Agilent Technologies

EducatorsCorner.com Experiments

Virtual Lab 2: Agilent VEE & Timbuktu

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

Purpose:

- Introduction to Agilent VEE & Timbuktu
- Computer data acquisition from instruments over GPIB with Agilent VEE
- Computer control of instrument settings and features over GPIB with Agilent VEE
- Design of User Objects in Agilent VEE
- Automation of data acquisition tasks via Agilent VEE
- Completion of a Bode plot magnitude generator

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
- HP VEE for Windows software, Version 3.2
- Timbuktu (remote access) software
- QuickPICT software
- Text: Robert Helsel, "Graphical Programming: A Tutorial for HP VEE", Prentice Hall, 1995
- Download Agilent VEE executable files

Introduction:

Timbuktu

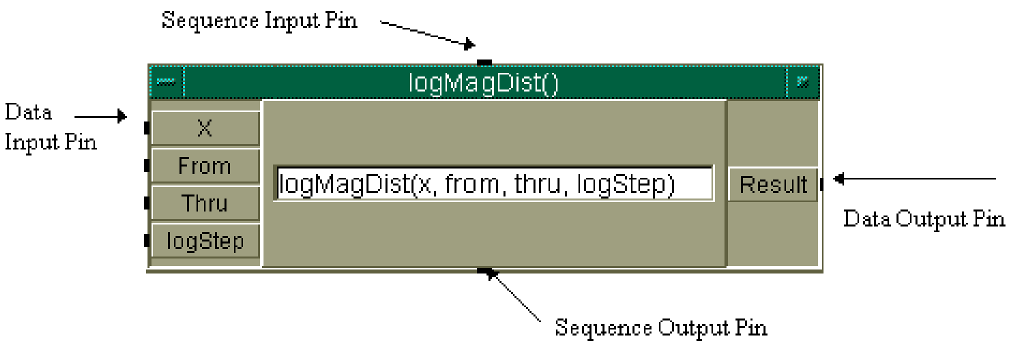
Timbuktu Pro (Farallon Computing, Inc.) is a program used to control another computer remotely, just as if you were at its keyboard and looking at its screen. Many server programs (various x-servers, etc.) allow you to connect to another computer and use its computing resources. Timbuktu adds the capability to manipulate resources connected to the computer you are controlling. However, Timbuktu is designed for the remote control of a computer, not the sharing of its resources. So when you connect to that computer, if anyone else is using it, you will be fighting over its control!

When you connect to a computer via Timbuktu Pro, a window will open on your local computer's screen. Inside this is a duplicate of the remote computer's screen. Everything happening on this duplicate screen is really happening! You can open applications, control instruments, and even turn off Timbuktu Pro on that computer... which essentially disconnects you from it permanently!

Note: *If you attempt to open a DOS prompt, it will bring up an icon if you do not bring it up in a window. Timbuktu is exclusively a Windows application. Even if you double click on the DOS prompt Icon, it won't open. Try selecting the icon and typing Alt-Enter to bring a DOS prompt up in a window.*

Agilent VEE

The Agilent Technologies Visual Engineering Environment (Agilent VEE) is a graphical programming environment optimized to reduce test development time for instrument control. In a conventional textual language, rules of syntax and operators control program execution. In Agilent VEE, the program is designed by connecting objects in the workspace together with "wires", lines that connect these objects in a meaningful fashion. These wires dictate when and where data will flow through the program (propagation).



Each object acts like a function call which manipulates certain arguments. It receives this data via a pin on the left side. The object performs some function to this data and then returns any results through pins on the right hand side. Each pin is for a specific input/output to the function, though that data may be an array or record of information. Data propagation through a network of objects, analogous to the way a variable passes through functions in C, is determined by the status of each object's pins. Only an object with data supplied to all its **Data In** pins will execute. Thus the objects with no data input pins execute first in the workspace. Their **Data Out** pins provide inputs to other objects, then those objects may execute, etc.

Notice that multiple objects may have the ability to execute at the same time. The determination of which executes first is indeterminate. However, the programmer may control this propagation by using sequence pins. An object with its **Sequence In** pin wired to something else may not execute until data (of any type) is passed to that pin. Then the object may execute, and it is not until this execution is complete that the **Sequence Out** pin sends its data out, if it is wired up. Consequently, the Sequence pins are acting just like Data In/Out pins, except the data isn't being used for anything.

Agilent VEE has a menu bar just as every other Windows based application. To designate a menu command, the sequence of menu levels to choose from will be designated just like the button choices for the instruments in Lab 1 (e.g. **File** → **Open**). Just below the menubar is a tool bar with the Agilent logo on it. Buttons located on the toolbar, used to execute your programs (e.g. **Run, Stop**), will be designated in a similar manner. The left mouse button is used to make choices and perform actions. The right mouse button will open a pop up menu for anything on the screen. The menu displayed when you click the right button over an object, its associated **Object Menu**, is one you will be seeing frequently.

Component Drivers

There are three ways to access the information on the Agilent equipment via Agilent VEE. All are obtained by choosing **I/O** → **Instrument** and the appropriate options. The first method of reading the data on an instrument is with its accompanying software **Instrument Panel**. Loading this driver brings up a colorful object, a version of the instrument's front panel. It is not an exact replica, but it takes advantage of any functions the instrument has and displays them on the screen.

Component Drivers are the easiest and most efficient tool for obtaining data from the instruments. Unlike the Instrument Panels, component drivers only take readings of a single type. Instrument Panels often take many readings of data your application does not require. After loading a component driver, inputs and outputs may be added to the resulting object as desired. The appropriate options are located in the pop-up menu displayed after clicking the right hand mouse button while over the object. The third option for obtaining information from instruments will be discussed in a later lab.

User Objects

You may have already imagined that using a graphical programming language can take up lots of screen space unless there is some way to compress it all. Similar to user defined functions in C, **`User Objects`** allow compression of graphical code into a small screen space. A user object is basically an extension of the workspace contained in a window or box that can be minimized or maximized. This sub-workspace may have data inputs and outputs, just like C functions, and also possesses sequencing pins to control execution.

Loops

For, if, and while statements are all available in the **Flow** → **Repeat** menu. For more detailed information on the use of each type, use the help library. There is one important note for using loop structures, however. The data output pin, whether it is connected to a data input or a sequence input, is wired to the beginning of whichever thread you wish to execute. The sequence out pin is used to signal that the entire set of repetitions has completed execution.

dB

The dB values calculated by the multimeter are relative to a reference value. You may recall that in Lab 1 we took an initial measurement to set the reference. In this lab, it will be convenient to calculate and set the reference explicitly. The tricky part of this calculation is that the reference value and subsequent readings are taken as dBm, the power delivered to a resistance referenced to 1mW.

$$dBm = 10 \times \log\left(\frac{reading(rms)^2}{R_{reference} \times 1mW}\right)$$

Once these readings are taken, the calculation for dB is simply:

$$dB = reading\ in\ dBm - reference\ value\ in\ dBm$$

The default resistance used by the multimeter is $R_{reference}$=600 ohms.

Procedure:

Part 1: Computer acquisition of data

Note: Once you start remote operations with the instruments, both the multimeter and the function generator will disable their front panels. To re-enable them, press **`Shift`** *on the multimeter or* **`LCL`** *on the function generator.*

- Hook up the function generator to the multimeter directly.
- Select **`I/O`** → **`Instrument(dmm)`** → **`Instrument Type(Component Driver)Get`** → **`Instr`**. Place it anywhere in the workspace. Do you notice any changes in the display on the multimeter front panel?

- Open the multimeter component driver's object menu with the right mouse button. Choose **Add Terminal** → **Data Output**. Choose **Reading** from the menu.
- Wire up an AlphaNumeric display to the reading output. Press **run** a few times. Notice how often and when the multimeter takes its readings when it is in remote mode. What quantity are we measuring anyway? Try changing this quantity using the multimeter front panel. Can you? Also note that your AlphaNumeric display may show a different number of significant figures than the multimeter screen! Double click on the AlphaNumeric title bar to edit the object properties. Choose **Number** → **Global(Off)** → **Significant Digits(12)** → **OK**. Then look at the data representation. Are there 12 significant figures to display?
- We want to measure frequency, but the multimeter is not configured for this. (Did you know the multimeter could be used as a frequency counter?!) To figure out how to read frequency, choose **Help** → **Instruments** and pick the help file corresponding to the multimeter's Agilent identification number (34401A). From the resulting pop-up window, choose **Components** (it's at the bottom of the list). In the resulting list of information the "Function" settings are presented. Find them and look up frequency among the options. The list of functions corresponds to their numerical order (e.g. 0,1,2...). Figure out which number frequency corresponds to and close the help dialog.
- Select **Add Terminal** → **Data Input**(**Function)**from the object menu. Notice that beside the word "Function" is the range of acceptable values corresponding to the values seen in the help file.
- Choose **Data** → **Constant** → **Integer** from the menu. Place the resulting object to the left of the multimeter object. Wire the constant to the input. Set the constant to the value you determined in step 2. Double click on the title bar of the constant and title it "Frequency." Minimize the integer constant object.
- Press **run**. What value is now displayed? Realize that every time you run this component driver, you are telling the multimeter to be set to measure frequency, regardless of whether it needs to be set again or not.
- Double click on the tool bar where it says "Untitled." Rename the Workspace "Read Frequency." Save the workspace as lab2pt1.vee.

Part 2: Automated Data Acquisition - Bode Plotter!

Remember when we used the dB function in Lab 1? The dB values are determined by comparing the present readings to the first reading taken. This first reading presents a problem if we want to take a Bode reading starting from 10Hz to 5MHz and we are testing a high pass filter! So, the following design will remotely set this reference value.

1. Select **File** → **New** to start a new workspace. Choose **Device** → **User Object**. Double click on its title bar and change the name to "Initialize FG& MM." Maximize the user object so there is more room to work.
2. Choose a Component Driver for both the function generator and multimeter in the **I/O** → **Instrument** dialog. Wire the **Sequence Out** pin of the function generator to the multimeter's **Sequence In** pin. Now the function generator will execute first.
3. Add **Amplitude** and **Waveform,** and **Output_Enable** data inputs to the function generator. Add **Function**, **Math**, and **dB_Ref** data inputs to the multimeter (in that order). Hook up the following constants to their respective inputs:

Constant	Value	Title	Wired to
Real	10.0	Vp	Amplitude
Integer	1	Sine Wave	Waveform
Integer	1	Output_Enable	Output_Enable
Integer	1	VAC	Function
Integer	2	dB	Math

4. Select **Math** → **Formula**, and place this object beside the **dB_Ref** input. The formula box performs whatever mathematical expression you enter. Double click on the "A" input and rename it "Vp." Then type into the text block the formula "10*log10(sq(Vpp)/(2*600*.001))". (Can you explain this formula?) Hook up the Vp input on the object to the Vp constant, and wire the result to the **dB_Ref** input. Name this object "dBm calc."
5. Minimize the user object "Initialize FG & MM".
6. Select **Flow** → **Repeat** → **For Log Range** from the menu bar and place it below the minimized user object. Wire the sequence pins so that the initialization takes place before the loop.
7. Place a function generator component driver on the desktop with a **Frequency** input and wire this to the data output pin of the for loop. This will allow us to cycle through a range of frequencies automatically. Place a multimeter component driver below the function generator with a **Reading** output. Wire the sequencing pins so the function generator executes first.
8. Choose **Display** → **X vs. Y Plot** form the menu. Wire the for loop output controlling the frequency to the **Xdata** input. Wire the multimeter reading to the **Ydata** input. Enter the Edit Properties dialog and change the title to "Magnitude." Switch the checkbox for **Show Terminals(off).** Then hit **Traces and Scales**. Within the pop up dialog, switch the X axis to **Log**. Change the x axis label to "frequency," and the y axis label to "dB." You may also change the colors, etc., to your preference.
9. Set up the low-pass filter from Lab 1 and hook the multimeter up to its output.
10. Click **Run** and see what happens. Try changing the for loop settings for range and resolution. To view the data on the magnitude plot, try hitting **Autoscale**. Within the right button pop-up menu for the plot there are also several zoom functions. To see the data point values, edit the property dialog for **Markers** and set it to 2. How far apart are the values for -3dB and -20dB?
11. The multimeter is not designed to take VAC data beyond 300kHz. Attempt to take data beyond this and see what the effects are. At 1MHz the data may be off as much as 30%, according to Agilent documentation.
12. Double click on the tool bar where it says "Untitled." Rename the Workspace "Bode-Magnitude." Save the workspace as lab2pt2.vee.

Note: If your program seems to be working except that your dB values are offset, check the reference resistance setting of the multimeter. Although the default is 600 ohms, the previous user may have set it to something else! You may want to set $R_{reference}$ to 600 ohms in your initialization object just to make sure.

You should be able to:

- Demonstrate your working Agilent VEE program saved as lab2pt1.vee (1.9).
- Demonstrate your working Agilent VEE program saved as lab2pt2.vee (2.12).