and columns from which to draw the data to be plotted. The name for the chart’s worksheet is then generated and the program checks to see if the plot already exists. If so, the user is alerted and the program moves on to the next disturbance to be plotted, if applicable. If the plot does not already exist, the source data for y-axis values is set, followed by the creation of a legend to identify each waveform and axes labeling. The x-

axis values (Cycles at 60 Hz) are calculated, the interval for tick marks is set, and finally a title is placed at the top of the chart (See Figures 6 & 7). On the x-axis, 0 cycles corresponds to the start row listed in the Disturbance Summary report. (See also Appendix A.)

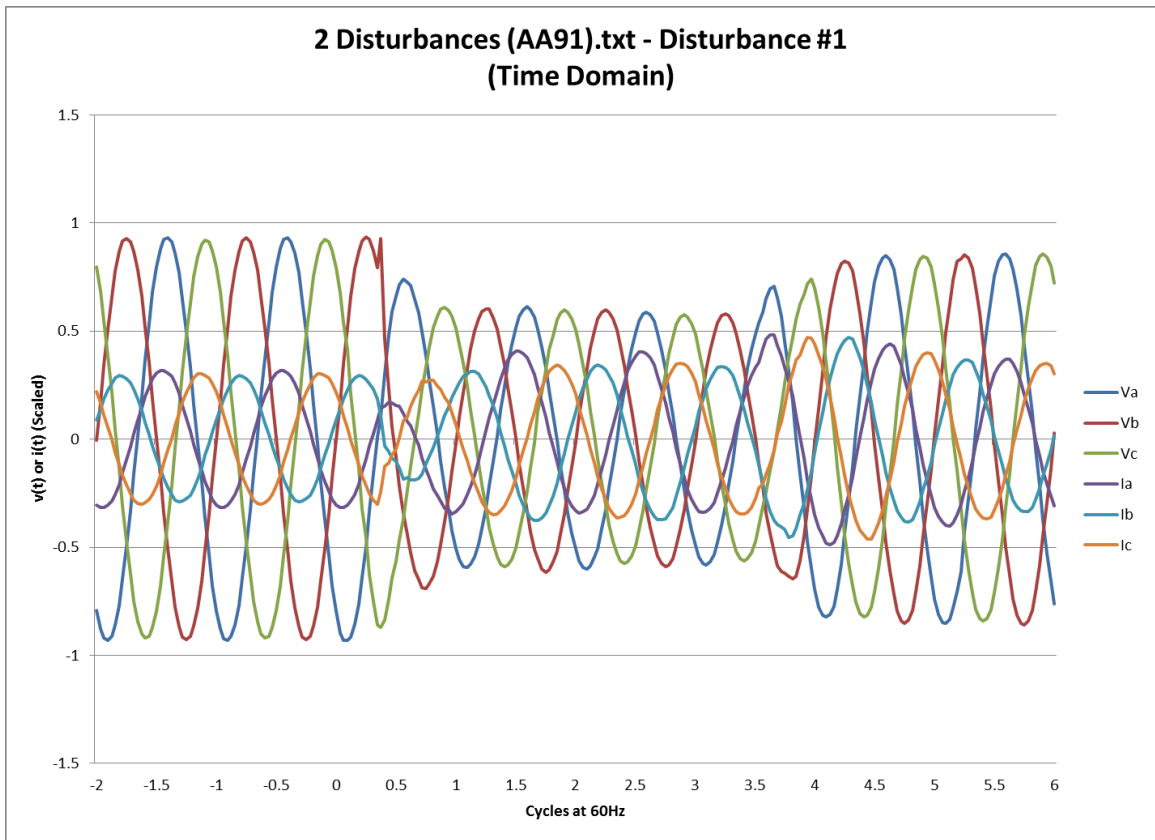


Figure 6. Time Domain Plot of Voltage and Current

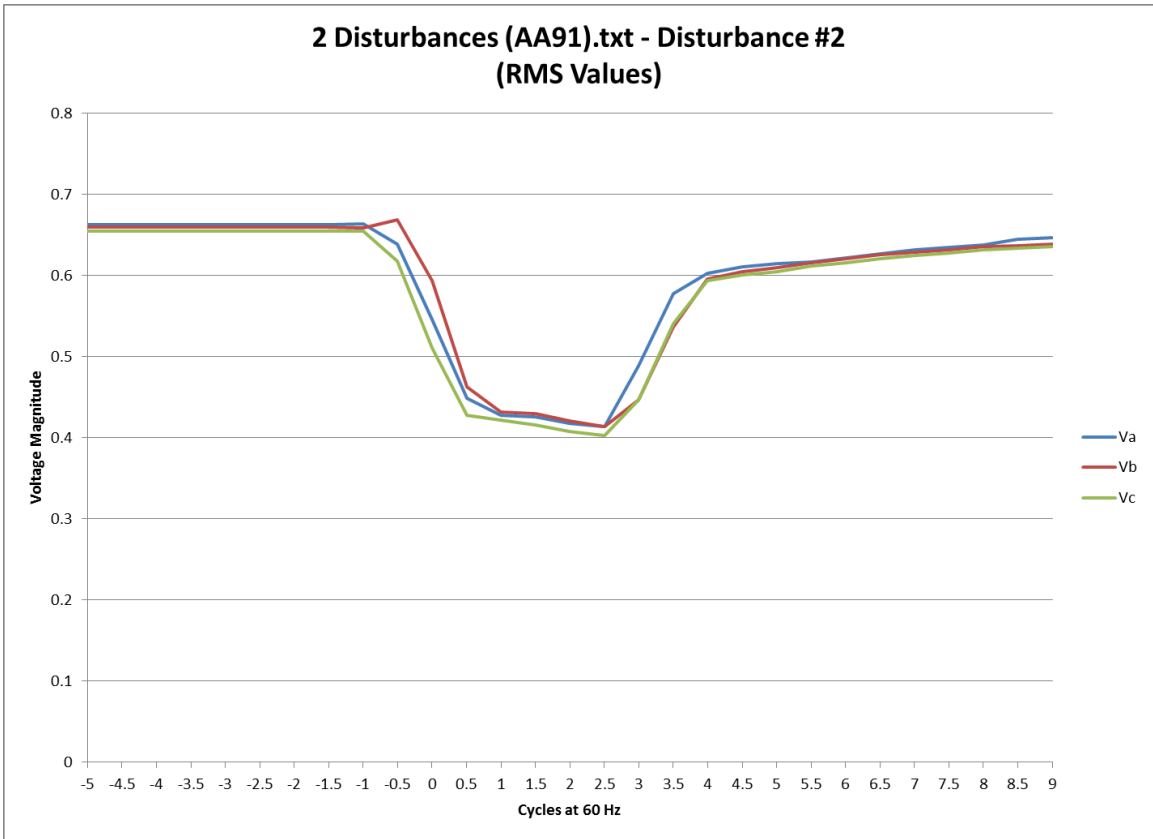


Figure 7. RMS Values Plot of Voltage

Chapter 6

Other Program Features

The main purpose of the program is to isolate disturbances in a data file and provide metrics to aid in their categorization, but a few other features have been added. They make the program easier to use, but have little to do with processing data.

6.1 Viewing Imported Data Files

The worksheets containing imported data are hidden after importing so that the workbook does not become too cluttered if several data files are analyzed at once. They can be viewed by clicking the “View Raw Data” button on the “Home” screen and selecting the desired disturbances (Fig. 8).

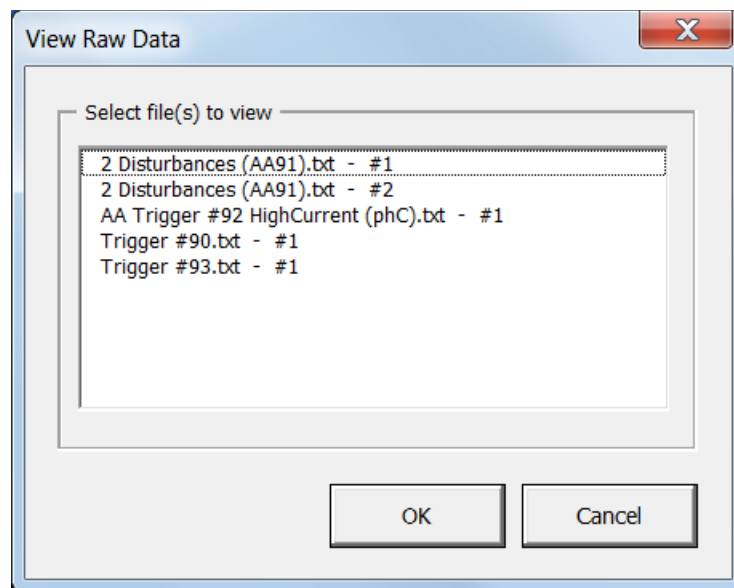


Figure 8. User Interface for Viewing Raw Data Files

6.2 Clearing the Workbook of Previous Results

Once the main program has been run, the workbook must be reset before processing more files. Clicking the “Reset Workbook” button on the “Home” screen deletes all plots and worksheets containing imported data. The “Summary” sheet is cleared of all results, as is the “Hidden” sheet (see Section 6.3). The default values for the number of header rows, sampling frequency, and thresholds which constitute a disturbance are restored to the program’s default values. If the values were changed and are needed for the next analysis, they will have to be re-entered.

6.3 The “Hidden” Worksheet

As the use of some program functions is optional, the code for each button on the “Home” screen is contained within its own VBA Module. Once the program within a module has been executed, all information stored in that module’s variables is emptied. Some information generated by one module is needed to run the code in another module. For example, if a user changes the default data file characteristics (header rows, sampling frequency, etc.), that information must be stored outside of the module in order for the main program to have access to it. Such data is stored in the “Hidden” worksheet, which is not visible to the user. Other information contained in this worksheet includes

- The row and column headings for the Disturbance Summary report
- The name of the worksheet containing imported data and the corresponding original file name
- Disturbance numbers, start rows, and end rows
- The x-axis values calculated for plotting
- The text files for help buttons.

During normal operation, a user will not need to view or manipulate the data in the “Hidden” worksheet.

6.4 Workbook Protection

The workbook has been password protected, making it a read-only file. When opening the workbook, the password must be entered in order to disable this feature. It can be saved under another name, and this new file will not be saved as read-only unless specified by the user. This feature is not meant to prevent someone from altering the workbook or its code, merely to prevent the original workbook from being changed in some way that renders the program inoperable and then saved without the ability to revert to the original, working program.

Chapter 7

Results

In total, the program processed eighteen files during the testing phase. The program operated as intended, detecting disturbances with single-cycle RMS values outside of specified limits. However, some program limitations were found.

7.1 Testing

Eleven files from actual data recorders tied to a 60 Hz system were made available for testing the program. None of these files contained current high enough to receive the “Undervoltage due to Overcurrent” classification, nor did any of these files contain more than one disturbance. From these data files, three more were created – one containing two disturbances, one with an RMS current more than double the nominal value, and one containing two disturbances and high RMS currents. Four files were created using a spreadsheet for the purpose of testing different sampling frequencies. The data in these files was completely fabricated, containing no “real-world” data. The file characteristics can be found in Table 3.

The program was able to correctly isolate the disturbances for all data files, those containing data from actual monitoring devices, modified data from monitoring devices, and simulated data. Select nominal RMS values and single-cycle RMS values calculated by the program were verified.

Table 3. Data File Characteristics

<i>Data Files</i>	
Data Source	File Count
Monitoring Device	11
Modified from Monitoring Device	3
Simulated	4
<i>Sampling Frequency</i>	
Frequencies Tested (samples/cycle)	File Count
20	1
32	14
50	1
78	1
150	1
<i>Disturbance Count</i>	
Disturbances per File	File Count
One	13
Two	3
Three	2
Disturbances Identified	25
Total Files Tested	18

7.2 Program Limitations

The program calculates nominal voltage from the first six cycles in a data file and the value is not recalculated. Thus, the possibility exists that a false disturbance will be detected if the voltage rises slowly, perhaps due to load shedding, during the timeframe in which data was recorded. Secondly, spikes in voltage may not be detected in files with high sampling frequencies. In such files, one data point does not carry as much weight in RMS calculations as it does in files with lower sampling frequencies. Thus, the higher the

sampling frequency, the higher a spike must be to cause one cycle's RMS value to exceed the threshold which constitutes a disturbance. A user with some understanding of power quality should be able to understand and recognize these issues by plotting the data.

Chapter 8

Conclusion

With the increased interest in power quality, more tools are needed to process data effectively and efficiently. The causes of electromagnetic phenomena must be identified in order to reduce or eradicate their effects on the end user. An important step in homing in on such causes is the ability to categorize disturbances and recognize patterns in waveforms that can be attributed to similar events that affect the power system. While some limitations exist, monitoring single-cycle RMS voltages updated every half cycle proved effective in detecting disturbances. With the ability to automate file importing, calculations, and plotting, Excel and VBA provide accessible tools that are well-suited to such a task.

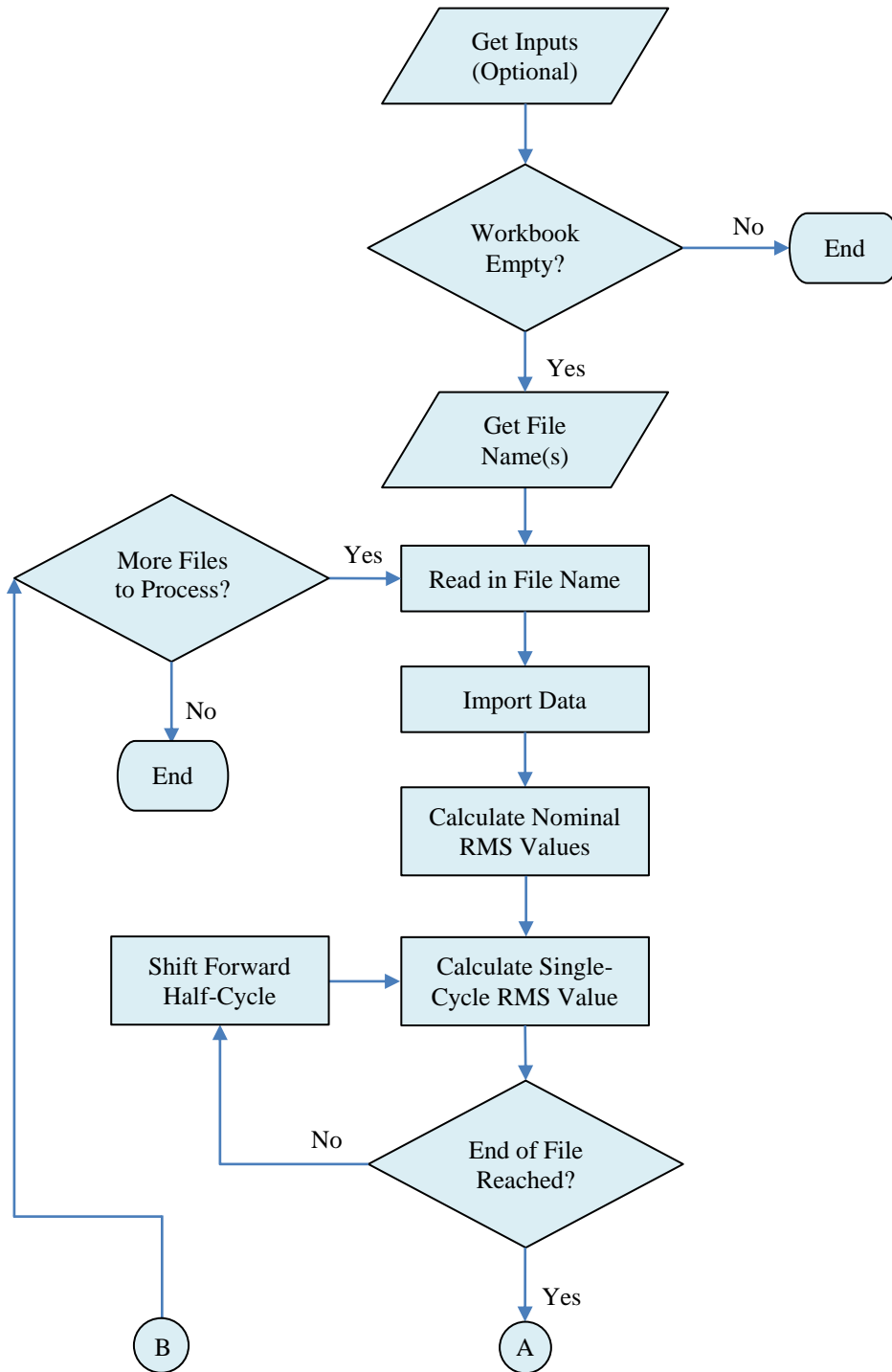
References

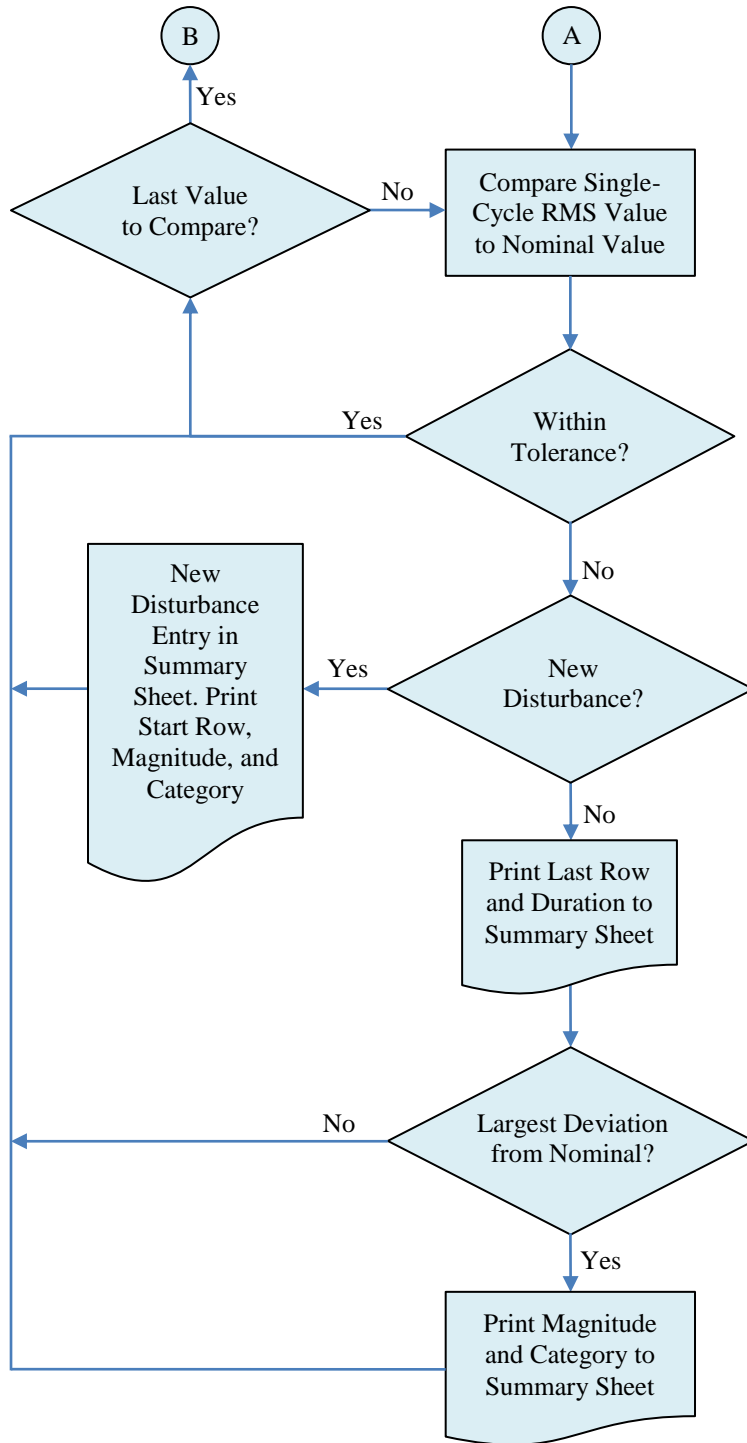
- [1] Vannoy, D.B. et al., "Roadmap for Power-Quality Standards Development," *IEEE Transactions on Industry Applications*, vol. 43, no. 2, pp. 412-421, Mar.-Apr. 2007.
- [2] IEEE Std 1159-2009, IEEE Practice for Monitoring Electric Power Quality.
- [3] Gross, Charles A., *Power System Analysis*, 2nd ed. New York: Wiley, 1986.
- [4] Mansor, M. and Rahim, N.A., "Voltage Sag Detection – A Survey," in *International Conference for Technical Postgraduates (TECHPOS)*, Kuala Lumpur, Malaysia, pp. 1-6, Dec. 2009.
- [5] Matz, Václav et al., "Automated Power Quality Monitoring System for On-line Detection and Classification of Disturbances," *Instrumentation and Measurement Technology Conference Proceedings*, Warsaw, Poland, pp. 1-6, May 2007.
- [6] Walkenbach, John., Microsoft® Excel® VBA Programming for Dummies, 2nd ed. Hoboken, NJ: Wiley, 2010.

Appendix A

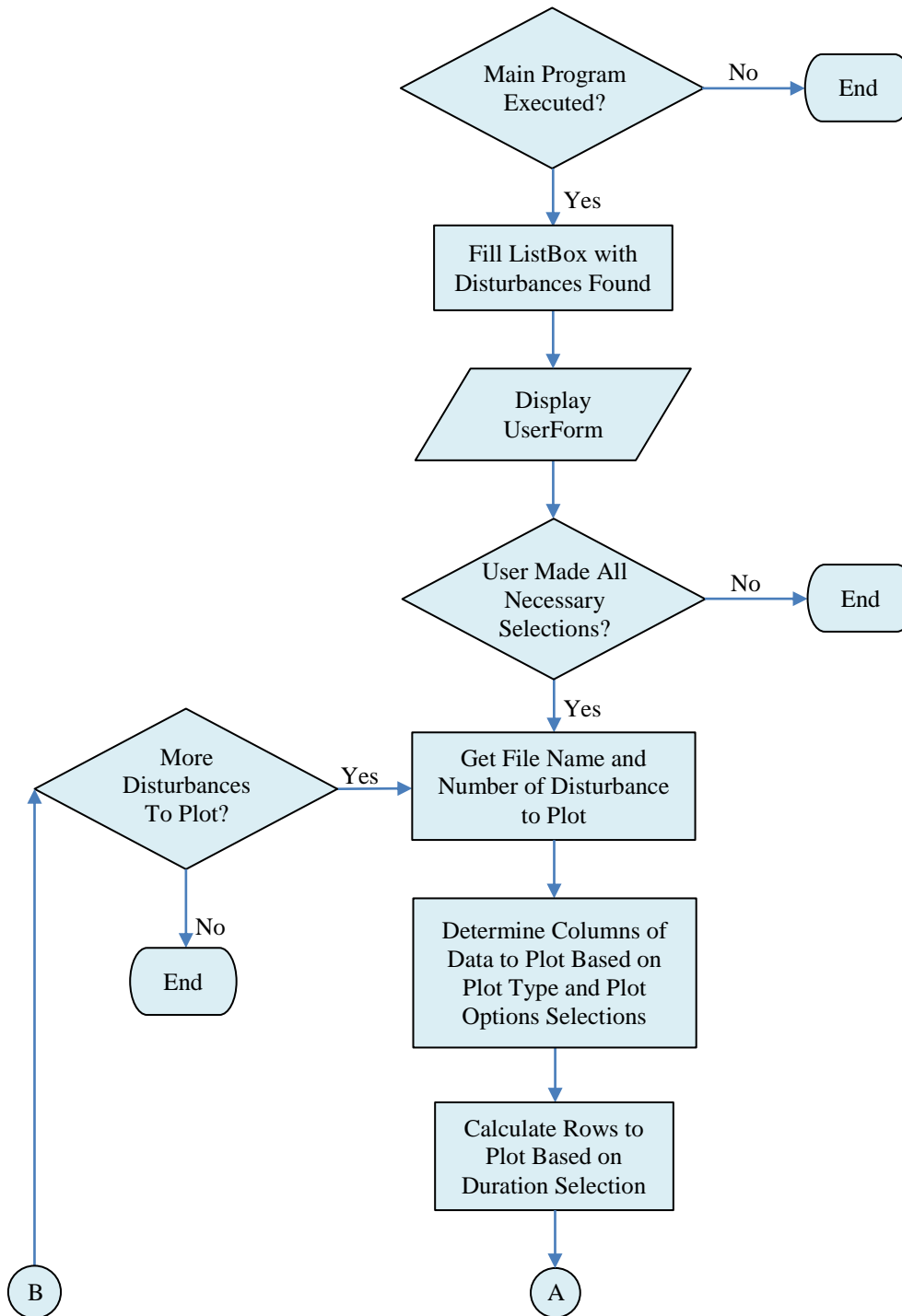
Program Flow Charts

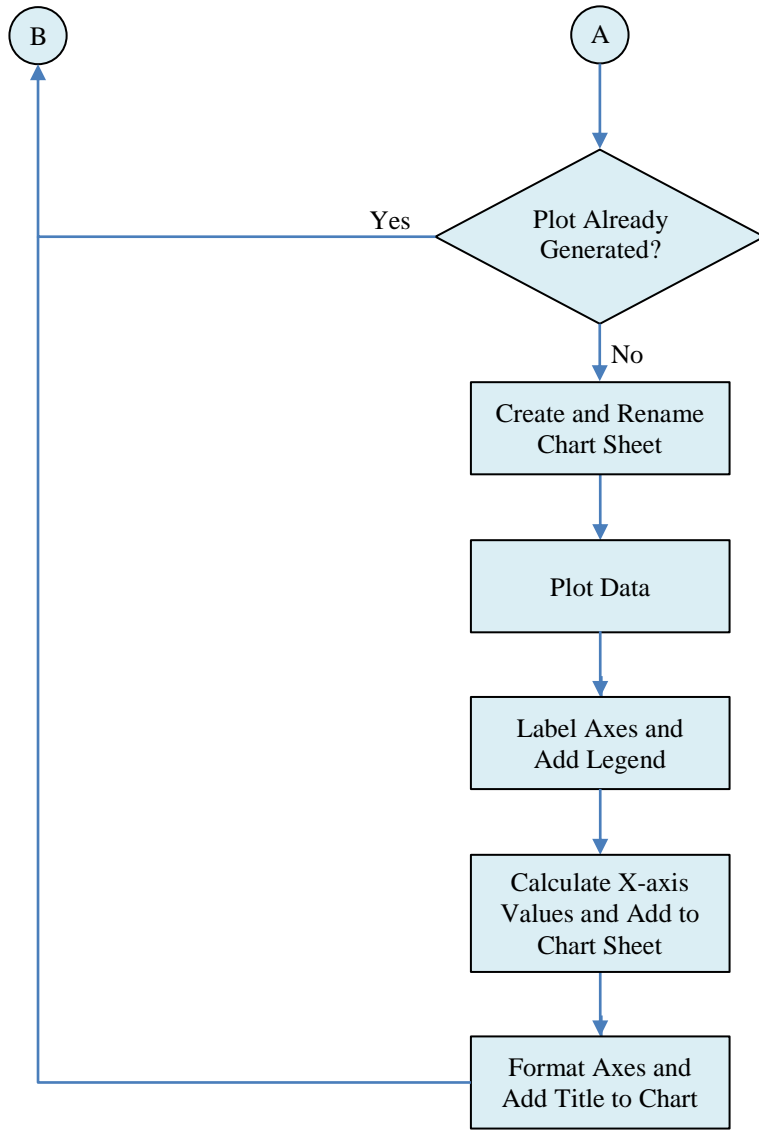
Main Program Flow Chart





Plotting Program Flow Chart





Appendix B

VBA Code

Module 1 (Main Program)

```
Option Explicit
Option Base 1

'GetFile
Dim filePaths As Variant 'file(s) selected for analysis
Dim fileIndex As Integer 'index for cycling through files selected
Dim lastData As Integer 'last row in Summary worksheet to contain data
Dim fileName As String 'Name of text file (without file path)

'GetInputs
Dim headRow As Variant 'number of rows to discard at beginning of file
Public sampFreq As Variant 'sampling frequency of measuring device

'AverageRMS
Dim startRow As Integer 'first row after header rows
Dim numCycles As Integer 'number of cycles over which to find
                        'average rms
Dim rmsAve(1 To 6) As Double 'stores average RMS values over first
                        'cycles in file

'CalcRMS
Dim lastRow As Integer 'last row of data file
Dim numElements As Double 'number of rows in rmsValues array
Dim rmsValues() As Double 'stores rms values for each cycle in data
                        'file in 1/2-cycle increments

'CompRMS
Dim rowIndex As Double, colIndex As Double 'row and column indices used
                        'when finding disturbance
Dim newDistFlag As Integer 'used to detect new disturbance within the
                        'same file
Dim distNum As Integer 'counts the number of disturbances within
                        'a data file
Dim voltagePercent As Double 'voltage of cycle as a percentage
                        'of the average
Dim currentPercent As Double 'current of cycle as a percentage
                        'of the average

'Disturbance
Dim plotTableRow As Integer

Sub Main() 'Open file with data

    Dim masterBook As Workbook, sourceBook As Workbook
    Dim fileType As String 'Allow only text file to be opened
    Dim title As String 'Title of Open File dialogue box
    Const filterIndex = 1 'Default file type to import is .txt
    Dim fileNameCount As Integer 'Number of characters in file name
                                'at end of file path
```

```

'Exit if workbook has not been reset
lastData = Worksheets("Summary").Cells(Rows.Count, "A") _
    .End(xlUp).Row
If Not lastData = 1 Or Sheets.Count <> 3 Then
    MsgBox "The Workbook still contains data from a previous" _
        & "analysis." & vbNewLine & vbNewLine & "Please" _
        & " hit the Reset Workbook button." & vbNewLine _
        & vbNewLine & "If default data file characteristics" _
        & " have been changed" & vbNewLine & "they will need" _
        & " to be entered again."
    Exit Sub
End If

Application.ScreenUpdating = False

Set masterBook = ActiveWorkbook

'Specify types of "openable" files
fileType = "Text Files (*.txt), *.txt,"
title = "Select a File to Import"

'Get the File Name(s)
filePaths = Application.GetOpenFilename _
    (fileType, filterIndex, title, , True)

'First row on Hidden sheet to store max and min rows for plotting
plotTableRow = 0

'Open Files
If IsArray(filePaths) Then

    'Print path of folder to Summary Sheet
    fileNameCount = Len(filePaths(1)) - _
        InStrRev(filePaths(1), "\")
    With Worksheets("Summary")
        .Range("A2") = "Folder:"
        .Range("A2").Font.bold = True
        .Range("B2") = Left(filePaths(1), (Len(filePaths(1)) _
            - fileNameCount - 1))
    End With

    'Analyze files one at a time
    For fileIndex = LBound(filePaths) To UBound(filePaths)

        'Import data from text file
        Sheets.Add After:=Sheets(Sheets.Count)
        fileNameCount = Len(filePaths(fileIndex)) - _
            InStrRev(filePaths(fileIndex), "\")
        fileName = Right(filePaths(fileIndex), fileNameCount)
        ActiveSheet.Name = "RawData" & fileIndex

        Workbooks.OpenText filePaths(fileIndex)

        'Copy Data to Erin's Thesis Workbook
        Set sourceBook = ActiveWorkbook
        Cells.Copy

```

```

masterBook.Activate
ActiveSheet.Paste

'Add file path to RawData# sheet
Rows("1:1").insert Shift:=xlDown
ActiveSheet.Range("A1") = filePaths(fileIndex)
ActiveSheet.Range("A1").Font.bold = True
ActiveSheet.Range("A1").Select

'Add column headings to RawData# worksheet for RMS Values
With ActiveSheet
    .Range("H3") = "Single-cycle RMS Values (updated one" _
        & "half-cycle at a time)"
    .Range("H4") = "Start Row"
    .Range("I4") = "Va"
    .Range("J4") = "Vb"
    .Range("K4") = "Vc"
    .Range("L4") = "Ia"
    .Range("M4") = "Ib"
    .Range("N4") = "Ic"
    .Range("H3:N4").Font.bold = True
End With

'Close Sourcebook
sourceBook.Activate
Application.CutCopyMode = False
sourceBook.Close False

'Print file name to Summary Sheet and format
lastData = Worksheets("Summary").Cells(Rows.Count, "A") _
    .End(xlUp).Row

With Worksheets("Summary")
    .Range("A" & lastData + 2) = "File:"
    .Range("A" & lastData + 2).Font.bold = True
    .Range("B" & lastData + 2) = fileName

    .Range("A" & lastData + 2, "G" & lastData + 2). _
        Interior.color = RGB(49, 134, 155)
    .Range("A" & lastData + 2, "G" & lastData + 2). _
        Font.color = RGB(255, 255, 255)
    .Range("B" & lastData + 2).Font.bold = True
End With

Call AverageRMS
Call CalcRMS
Call CompRMS

'Print message to spreadsheet if no disturbance is found
If Worksheets("Summary").Range("A" & lastData + 2) = _
    "File:" Then
    Worksheets("Summary").Range("A" & lastData + 4) = _
        "No Disturbance Found"
End If

'Add bottom border to table for current file
lastData = Worksheets("Summary").Cells(Rows.Count, "A") _

```



```

        .End(xlUp).Row
Worksheets("Summary").Range("A" & lastData, "G" &
lastData).Borders(xlEdgeBottom).Weight = xlThin

'Hide RawData worksheets
Worksheets("RawData" & fileIndex).Visible = False

Next fileIndex

Else 'Cancel button clicked. Alert user.
    MsgBox "No file was selected."
    Exit Sub
End If

'Set number of decimal places to display in Summary Sheet
Worksheets("Summary").Columns("E:F").NumberFormat = "0"

Worksheets("Summary").Select

End Sub

Sub AverageRMS()

    Dim i As Integer, n As Integer 'row and column indices
    Dim rmsSum As Double 'used to calculate rms

    'Read in values from "Hidden" worksheet
    headRow = Range("HeaderRows") + 1
    sampFreq = Range("SamplingFrequency")

    startRow = headRow + 1 'first row containing useful data
    numCycles = 6 '6 cycles = 0.1 seconds
    rmsSum = 0

    'Calculate rms value for first numCycles
    For n = 1 To 6
        For i = startRow To (headRow + numCycles * sampFreq)
            rmsSum = rmsSum + (Cells(i, n)) ^ 2
        Next i

        rmsAve(n) = Sqr(rmsSum / (numCycles * sampFreq))
        rmsSum = 0
    Next n

End Sub

Sub CalcRMS()

    Dim numElD As Double 'number of cycles that will be examined
    Dim m As Integer 'index for rmsValues array
    Dim i As Integer 'index for moving through code
        '1/2-cycle at a time
    Dim n As Integer 'column index
    Dim j As Integer 'index for stepping through one cycle
    Dim rmsSum As Double 'used to calculate rms

    'Calculate number of array elements needed to store RMS values

```

```

lastRow = WorksheetFunction.CountA(Range("A:A"))
numEld = (lastRow - headRow) / (sampFreq / 2)
numElements = Application.WorksheetFunction. _
    RoundDown(numEld, 0) - 1
ReDim rmsValues(1 To numElements, 1 To 7)

m = 1

'Calculate rms values in 1/2-cycle steps and store in rmsValues
For i = startRow To (lastRow - (sampFreq - 1)) _
    Step (sampFreq / 2)

    'Store first row number of each cycle
    rmsValues(m, 1) = i
    ActiveSheet.Cells(m + 4, 8) = i

    For n = 1 To 6
        For j = 0 To (sampFreq - 1)
            rmsSum = rmsSum + (Cells(i + j, n)) ^ 2
        Next j

        rmsValues(m, n + 1) = Sqr(rmsSum / sampFreq)
        ActiveSheet.Cells(m + 4, n + 8) = rmsValues(m, n + 1)
        rmsSum = 0
    Next n

    m = m + 1

Next i

End Sub

Sub CompRMS()

Dim minVol As Double 'minimum acceptable percent of normal rms
Dim maxVol As Double 'maximum acceptable percent of normal rms

distNum = 0
newDistFlag = 3
minVol = Range("MinimumVoltage")
maxVol = Range("MaximumVoltage")

'Find disturbances by comparing rmsAve and rmsValues matrices
For rowIndex = 1 To numElements
    For colIndex = 1 To 3

        newDistFlag = newDistFlag + 1

        If rmsValues(rowIndex, colIndex + 1) >= _
            maxVol * rmsAve(colIndex) Or _
            rmsValues(rowIndex, colIndex + 1) <= _
            minVol * rmsAve(colIndex) Then
            voltagePercent = rmsValues(rowIndex, colIndex + 1) _
                / rmsAve(colIndex) * 100
            currentPercent = rmsValues(rowIndex, colIndex + 4) _
                / rmsAve(colIndex + 3) * 100
        End If
    Next colIndex
Next rowIndex

```

```

        Call Disturbance
    End If

    Next colIndex
Next rowIndex

End Sub

Sub Disturbance()

'Print disturbance number and copy table to Summary sheet
If newDistFlag > 3 Then

    distNum = distNum + 1

    'Fill in information needed to plot disturbance
    plotTableRow = plotTableRow + 1
    With Worksheets("Hidden")

        .Cells(plotTableRow, 9) = "RawData" & fileIndex
        .Cells(plotTableRow, 10) = fileName
        .Cells(plotTableRow, 11) = "#" & distNum
        .Cells(plotTableRow, 12) = rmsValues(rowIndex, 1)

    End With

    lastData = Worksheets("Summary") _
        .Cells(Rows.Count, "A").End(xlUp).Row

    'Print and format results table to Summary Sheet
    If distNum = 1 Then
        Sheets("Hidden").Range("A6:G10").Copy _
            Worksheets("Summary").Range("A" & lastData + 1)
        Worksheets("Summary").Range("A" & lastData + 2) = _
            "Disturbance " & distNum
    Else
        Sheets("Hidden").Range("A8:G10").Copy _
            Worksheets("Summary").Range("A" & lastData + 2)
        Worksheets("Summary").Range("A" & lastData + 1) = _
            "Disturbance " & distNum
        Worksheets("Summary").Range("A" & lastData + 1). _
            Font.color = RGB(49, 134, 155)
        Worksheets("Summary").Range("A" & lastData + 1). _
            Font.bold = True
    End If
End If

newDistFlag = 0
lastData = Worksheets("Summary").Cells(Rows.Count, "A") _
    .End(xlUp).Row

With Worksheets("Summary")

    'Print disturbance info to Summary sheet
    If .Cells((lastData - 3 + colIndex), 2) = "-" Then

        'Print first row of data in which disturbance is found

```

```

.Cells((lastData - 3 + colIndex), 2) = _
    rmsValues(rowIndex, 1)

'Print Voltage and Current % to Summary sheet
.Cells((lastData - 3 + colIndex), 5) = voltagePercent
.Cells((lastData - 3 + colIndex), 6) = currentPercent

'Print voltage condition to Summary sheet
If voltagePercent > 100 Then
    .Cells((lastData - 3 + colIndex), 7) = "Overvoltage"
ElseIf (voltagePercent < 100) And _
    (rmsValues(rowIndex, colIndex + 4) > _
        rmsAve(colIndex + 3) * 2) Then
    .Cells((lastData - 3 + colIndex), 7) = _
        "Undervoltage due to Overcurrent"
Else
    .Cells((lastData - 3 + colIndex), 7) = _
        "Undervoltage"
End If

Else
'Print latest row of data in which disturbance is found
.Cells((lastData - 3 + colIndex), 3) = _
    rmsValues(rowIndex, 1)

'Calculate duration of disturbance
.Cells((lastData - 3 + colIndex), 4) = _
    (.Cells((lastData - 3 + colIndex), 3) - _
        .Cells((lastData - 3 + colIndex), 2)) / sampFreq + 0.5

'Print Voltage and Current % to Summary sheet
'if its deviation is largest so far
If Abs(voltagePercent - 100) _
    > Abs(.Cells((lastData - 3 + colIndex), 5) - 100) Then
    .Cells((lastData - 3 + colIndex), 5) = _
        voltagePercent
    .Cells((lastData - 3 + colIndex), 6) = _
        currentPercent

'Add voltage condition to Summary Sheet
If voltagePercent > 100 Then
    .Cells((lastData - 3 + colIndex), 7) = _
        "Overvoltage"
ElseIf (voltagePercent < 100) And _
    (rmsValues(rowIndex, colIndex + 4) > _
        rmsAve(colIndex + 3) * 2) Then
    .Cells((lastData - 3 + colIndex), 7) = _
        "Undervoltage due to Overcurrent"
Else
    .Cells((lastData - 3 + colIndex), 7) = _
        "Undervoltage"
End If

End If

'Last cell to plot
Worksheets("Hidden").Cells(plotTableRow, 13) = _

```

```
        rmsValues(rowIndex, 1)
    End If
End With
End Sub
```

Module 2 (Reset Workbook)

Option Explicit

Sub ResetThesisWorkbook()

Dim Ans As Integer

Dim i As Integer

Dim sheetName As String

```
Ans = MsgBox("Resetting the workbook will also _
            & " restore the program defaults.", 1) _
'Exit if user hit cancel
If Ans = 2 Then Exit Sub
```

Application.DisplayAlerts = False

i = 1

```
Do Until Sheets.Count = 3
    sheetName = Sheets(i).Name
    If Sheets(i).Name <> "Home" And _
        Sheets(i).Name <> "Summary" _
        And Sheets(i).Name <> "Hidden" Then
        Sheets(i).Delete
        i = i - 1
    End If
    i = i + 1
Loop
```

Application.DisplayAlerts = True

```
Worksheets("Hidden").Activate
Range("HeaderRows") = 3
Range("SamplingFrequency") = 32
Range("MinimumVoltage") = 0.9
Range("MaximumVoltage") = 1.1
Range("I:ZZ").Clear
Range("A1").Activate
```

```
Worksheets("Summary").Activate
Range("2:1048576").Clear
Range("A1").Activate
```

```
Worksheets("Home").Activate
Cells.ClearContents
Range("A1").Select
```

End Sub

Module 3 (Edit Default Data File Characteristics)

Option Explicit

Sub GetInputs()

```
Dim userHeadRow As Variant
Dim userSampFreq As Variant
Dim userMinVol As Variant
Dim userMaxVol As Variant

'Clear default values
Range("HeaderRows").Clear
Range("SamplingFrequency").Clear

'Acquire number of header rows from user
Do Until IsNumeric(userHeadRow) And Not IsEmpty(userHeadRow) _
    And userHeadRow >= 0
    userHeadRow = InputBox("Enter the number of Header Rows in" _
        & " the Data File." & vbNewLine & vbNewLine & _
        "A value of 3 will be used if nothing is" _
        & " entered." & vbNewLine & "The value must" _
        & " be a positive integer", _
        "Enter Number of Header Rows", 3)
    If userHeadRow = "" Then
        userHeadRow = 3
    ElseIf userHeadRow < 0 Then
        MsgBox "You must enter a positive number."
    ElseIf IsNumeric(userHeadRow) Then
        If (userHeadRow - Int(userHeadRow) <> 0) Then
            userHeadRow = Empty
            MsgBox "You must enter a positive integer."
        End If
    ElseIf Not IsNumeric(userHeadRow) Then
        MsgBox "You must enter a numeric value."
    End If
Loop

Range("HeaderRows").Value = userHeadRow

'Acquire sampling frequency from user
Do Until IsNumeric(userSampFreq) And Not IsEmpty(userSampFreq) _
    And userSampFreq > 0
    userSampFreq = InputBox("Enter the Sampling Frequency of the" _
        & " Data File" & vbNewLine & "(in samples/" _
        & "cycle)." & vbNewLine & vbNewLine & "A" _
        & " value of 32 will be used if nothing is" _
        & " entered." & vbNewLine & "The value must" _
        & " be a positive even integer.", _
        "Enter Sampling Frequency", 32)
    If userSampFreq = "" Then
        userSampFreq = 32
    ElseIf userSampFreq <= 0 Then
        MsgBox "You must enter a positive number."
    ElseIf IsNumeric(userSampFreq) Then
        If (userSampFreq - Int(userSampFreq) <> 0) _
```

```

        Or Not WorksheetFunction.IsEven(userSampFreq) Then
            userSampFreq = Empty
            MsgBox "You must enter an even positive integer."
        End If
    ElseIf Not IsNumeric(userSampFreq) Then
        MsgBox "You must enter a numeric value."
    End If
Loop

Range("SamplingFrequency").Value = userSampFreq

'Acquire minimum voltage tolerance from user
Do Until IsNumeric(userMinVol) And Not IsEmpty(userMinVol) _
    And userMinVol > 0
    userMinVol = InputBox("Enter the minimum acceptable" _
        & " voltage (in p.u.)." & vbNewLine _
        & vbNewLine & "A value of 0.9 will be used" _
        & " if nothing is entered.", _
        "Enter Minimum Voltage Tolerance", 0.9)
    If userMinVol = "" Then
        userMinVol = 0.9
    ElseIf userMinVol <= 0 Then
        MsgBox "You must enter a positive number."
    ElseIf Not IsNumeric(userMinVol) Then
        MsgBox "You must enter a numeric value."
    End If
Loop

Range("MinimumVoltage").Value = userMinVol

'Acquire maximum voltage tolerance from user
Do Until IsNumeric(userMaxVol) And Not IsEmpty(userMaxVol) _
    And userMaxVol > 0
    userMaxVol = InputBox("Enter the maximum acceptable" _
        & " voltage (in p.u.)." & vbNewLine _
        & vbNewLine & "A value of 1.1 will be used" _
        & " if nothing is entered.", _
        "Enter Maximum Voltage Tolerance", 1.1)
    If userMaxVol = "" Then
        userMaxVol = 1.1
    ElseIf userMaxVol <= 0 Then
        MsgBox "You must enter a positive number."
    ElseIf Not IsNumeric(userMaxVol) Then
        MsgBox "You must enter a numeric value."
    ElseIf userMaxVol <= userMinVol Then
        MsgBox "The maximum tolerance must be greater than" _
            & " the minimum tolerance."
        userMaxVol = Empty
    End If
Loop

Range("MaximumVoltage").Value = userMaxVol

End Sub

```


Module 4 (Call Plotting UserForm)

Option Explicit

```
Public Sub UserPlotDialogBox()  
  
    If IsEmpty(Worksheets("Hidden").Range("I1")) Then  
        MsgBox ("Please run the program first.")  
    Else  
        UserForm1.Show  
    End If  
  
End Sub
```

UserForm 1 (Plotting UserForm, see Fig. 4)

Option Explicit

```
Dim selectionMade As Integer 'Flag for whether or not a  
                             'disturbance was selected  
Dim listBoxCount As Integer 'Number of items to display in listbox  
Dim plotStartRow As Single, plotEndRow As Single 'Rows in data  
                                                'sheet to plot  
Dim plotStartCol As String, plotEndCol As String 'Columns in data  
                                                'sheet to plot  
Dim plotFileIndex As Integer 'File currently being plotted  
Dim dataSheetName As String 'name of RawData worksheet currently  
                             'being plotted  
Dim preCycles As Double, postCycles As Double 'Cycles plotted before  
                                                'and after disturbance  
Dim plotSheetName As String 'Name that will be given  
                             'to chart worksheets  
Dim seriesCount As Integer  
  
Private Sub CancelButton_Click()  
    Unload UserForm1  
End Sub  
  
Private Sub CommandButton1_Click()  
  
Dim plotHelp As Object  
  
    Application.ScreenUpdating = False  
    Set plotHelp = ThisWorkbook.Worksheets("Hidden").OLEObjects(2)  
    plotHelp.Verb Verb:=xlPrimary  
    Worksheets("Hidden").Visible = False  
    Worksheets("Home").Activate  
  
End Sub  
  
Private Sub Frame1_Click()  
  
End Sub  
  
Private Sub ListBox1_Click()
```

```

End Sub

Private Sub OKButton_Click()

    selectionMade = 0

    'Ensure user selected a disturbance to plot
    For plotFileIndex = 0 To ListBox1.ListCount - 1

        If ListBox1.Selected(plotFileIndex) Then
            selectionMade = 1
        End If

    Next plotFileIndex

    If selectionMade = 0 Then
        MsgBox "No disturbance was selected."
        Unload UserForm1
        Exit Sub
    End If

    'Ensure user has selected something to plot
    If CheckBox1.Value = False And CheckBox2.Value = False Then
        MsgBox "You must select which quantities to plot." _
            & vbNewLine & "Please check at least one box" _
            & " under Plot Options."
        Unload UserForm1
        Exit Sub
    End If

    'Process User-defined duration data
    If OptionButton4 = True Then
        If IsNumeric(TextBox1) And IsNumeric(TextBox2) _
            And TextBox1 >= 0 And TextBox2 >= 0 _
            And TextBox1 <= 1000000 And TextBox2 <= 1000000 Then
            'Calculate cycles to plot before and after disurbance
            preCycles = WorksheetFunction.
                RoundUp(2 * TextBox1, 0) / 2
            postCycles = WorksheetFunction.
                RoundUp(2 * TextBox2, 0) / 2
        Else: MsgBox "Plot duration inputs must be" _
            & " between 0 and 1,000,000."
            Unload UserForm1
            Exit Sub
        End If
    End If

    'Set columns to plot based on user's selection
    If OptionButton1 = True Then 'Time domain plot
        If CheckBox1.Value = True And CheckBox2.Value = True Then
            plotStartCol = "A"
            plotEndCol = "F"
        ElseIf CheckBox1.Value = True Then
            plotStartCol = "A"
            plotEndCol = "C"
        ElseIf CheckBox2.Value = True Then
            plotStartCol = "D"
        End If
    End If

```

```

        plotEndCol = "F"
    End If
ElseIf OptionButton2 = True Then 'RMS plot
    If CheckBox1.Value = True And CheckBox2.Value = True Then
        plotStartCol = "I"
        plotEndCol = "N"
    ElseIf CheckBox1.Value = True Then
        plotStartCol = "I"
        plotEndCol = "K"
    ElseIf CheckBox2.Value = True Then
        plotStartCol = "L"
        plotEndCol = "N"
    End If
End If

'Cycle through plotting files
For plotFileIndex = 0 To ListBox1.ListCount - 1

    If ListBox1.Selected(plotFileIndex) Then
        'Name of worksheet containing Raw Data
        'which is currently being plotted
        dataSheetName = Worksheets("Hidden"). _
            Range("I" & plotFileIndex + 1).Value

        If OptionButton1 = True Then 'User wants time domain plot

            'Calculate row numbers in worksheet which
            'function as bounds for plotting
            If OptionButton4 = True Then 'User wants to define
                'plot duration
                plotStartRow = Worksheets("Hidden"). _
                    Range("L" & (plotFileIndex + 1)) _
                    - preCycles * Range("SamplingFrequency")
                plotEndRow = Worksheets("Hidden"). _
                    Range("M" & (plotFileIndex + 1)) _
                    + (postCycles + 0.5) * Range("SamplingFrequency")

                'Reset row numbers for plotting if user input
                'numbers which exceed limits of data
                If plotStartRow < Range("HeaderRows") + 2 Then
                    plotStartRow = Range("HeaderRows") + 2
                    preCycles = (Worksheets("Hidden"). _
                        Range("L" & (plotFileIndex + 1)) - _
                        (Range("HeaderRows") + 2)) / _
                        Range("SamplingFrequency")
                End If
                If plotEndRow > WorksheetFunction.CountA _
                    (Worksheets(dataSheetName).Range("A:A")) Then
                    plotEndRow = WorksheetFunction.CountA _
                        (Worksheets(dataSheetName).Range("A:A"))
                End If

            Else 'User wants to plot all data
                plotStartRow = Range("HeaderRows") + 2
                preCycles = (Worksheets("Hidden"). _
                    Range("L" & (plotFileIndex + 1)) - _
                    (Range("HeaderRows") + 2)) / _

```

```

        Range("SamplingFrequency")
        plotEndRow = WorksheetFunction.CountA _
            (Worksheets(dataSheetName).Range("A:A"))
    End If

    Call TimeDomainPlot
End If

If OptionButton2 = True Then 'User wants RMS plot

    'Calculate row numbers in worksheet which
    'function as bounds for plotting
    If OptionButton4 = True Then 'User wants to define
        'plot duration
        plotStartRow = (Worksheets("Hidden"). _
            Range("L" & (plotFileIndex + 1)) _
            - (Range("HeaderRows") + 2)) / _
            (Range("SamplingFrequency") / 2) _
            + 5 - 2 * preCycles
        plotEndRow = (Worksheets("Hidden"). _
            Range("M" & (plotFileIndex + 1)) _
            - (Range("HeaderRows") + 2)) / _
            (Range("SamplingFrequency") / 2) _
            + 5 + 2 * postCycles + 1

        'Reset row numbers for plotting if user input
        'numbers which exceed limits of data
        If plotStartRow < 5 Then
            plotStartRow = 5
            preCycles = (Worksheets("Hidden"). _
                Range("L" & (plotFileIndex + 1)) - _
                (Range("HeaderRows") + 2)) / _
                Range("SamplingFrequency")
        End If
        If plotEndRow > Worksheets(dataSheetName). _
            Cells(Worksheets(dataSheetName) _
                .Rows.Count, "H").End(xlUp).Row Then
            plotEndRow = Worksheets(dataSheetName) _
                .Cells(Worksheets(dataSheetName) _
                .Rows.Count, "H").End(xlUp).Row
        End If

    Else 'User wants to plot all data
        plotStartRow = 5
        preCycles = (Worksheets("Hidden"). _
            Range("L" & (plotFileIndex + 1)) - _
            (Range("HeaderRows") + 2)) / _
            Range("SamplingFrequency")
        plotEndRow = Worksheets(dataSheetName) _
            .Cells(Worksheets(dataSheetName) _
            .Rows.Count, "H").End(xlUp).Row
    End If

    Call RMSPlot
End If

End If

```

```

Next plotFileIndex

Sheets(plotSheetName).Select

Unload UserForm1

End Sub

Private Sub OptionButton3_Click()

End Sub

Sub UserForm_Initialize()

Dim i As Integer 'loop counter

listBoxCount = Application.WorksheetFunction.CountA(Worksheets("Hidden").Range("I:I"))

'Fill the list box
With ListBox1

    For i = 1 To listBoxCount
        .AddItem Worksheets("Hidden").Range("J" & i) & _
            " - " & Worksheets("Hidden").Range("K" & i)
    Next i

End With

'Select the first list item
ListBox1.ListIndex = 0

End Sub

Sub TimeDomainPlot()

Dim ws As Chart 'Chart names; circumvent problem of creating
                'same plot twice
Dim plottedQuantities As String 'Used in sheet name to tell if
                                'V, I, or V&I are plotted
Dim yLabel As String 'y-axis label
Dim xaxisvalues As Integer 'Number of rows plotted
Dim lastCol As Integer 'last column in "Hidden" worksheet
                        'containing data
Dim n As Integer, m As Integer 'loop counters

'Generate name for chart sheet
If CheckBox1 = True And CheckBox2 = True Then
    plottedQuantities = " (VI)"
ElseIf CheckBox1 = True Then
    plottedQuantities = " (V)"
ElseIf CheckBox2 = True Then
    plottedQuantities = " (I)"
End If

```

```

plotSheetName = Worksheets("Hidden"). _
    Range("I" & plotFileIndex + 1) & "-" _
    & Worksheets("Hidden").Range("K" & plotFileIndex + 1) _
    & ", Time" & plottedQuantities

'Ensure this plot has not already been generated
For Each ws In ThisWorkbook.Charts
    If ws.Name = plotSheetName Then
        MsgBox "A time domain plot for " & Worksheets("Hidden") _
            .Range("J" & plotFileIndex + 1).Value _
            & " - Disturbance " & Worksheets("Hidden"). _
            Range("K" & plotFileIndex + 1) & plottedQuantities _
            & " has already been generated."
        Exit Sub
    End If
Next ws

'Rename chart sheet
Charts.Add.Name = plotSheetName
ActiveSheet.Move After:=Sheets(Sheets.Count)

'Select chart type, select data to plot,
'LABEL MY AXES (Thanks Dr. G!)
'Rename Series
With ActiveChart
    .PlotArea.Select
    .ChartType = xlLine

    'Select source data for graph
    .SetSourceData Source:=Sheets(dataSheetName). _
        Range(plotStartCol & plotStartRow, _
            plotEndCol & plotEndRow)

    'Rename labels in legend of chart
    If CheckBox1.Value = True And CheckBox2.Value = True Then
        .SeriesCollection(1).Name = """"Va""""
        .SeriesCollection(2).Name = """"Vb""""
        .SeriesCollection(3).Name = """"Vc""""
        .SeriesCollection(4).Name = """"Ia""""
        .SeriesCollection(5).Name = """"Ib""""
        .SeriesCollection(6).Name = """"Ic""""
        yLabel = "v(t) or i(t) (Scaled)"
        seriesCount = 6
    ElseIf CheckBox1.Value = True Then
        .SeriesCollection(1).Name = """"Va""""
        .SeriesCollection(2).Name = """"Vb""""
        .SeriesCollection(3).Name = """"Vc""""
        yLabel = "v(t) (Scaled)"
        seriesCount = 3
    ElseIf CheckBox2 = True Then
        .SeriesCollection(1).Name = """"Ia""""
        .SeriesCollection(2).Name = """"Ib""""
        .SeriesCollection(3).Name = """"Ic""""
        yLabel = "i(t) (Scaled)"
        seriesCount = 3
    End If

```

```

'Label axes
.Axes(xlCategory, xlPrimary).HasTitle = True
.Axes(xlCategory, xlPrimary).AxisTitle. _
    Characters.Text = "Cycles at 60Hz"
.Axes(xlValue, xlPrimary).HasTitle = True
.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text _
    = yLabel

.PageSetup.LeftFooter = Worksheets(dataSheetName).Range("A1")
.PageSetup.RightFooter = "&D &T"

End With

'Adjust x-axis values
xaxisvalues = plotEndRow - plotStartRow + 1
lastCol = Sheets("Hidden").Cells(1, Worksheets("Hidden"). _
    Columns.Count).End(xlToLeft).Column

'Print first x-axis value to "Hidden"
Worksheets("Hidden").Cells(1, lastCol + 1) = (-1 * preCycles)

'Calculate and print subsequent x-axis values to "Hidden"
For n = 1 To (xaxisvalues - 1)
    Worksheets("Hidden").Cells(n + 1, lastCol + 1) = _
        Worksheets("Hidden").Cells(n, lastCol + 1).Value _
            + 1 / Range("SamplingFrequency")
Next n

'Format chart
With ActiveChart
    For m = 1 To seriesCount
        .SeriesCollection(m).XValues = Worksheets("Hidden"). _
            Columns(lastCol + 1)
    Next m
    .Axes(xlCategory).TickMarkSpacing = _
        Range("SamplingFrequency") / 2
    .Axes(xlCategory).TickLabelSpacing = _
        Range("SamplingFrequency") / 2
    .Axes(xlCategory).TickLabelPosition = xlLow
    .SetElement(msoElementChartTitleAboveChart)
    .ChartTitle.Text = Worksheets("Hidden"). _
        Range("J" & plotFileIndex + 1) _
            & " - Disturbance " & Worksheets("Hidden") _
                .Range("K" & plotFileIndex + 1) _
                    & Chr(13) & "(Time Domain)"
    .ChartArea.Select
End With

End Sub

Sub RMSPlot()

Dim ws As Chart 'Chart names; circumvent problem of
                'creating same plot twice
Dim plottedQuantities As String 'Used in sheet name to tell if
                                'V, I, or V&I are plotted
Dim yLabel As String 'y-axis label

```

```

Dim xaxisvalues As Integer 'Number of rows plotted
Dim lastCol As Integer 'last column in "Hidden" worksheet
                        'containing data
Dim n As Integer, m As Integer 'loop counter

'Generate name for chart sheet
If CheckBox1 = True And CheckBox2 = True Then
    plottedQuantities = " (VI)"
ElseIf CheckBox1 = True Then
    plottedQuantities = " (V)"
ElseIf CheckBox2 = True Then
    plottedQuantities = " (I)"
End If

plotSheetName = Worksheets("Hidden").
                Range("I" & plotFileIndex + 1) & "-" & Worksheets("Hidden")
                .Range("K" & plotFileIndex + 1) & ", RMS" & plottedQuantities

'Ensure this plot has not already been generated
For Each ws In ThisWorkbook.Charts
    If ws.Name = plotSheetName Then
        MsgBox "A plot of the RMS values for " &
                Worksheets("Hidden").Range("J" & plotFileIndex + 1).Value &
                " - Disturbance " & Worksheets("Hidden")
                .Range("K" & plotFileIndex + 1)
                & plottedQuantities & " has already been generated."
        Exit Sub
    End If
Next ws

'Rename chart sheet
Charts.Add.Name = plotSheetName
ActiveSheet.Move After:=Sheets(Sheets.Count)

'Select chart type, select data to plot,
'LABEL MY AXES (Thanks Dr. G!)
'Rename Series
With ActiveChart
    .PlotArea.Select
    .ChartType = xlLine

    'Select source data for graph
    .SetSourceData Source:=Sheets(dataSheetName).
                    Range(plotStartCol & plotStartRow, plotEndCol & plotEndRow)

    'Rename labels in legend of chart
    If CheckBox1.Value = True And CheckBox2.Value = True Then
        .SeriesCollection(1).Name = """"Va""""
        .SeriesCollection(2).Name = """"Vb""""
        .SeriesCollection(3).Name = """"Vc""""
        .SeriesCollection(4).Name = """"Ia""""
        .SeriesCollection(5).Name = """"Ib""""
        .SeriesCollection(6).Name = """"Ic""""
        yLabel = "Voltage or Current Magnitude"
        seriesCount = 6
    ElseIf CheckBox1.Value = True Then
        .SeriesCollection(1).Name = """"Va""""

```



```

        .SeriesCollection(2).Name = ""Vb""
        .SeriesCollection(3).Name = ""Vc""
        yLabel = "Voltage Magnitude"
        seriesCount = 3
    ElseIf CheckBox2 = True Then
        .SeriesCollection(1).Name = ""Ia""
        .SeriesCollection(2).Name = ""Ib""
        .SeriesCollection(3).Name = ""Ic""
        yLabel = "Current Magnitude"
        seriesCount = 3
    End If

    'Label axes
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text _
        = "Cycles at 60 Hz"
    .Axes(xlValue, xlPrimary).HasTitle = True
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = _
        yLabel

    .PageSetup.LeftFooter = Worksheets(dataSheetName).Range("A1")
    .PageSetup.RightFooter = "&D &T"
End With

'Adjust x-axis values
xaxisvalues = plotEndRow - plotStartRow + 1
lastCol = Sheets("Hidden").Cells(1, Worksheets("Hidden"). _
    Columns.Count).End(xlToLeft).Column

'Print first x-axis value to "Hidden"
Worksheets("Hidden").Cells(1, lastCol + 1) = (-1 * preCycles)

'Calculate and print subsequent x-axis values to "Hidden"
For n = 1 To (xaxisvalues - 1)
    Worksheets("Hidden").Cells(n + 1, lastCol + 1) = _
        Worksheets("Hidden").Cells(n, lastCol + 1).Value + 0.5
Next n

'Format chart
With ActiveChart
    For m = 1 To seriesCount
        .SeriesCollection(m).XValues = Worksheets("Hidden") _
            .Columns(lastCol + 1)
    Next m
    .Axes(xlCategory).TickMarkSpacing = 1
    .Axes(xlCategory).TickLabelSpacing = 1
    .Axes(xlCategory).AxisBetweenCategories = False
    .Axes(xlCategory).TickLabelPosition = xlLow
    .SetElement(msoElementChartTitleAboveChart)
    .ChartTitle.Text = Worksheets("Hidden"). _
        Range("J" & plotFileIndex + 1) & " - Disturbance " & _
        Worksheets("Hidden").Range("K" & plotFileIndex + 1) _
        & Chr(13) & "(RMS Values)"
    .ChartArea.Select
End With

End Sub

```

Module 5 (Call View Raw Data UserForm)

Option Explicit

```
Sub ViewRawData()  
  
    If IsEmpty(Worksheets("Hidden").Range("I1")) Then  
        MsgBox ("Please run the program first.")  
    Else  
        UserForm2.Show  
    End If  
  
End Sub
```

UserForm2 (View Raw Data UserForm, see Fig. 8)

Option Explicit

```
    Dim listBoxCount As Integer 'Number of items to  
                                'display in listbox  
    Dim viewFileIndex As Integer 'File to Unhide  
    Dim sheetName As String  
  
Private Sub CancelButton_Click()  
    Unload UserForm2  
End Sub  
  
Private Sub Frame1_Click()  
  
End Sub  
  
Private Sub OKButton_Click()  
  
    'Cycle through files selected for viewing  
    For viewFileIndex = 0 To ListBox1.ListCount - 1  
  
        If ListBox1.Selected(viewFileIndex) Then  
  
            sheetName = Worksheets("Hidden").  
                Range("I" & (viewFileIndex + 1))  
            Worksheets(sheetName).Visible = True  
  
        End If  
  
    Next viewFileIndex  
  
    'Tell user if nothing was selected.  
    If sheetName = "" Then  
        MsgBox "No file was selected."  
    Else  
        Worksheets(sheetName).Select  
    End If  
  
    Unload UserForm2  
  
End Sub
```

```

Sub UserForm_Initialize()

    Dim i As Integer 'loop counter

    listBoxCount = Application.WorksheetFunction. _
        CountA(Worksheets("Hidden").Range("I:I"))

    'Fill the list box
    With ListBox1

        For i = 1 To listBoxCount
            .AddItem Worksheets("Hidden").Range("J" & i) & _
                " - " & Worksheets("Hidden").Range("K" & i)
        Next i

    End With

    'Select the first list item
    ListBox1.ListIndex = 0

End Sub

```

Module 6 (Help Button)

```
Option Explicit

Sub ViewHelpFile()

Dim help As Object

    Application.ScreenUpdating = False
    Set help = ThisWorkbook.Worksheets("Hidden").OLEObjects(1)
    help.Verb Verb:=xlPrimary
    Worksheets("Hidden").Visible = False
    Worksheets("Home").Activate

End Sub
```