



Virtual Lab 4: Handling and Displaying Data

By: Steve Badelt and Daniel D. Stancil
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA

Purpose:

- Front Panel User Interfaces
- File I/O with Agilent VEE
- Porting to MATLAB
- Completion of the improved Bode plot generator
- Optimizing Agilent VEE for speed

Equipment:

- Agilent 34401A Multimeter
- Agilent 54601A Oscilloscope with Agilent 54657A Measurement/Storage Module
- Agilent 8116A Pulse/Function Generator or Agilent 33120A Function/Arb Generator
- Agilent 6236B Triple-output Power Supply
- Connectix Video Camera
- Intel 100MHz Pentium computer, 32MB Ram with GPIB Card
- Agilent VEE for Windows software, Version 3.2
- Timbuktu (remote access) software
- Text: Robert Helsel, "Graphical Programming: A Tutorial for HP VEE", Prentice Hall, 1995
- [Download Agilent VEE executable files](#)

Introduction:

In this lab, we will be completing an improved Bode plot measurement program using the functions you created in Lab 3. The new program will have data saving features and an easy to use control panel for finishing touches.

File I/O:

The **I/O→To→File** object may be used to save data in a variety of formats. Inputs to the object (i.e., your data) can be saved to the file, and any string may be added to sign or timestamp the file. On initialization of To File, the option exists to place the write pointer at the beginning or end of the file. Therefore, data may be placed in a new file or appended to an old file. The file to be saved or modified may be set by clicking on the filename button. However, a more sophisticated way is to use an object that prompts the user with the standard Windows file dialog (**Data→Dialog Box→File Name Selection**). Upon completion of your program, AgilentVEE closes any files it was using automatically.

Any array of data may be read into MATLAB using the Load function. The syntax,

```
load xxxxxxxx.yyy
```

loads an m line, n value per line file into an m-by-n array.

The Front Panel:

In addition to user objects, there is another technique for reducing the clutter of wires on your workspace. Hide it! At any point during the programming process, the user may select an object and choose **Edit→Add to Panel**. The first time this action is taken, two buttons will appear in the Toolbar, **Detail** and **Panel**. One of these is always active. If Detail is depressed, all the wiring between components is displayed. When Panel is active, those objects added to



the front panel in the manner described above will be displayed without the connecting wires. Within the panel view, background graphics and labels may be chosen to accentuate objects and provide an easy user interface for your homemade "virtual instrument."

Procedure:

Part 1: Saving Data

1. Place **Flow**→**Repeat**→**For Count** on the workspace. This repeats whatever thread is connected to its output N times, outputting numbers sequentially from 0 to N-1.
2. Wire the output of the For Count object to both **Math**→**Power**→**sq()** and **Math**→**Power**→**sqrt()**.
3. Choose **I/O**→**To**→**File**. Add a data input (The assigned name "b" will be fine.) and select **Add**→**Control Input(File Name)**. A control input is different from a data input in function only. Each must be supplied for an object to execute. A control input is simply geared towards how an object operates, as compared to on what it operates, as with a data input.
4. Wire the result of $sq(x)$ to A, and the result of $sqrt(x)$ to B.
5. Select **Data**→**Dialog Box**→**File Name Selection** from the menu. Wire its File Name output to the File Name input of the To File object. Choose "Writing" in the "Select File for:" box in the File Name Selection object.
6. Double click on the first line of the To File transaction list (it should be highlighted). In the appropriate fields, select WRITE, TEXT, a, REAL FORMAT, and STANDARD. Turn the EOL (end of line) button off so that more data may be added to each line of the file after the A input.
7. Add another transaction. Choose WRITE, TEXT, "\t" (with quotation marks), and STRING FORMAT. Again switch the EOL button to its off position.
8. Now add the final transaction. Repeat the transaction for writing the variable A, but this time enter the variable B, and ensure that EOL is turned on. This will be the last entry in each file line.
9. Now choose **Run**, and save the data to any file you choose. Ensure the data is in the appropriate two-column format by using a text editor such as the Windows accessory Notepad.
10. Save your Agilent VEE program to lab4pt1.vee.

Part 2: The Complete Bode Plot

Using the user objects developed in Lab 3, complete the improved Bode plot generator. You can cut and paste between workspaces just like in most other Windows programs. Test the program to guarantee it works. Channel 1 of the oscilloscope should be attached to a filter input, and Channel 2 should be monitoring the output. You should be able to set the input voltage, range of frequency sweep, and points per decade. The output should consist of two graphs: magnitude in dB, and phase in degrees. Both should be plotted against frequency on a log scale. Use Save As to save your program as lab4pt2.vee. Note: To obtain the frequency for your plots and the time base scale, you can either use the frequency values from the frequency sweep object, or read the frequency from the oscilloscope. Which do you think would be faster? Which do you think would be most accurate?



Part 3: Making it look good.

Individually select both graphs, the Input Voltage, and the Frequency Sweep objects on the workspace. For each choose **Edit**→**Add to Panel**. Then press the Panel view button from the toolbar. Arrange, resize, and color the objects on the front panel in a manner that is easy to read and easy to look at for extended periods of time. Choose **File**→**Edit Properties** while in panel view to modify the properties of the panel workspace. Don't forget to save your work!

Part 4: Optimizing for Speed

The execution speed of an Agilent VEE program is determined by three basic things: the time it takes the instrument to take the data, the time it takes to transfer the data to/from the instrument, and the execution time of the Agilent VEE program itself. We only have control over the speed of the Agilent VEE program in our labs, whereas the real bottleneck is usually the instrument response time. However, when program execution time is a factor, several things can be done to speed things up .

You can monitor the execution time with the **Device**→**Timer** object. This object measures the elapsed time in seconds between receiving signals on its two input terminals (upper terminal is start, lower terminal is stop).

1. Connect a **Flow**→**Start** object to the sequence-in pin of the For Range object in the Bode program (the For Range object controls the frequency range of the plot).
2. Select a **Device**→**Timer** object and connect the Start object to the upper input, and the sequence-out pin of the For Range object to the Timer stop input.
3. Run the program to determine the execution time.
4. Optimize your program to minimize execution time. Three specific techniques for reducing execution time are listed below. Additional ideas can be found in Chapter 11 of **Graphical Programming: A Tutorial for Agilent VEE**. Be sure to save your optimized program. Key techniques for reducing execution time are:
 - Reduce the number of objects in your program
 - Show objects as icons whenever possible
 - Manipulate data in arrays whenever possible (the original program performed calculations and plotted each point separately)

You should be able to:

- Demonstrate writing to a file using your program lab4pt1.vee (1.10).
- Demonstrate the optimized program saved in part 4 (lab4pt2.vee).