

Agilent 81200 How to Use the 81200 Data Generator/Analyzer Platform together with Agilent VEE for Signal Integrity Analysis

Product Note

Introduction

As designs get faster and more complex, signal integrity is becoming increasingly important within the design validation process.

There are two main issues with signal integrity, these being waveform and jitter. Traditionally, this is an analog instrument domain, for products such as scopes.

At a first glance, it sounds contradictory to use digital equipment to perform analog type measurements, but we intend to prove otherwise. This product note explains how waveform measurements can be performed using Agilent Technologies' 81200 data generator/analyzer together with Agilent VEE.



Besides the basics, this product note will discuss, using an example program, how an output waveform (similar to that of a scope shot) is reconstructed on the 81200 with the help of Agilent VEE. This reconstruction is referred to as "soft scope" later in the document.

This method is not targeted to replace the traditional usage of an oscilloscope, as it still performs this type of measurement much faster. Instead, VEE provides a more convenient way to achieve the same results, without the need to integrate a scope into the measurement setup, thus saving time, money, and reducing setup complexity.

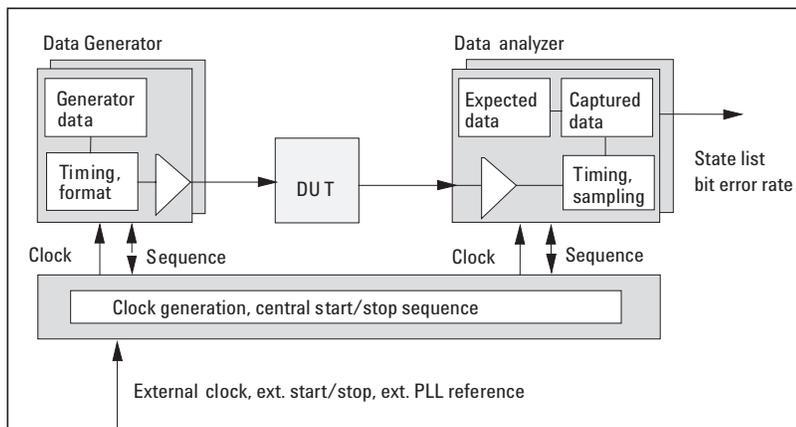
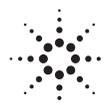


Figure 1: Block Diagram of the Agilent 81200



Why this is possible

The Agilent 81200 data generator/analyzer platform provides the following prerequisites to carry out signal analysis together with VEE:

Its principal task is to perform a functional test. This means that at the same time it sends out, it receives a pattern from the device-under-test (DUT), which is compared on the fly. Every channel runs synchronously, as the block diagram in figure 1 demonstrates. The result of the functional test is PASS or FAIL.

With the help of VEE, an automated test flow can be achieved where one or more parameters (timing, levels, threshold) can be changed within the interest range. The program is

responsible for setting the appropriate parameters, initializing the test, reading back the error result, and generating the graph with the waveform trace. An example of such a graph is given in figure 2; it looks very similar to a scope waveform. In order for this to make sense and to ensure that the result is accurate and reliable, the test system receiver input must meet the following requirements:

- High input sensitivity, which means it can detect an input signal with a small amplitude. Only that would allow moving along the input signal transition from 10 to 90% and at the same time still detect if the input signal is higher or lower than the given threshold.

- Sufficient timing accuracy and resolution as well as threshold accuracy, for a precise sample of the input signal.

Using VEE makes it possible to generate a two-dimensional parameter variation within an automated test flow.

The 81200 Data Generator/Analyzer Platform standardizes interfaces (GPIB) to integrate other programmable instruments. In this example, a power supply for controlling the DUT's Vcc is included. Other instruments can be integrated, in order to build up just enough performance and functionality needed for an analysis task.

Built in Controller Windows NT C/C++ program as executable

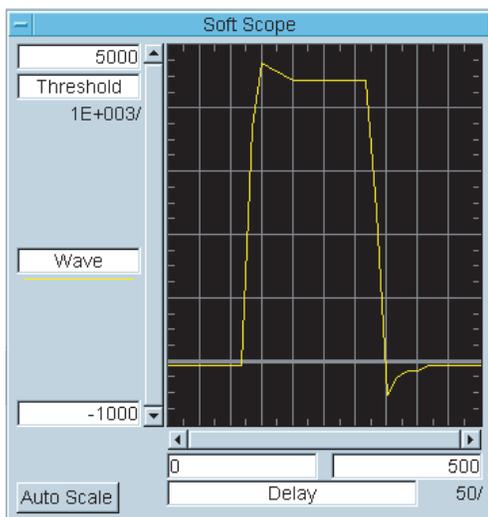


Figure 2: Soft Scope waveform graph result

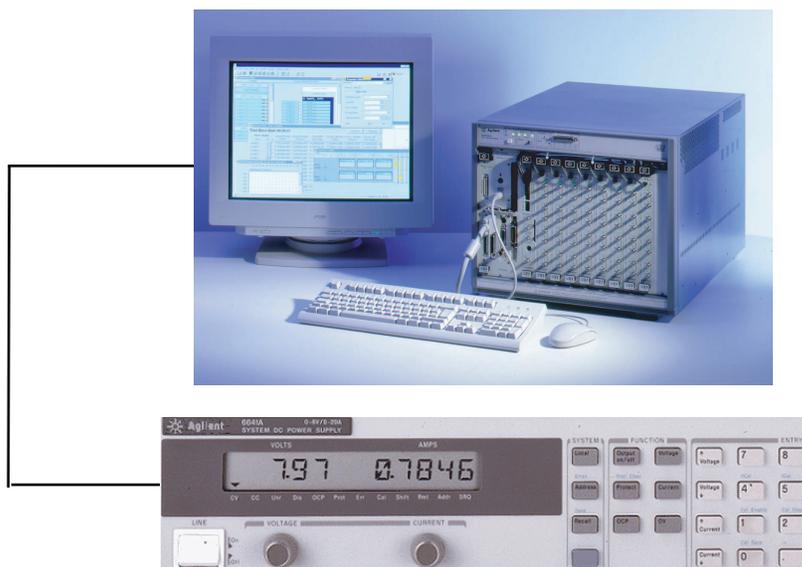


Figure 3: Setup with controller and GPIB power Supply

How it works

To perform a functional test, it is necessary to have a test pattern together with timing and level/threshold information. The test pattern has a part for stimulation and another part for comparing the received bits, referred to as expected data. Timing and level information define how the signals are applied, whereas the analyzer will sample the received signal in point of time and level. With the sampling performed at the proper point, the device output data matches the expected data. In this case, no errors are recorded during the test. Any mismatch of DUT output data will result in errors being recorded.

To perform a waveform reconstruction, a simple pattern for the expected data is used. This contains a single 1; all other bits are set to "don't care." With this applied, the left part of figure 4 shows that an error occurs when the sampling point is set to point A or any other point above the waveform. No error occurs when the threshold is set to point B or any other point below the waveform. This is used to find the closest point along the input signal.

The automated search is performed in the way that the threshold is lowered when errors occur, and raised when no errors occur. This is shown in the right part of figure 4. The binary search method is then applied. The approximation starts with two points 1 and 2. They are known as below and above the waveform. At the lower end, no errors are expected, but at the upper end, errors are expected. The next test takes place at point 3 located in the middle. Depending on the error result, the next test point 4 will be either in between 1 and 3 (if there are no errors) or between 2 and 3 (if errors have been found). The points 5 and 6 will be obtained in the same way.

Each time the interval is divided by a factor of two, the approximation gets closer and closer to the input waveform. It does not make sense to run it infinitely, a practical place to stop would be when the new calculated interval falls below the DC accuracy of the analyzer input.

The approximation is performed at every sampling point delay, which is controlled by a loop running from a start to stop delay value, which has a given number for increment. The latest threshold point found error-free will be stored at every delay point. When all tests have been completed, the stored points are plotted and joined on the graph to produce the soft scope waveform. The binary search method is much faster than a linear change of threshold. Using a 5V window and 20mV accuracy, the linear change would require 250 tests, while the algorithm is completed after just 10 tests.

The Result

The test device is a HCT 191 counter device. The test program toggles one of its outputs and the soft scope method is applied. The result is displayed in the graph provided in figure 2. As a reference, the same output is checked with a regular scope (Agilent Infiniium, 1.5 GHz bandwidth). This picture is provided in figure 5.

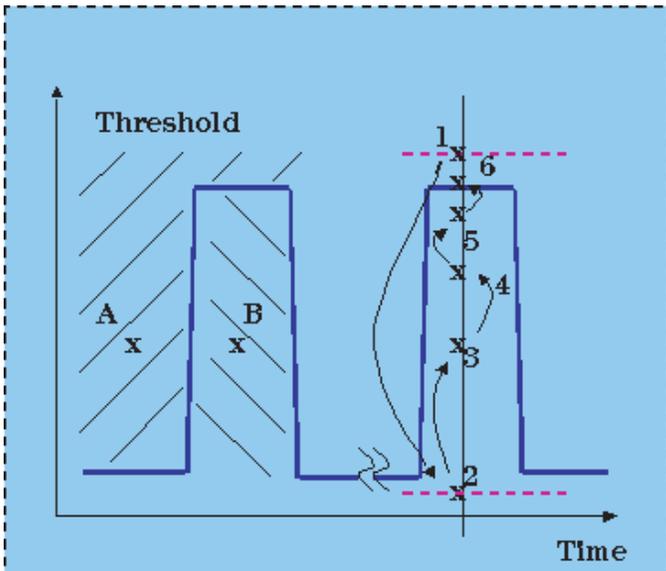


Figure 4: Error Zones and approximation method



Figure 5: Analog scope shot

VEE program

The following section explains how the Agilent VEE program provides the soft scope method. VEE is loaded onto the 81200 controller and runs the program `soft_scp.vee`. Figure 6 shows the main screen of this program.

The panel given is actually the view when a run has already been completed. After loading, all fields will be empty.

The elements on the Main panel are:

- start button
- system identification
- 6 input values defining measurement range
- waveform graph
- status indication.

Click on the start button to start the program. First the system resources are queried and displayed in the response fields. This is not necessary for the soft scope measurement; it just provides assurance that the necessary system resources are available.

The program will then ask for the range and resolution of the measurement. For every value, a default is provided. The requested values are Start Delay, Stop Delay, Delay Resolution, Start Threshold, Stop Threshold, and Threshold Resolution.

Before the measurement begins, a range check is taken. A warning will be given if no signal is found in the given range. Once a proper signal has been found, the measurement begins.

The graph is drawn after completing the test at every delay point. This lasts about 20 seconds, using the defaults for the number of delay steps and threshold resolution given. Changing this number means the time changes significantly.

Figure 7 shows the main part of the program. This is obtained by switching from the Main panel to the main program. The main program elements are now discussed in more detail:

1. "Import Library" describes the 81200 functions in the VEE program.
2. A handle is created. Default is set as `_hdl`. This is necessary to establish communication with the 81200. The handle is assigned to variable `A`.
3. The system resources are queried and displayed.

4. The setting "scope" will be loaded. This provides all the necessary settings for running the intrinsic functional test.
5. The program will ask for the range and resolution of the measurement. For every value, a default value is provided.
6. In the subroutine "check range," the program will check if a signal is found in the given range.
7. If a signal is found, the segment "loop" performs the measurement. This subroutine is also responsible for plotting the graph.

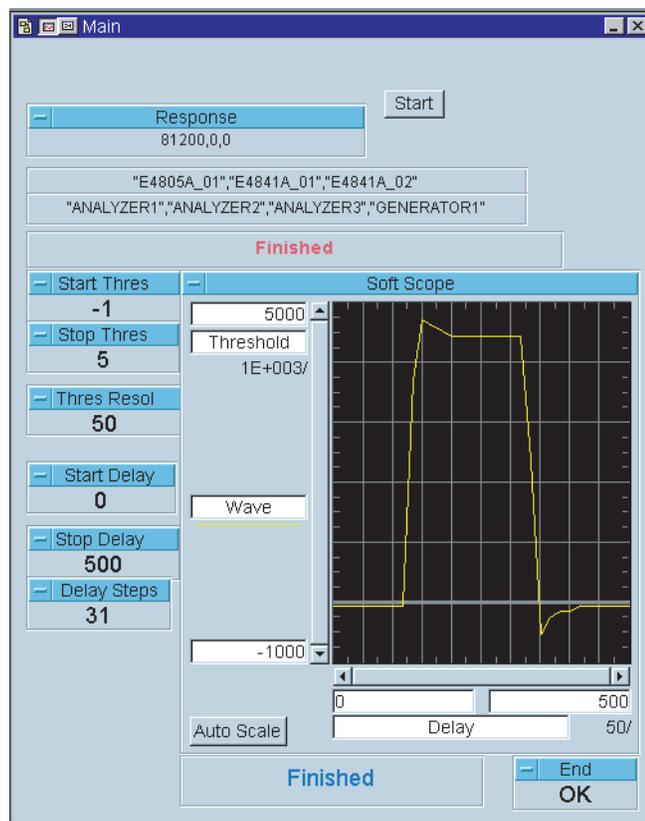


Figure 6: Main panel of the Soft Scope Program

8. When the measurement is completed, the 81200 stops, the handle is deleted, and the program disconnects to the 81200. If for any reason the program does not terminate correctly, the handle created may not be deleted. In this instance, the next run of the program would create a new handle (_hdlx). To prevent this, the Abort button can be used to execute a disconnect procedure.

9. The figure for delay resolution should always be entered as an odd number. To obtain 10 intervals, 11 measurement points including a start and stop point are required. The figure given for the threshold resolution is used for terminating the search algorithm.

The following chapter discusses further individual subroutines of the program, in more detail.

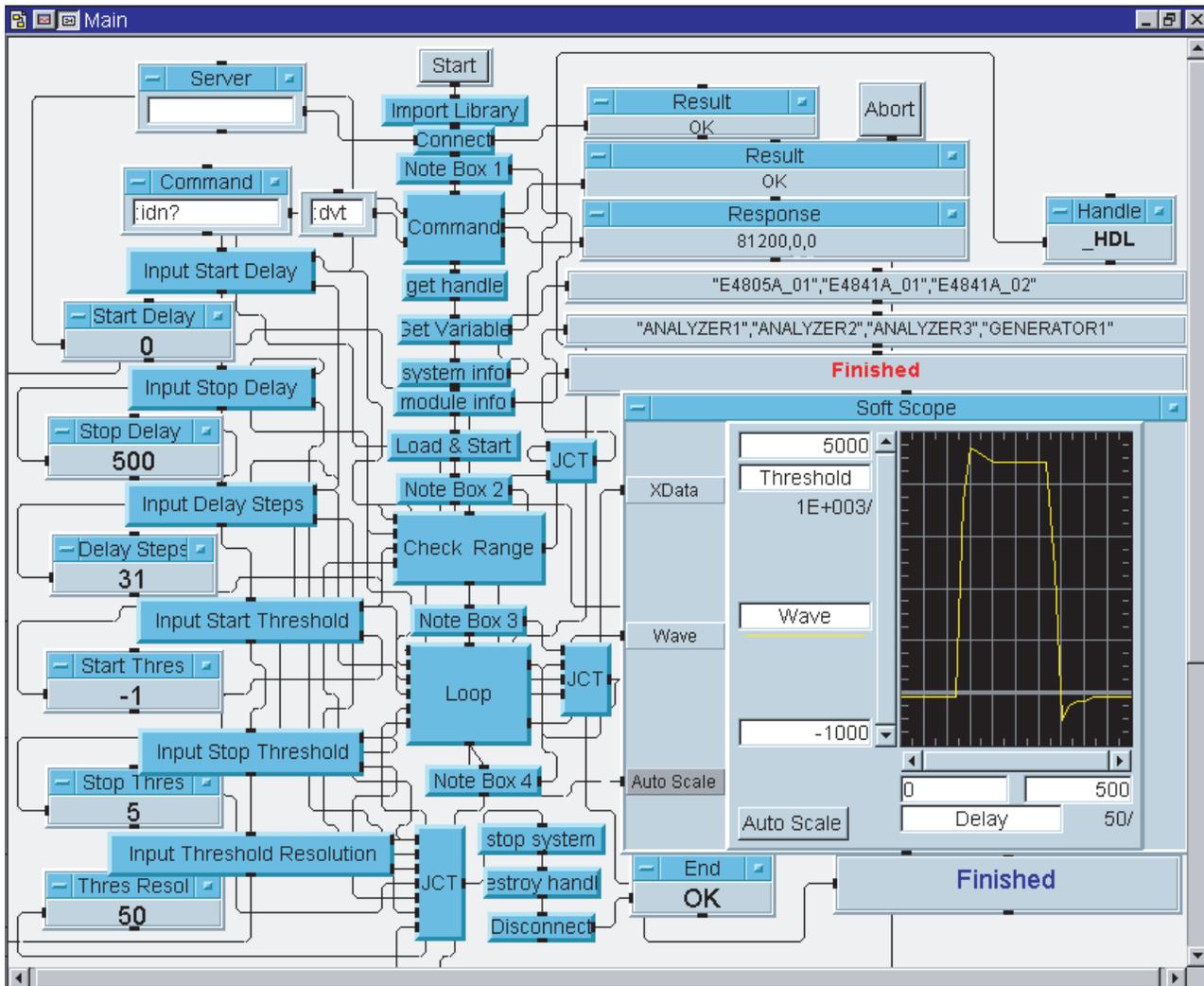


Figure7: Main part of the Soft Scope Program

Import Library

The Import Library loads the Agilent 81200 dynamic link library (dll file) and the compiled functions (h file). With that, the following compiled functions are added to VEE:

- Call
- Connect
- Disconnect
- Get Error String.

Connect

This user function calls up the 81200 and provides an error check relating to the return value (see figure 9). If no error occurs, the output will be "OK." However, if an error occurs, an error string will be provided.

If VEE runs on another system, the computer name should be entered in the server text box (see figure 7). If VEE runs on the same system, the server text box remains empty.

Command

This user function operates the compiled function Call_81200 within the Call icon.

This function needs an input from the command string and the value of handle. It will output an error string if an error occurs during transfer (RetVal). In case of a query, the data is obtained from "response." The buffer size returns a value for the total length of string. Within the "call," a default buffer with a length of 500 characters is set. If longer strings are queried (error count of a large port), this value must be adjusted.

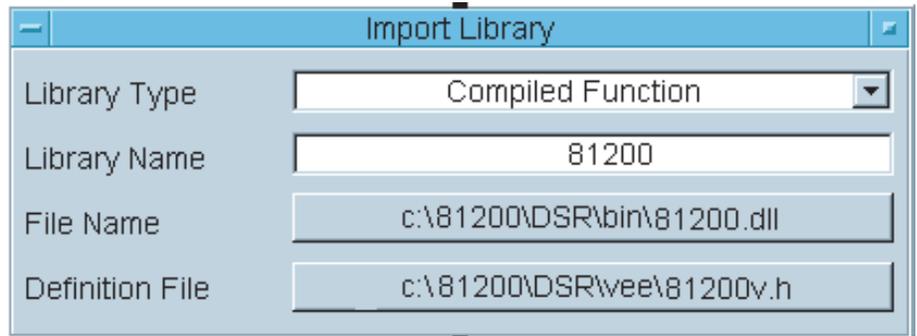


Figure 8: Imported functions from Import Library

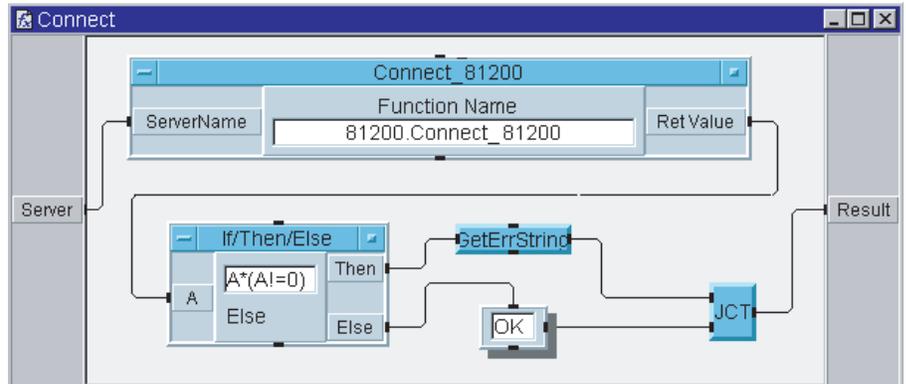


Figure 9: Connect

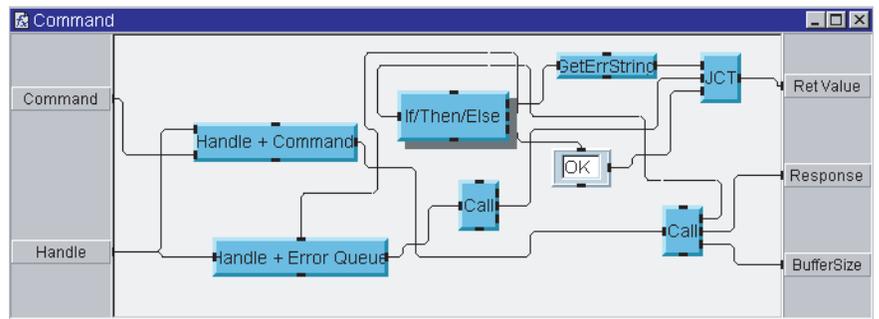


Figure 10: Command for handling Command query

Get Handle subroutine

The “handle ”is the name used to communicate with the 81200 system. Figure 11 demonstrates how this is done. This example defines the handle _hdl, which is stored in variable globalA. To create a user handle, the system default handle :dvt:inst: must be used.

Load Setting subroutine

Loading of the setting “scope ”is performed in figure 12. It is important to know that the procedure “command ”always returns an error code. As long as a programming step is successful, the error return is OK. If a problem occurs, an error string is returned which will be displayed in the “response ”field.

Check Range subroutine

A range check is performed as shown in figure 13. The input values given are used to check if the generator waveform supplied to the analyzer channel applies to them.

The check is performed in the way that four measurements are taken, with the input values given for Start and Stop Delay, and Threshold. As the program uses the expected pattern of logic 1, it is expected that the test at Stop Threshold will fail, whereas the test at Start Threshold will pass.

If that does not happen, it is assumed that no signal is found. The user will receive a message, either to increase or decrease the threshold range, as shown in the text boxes in figure 13. The program will terminate by deleting the current handle and so disconnect. If a signal is found, the program will continue.

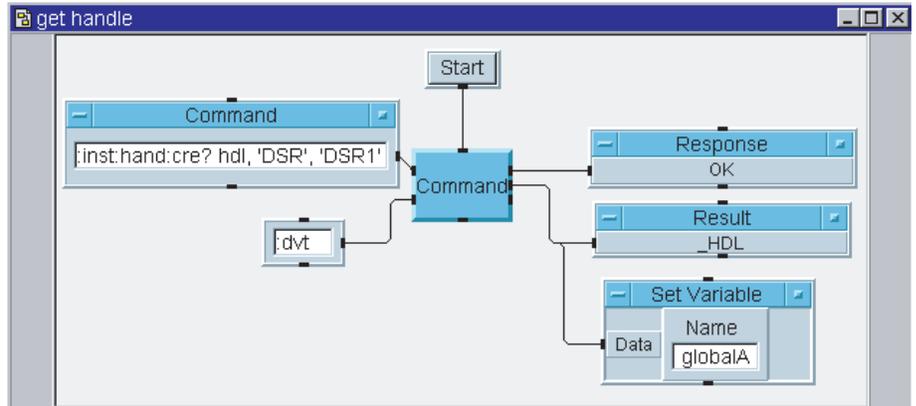


Figure 11: Get Handle subroutine

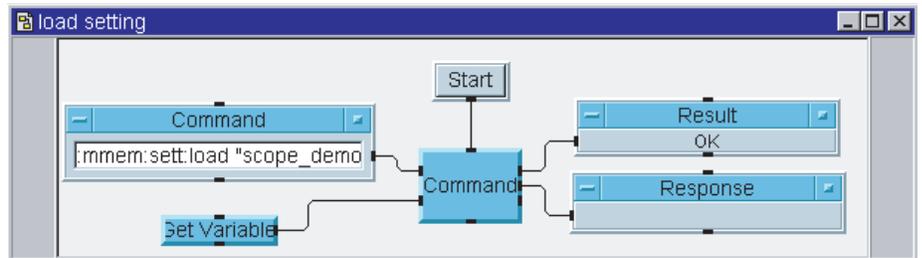


Figure 12: Load Setting subroutine

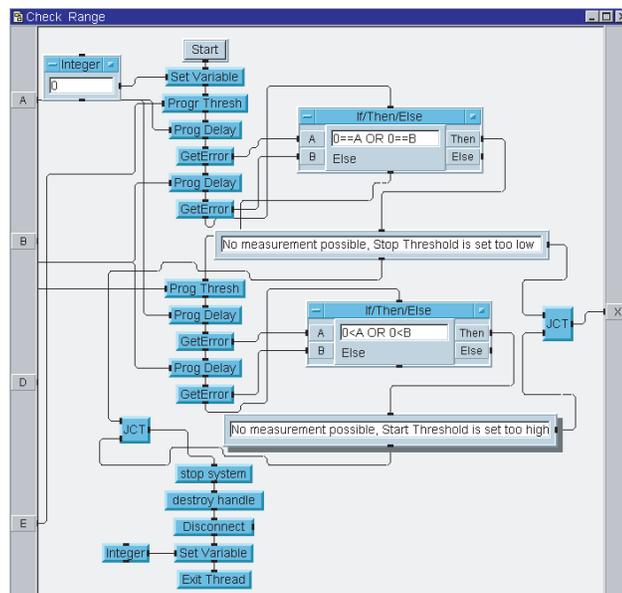


Figure 13: Check Range subroutine

Measurement Loop subroutine

The measurement loop is taken on the count of delay intervals given by the delay resolution. This happens in the Loop module, shown in figure 14. This module executes the programming for the individual sampling point delay.

The analyzer threshold is obtained from the Interval module, shown in figure 15. Delay and Threshold are stored in two arrays, which are used to form the final graph.

As the Main panel and the main program only show the graph when all tests have been completed, it may be convenient to open the

Loop subroutine while the program is running.

The preview window refreshes itself while the measurement is being taken. You are therefore able to watch in real time how the graph is progressing. A double click on the Loop block icon opens the subroutine, as shown in figure 14.

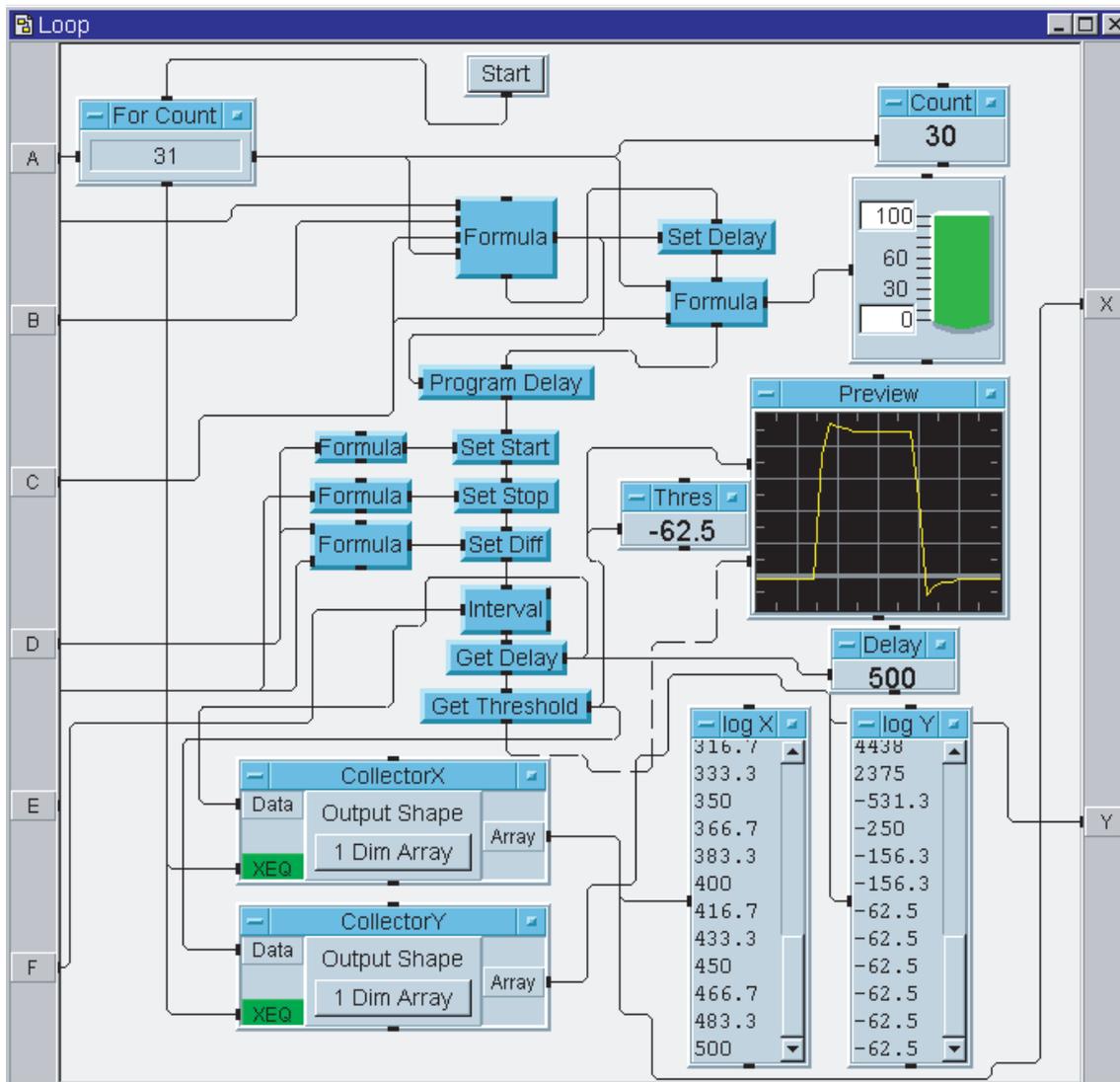


Figure 14: Loop Subroutine

Interval subroutine

A check is carried out at every delay interval point, whether the Agilent 81200 analyzer reports errors or not. Figure 15 shows the Interval subroutine, which handles the binary search algorithm. First, it programs the new midpoint threshold between the first two given start values for Start and Stop Threshold.

Then it checks for errors in the Get Error subroutine. Depending on errors, it assigns the old midpoint value to be either the new start or stop point. If no error occurs, the threshold value is stored in variable F. It then calculates the new interval and checks this value for repeating the loop or leaving the subroutine, returning the last error-free threshold with its variable.

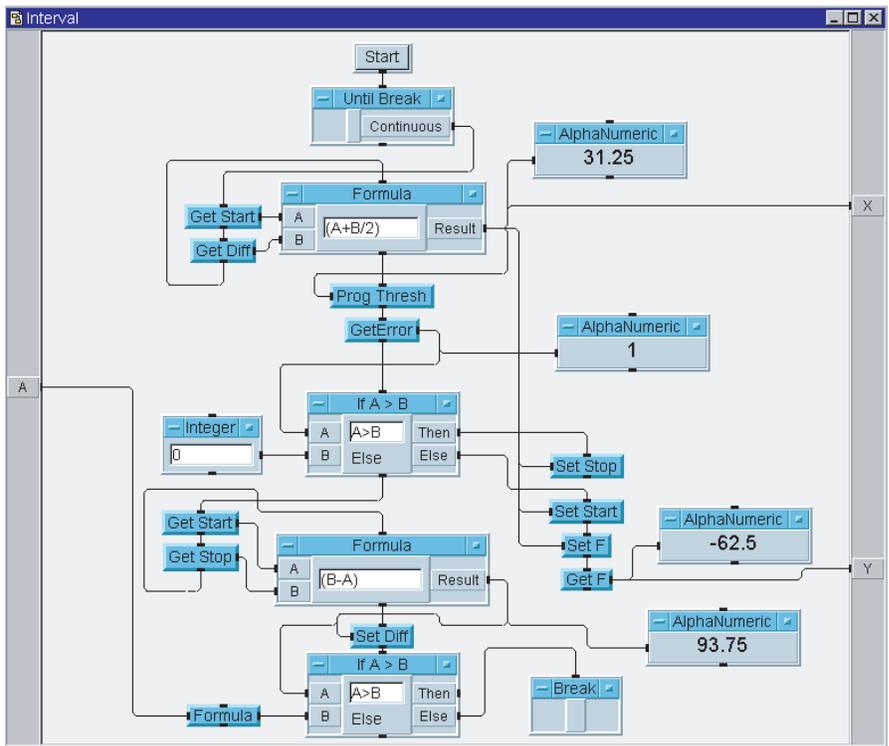


Figure 15: Interval Subroutine

Get Error subroutine

The command string queries:

- number of processed bits
- number of errors occurred.

First the string is separated. The next step is to check that the number of processed bits is greater than 0. If so, the number of errors is checked.

The variable is set for 0 should there be no errors, otherwise it is set to 1.

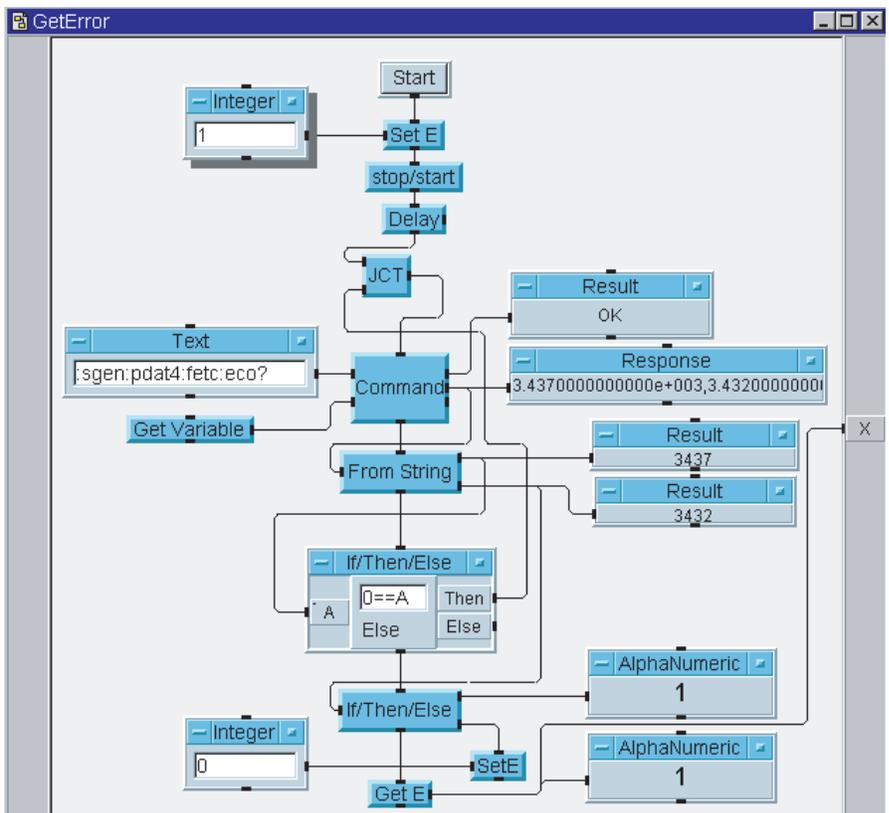


Figure 16: Get Error Subroutine

Setup

1. Install VEE on the Agilent 81200.
2. Log on as administrator (press <shift>key during boot); the following path variable has to be added to the systems environment setting: c:\81200 \DSR \bin.
3. Set up by setting "scope" on the 81200. This will be loaded from the VEE demo program. (Further details follow.)
4. Start VEE and load the program: scope.vee.
5. Connect the generator output to the receiver input.
6. A configuration of the VEE is not necessary. This happens in the program belonging to the Import Library subroutine. This subroutine imports the following library elements:
 - header: 81200v.h found in c:\81200 \DSR \vee
 - dll: 81200.dll found in c:\81200 \DSR \binThese elements are there per default.
7. Run the program.

Setting Scope

If a test device (HCT 191) is not available, then this program could be used to reconstruct a generator output signal with the analyzer. In this case, just a cable is needed to connect the generator output with the analyzer input. To do so, the following setting "scope" has to be setup manually on the 81200. Make sure the subroutine "load setting" refers to this as setting "scope"!

The setting "scope" needs to be carried out as follows:

1. Set up four ports, each with one terminal. The first port has to be a generator; the fourth port has to be an analyzer. Connect this to any system channel.
2. Select the BER measurement, with all failures enabled.
3. Setup two data subroutines with length 16. One is type 01 and contains alternate patterns of 1010..., starting with 1. The other is of type 0_x1 and contains one 1, all other bits are x, starting with the 1.
4. Assign the first pattern to the generator port, the second to the analyzer port. Set the infinite loop.
5. Set central board parameters to 10 MHz with Mux=1. All other parameters can be set as default.
6. Set the generator channel parameters for delay NRZ and 25 ns, pin mux=1, V_{hi}=1.5V, V_{lo}=0.5V, 50 W, enable output.
7. Set the analyzer channel parameters for delay 75 ns, pin mux=1, threshold=1V, 50 W, enable input.
8. Start the 81200 and check the BER measurement; it should give BER=0. Stop the 81200.
9. Save the setting as "scope."
10. Now you're ready to run the VEE example.

Summary

To use VEE on the 81200, the following is needed:
Agilent VEE on CD-Rom for Windows 95/98/NT ®: E2120F

Related Literature

	Pub. Number
Agilent 81200 Data Generator/Analyzer Platform, brochure	5968-4261E
Agilent 81200 Data Generator/Analyzer Platform data sheet	5965-3415E
Agilent 81200 Data Generator/Analyzer Platform configuration guide	,5965-3417E
Data Transfer between Design,Simulation and the Agilent 81200,product note	5967-6276E
Agilent E4839A Test Fixture,Convenient and Reliable DUT Fixture for the 81200 Platform,data sheet	5968-3580E

For more information,please visit us at:www.agilent.com/find/dvt
Windows 95/98/NT

®

is a U.S. registered trademark of Microsoft Corporation.

Agilent Technologies' Test and Measurement Support, Services, and Assistance

Agilent Technologies aims to maximize the value you receive,while minimizing your risk and problems.We strive to ensure that you get the test and measurement capabilities you paid for and obtain the support you need.Our extensive support resources and services can help you choose the right Agilent products for your applications and apply them successfully. Every instrument and system we sell has a global warranty.Support is available for at least five years beyond the production life of the product.Two concepts underlie Agilent 's overall support policy: "Our Promise "and "Your Advantage."

Our Promise

"Our Promise "means your Agilent test and measurement equipment will meet its advertised performance and functionality. When you are choosing new equipment, we will help you with product information,including realistic performance specifications and practical recommendations from experienced test engineers.When you use Agilent equipment,we can verify that it works properly,help with product operation,and provide basic measurement assistance for the use of specified capabilities,at no extra cost upon request.Many self-help tools are available.

Your Advantage

"Your Advantage "means that Agilent offers a wide range of additional expert test and measurement services,which you can purchase according to your unique technical and business needs.Solve problems efficiently and gain a competitive edge by contracting with us for calibration,extra-cost upgrades,out-of-warranty repairs,and on-site education and training,as well as design,system integration,project management,and other professional services. Experienced Agilent engineers and technicians worldwide can help you maximize your productivity,optimize the return on investment of your Agilent instruments and systems,and obtain dependable measurement accuracy for the life of those products. Get assistance with all your test and measurement needs at: www.agilent.com/find/assist Product specifications and descriptions in this document subject to change without notice.

Copyright ©1999,2000 Agilent Technologies
Printed in Germany January 28 2002
5968-3857E



Agilent Technologies