Session 2559

Preparing a Virtual Engineering Environment Laboratory Instructional Package

Thomas E. Hulbert, Robert B. Angus

Northeastern University; Boston, MA 02115

Introduction

This paper will describe the development of a process and techniques for students and technical personnel to learn and apply test and measurement systems. The courses, outlines, lessons, projects, and instructional materials were developed by two faculty members. The two of us have a combined background of more than seventy-five years of teaching and industrial experience.

During the designing and running of real manufacturing processes, and the design and analysis of experiments, the appendices in the developed book become extremely valuable to the user. The casual practitioner can quickly locate the necessary design material without having to re-learn or re-do each of the laboratory experiments. After completing the lessons, the most powerful feature of this book is its use as a reference.

The idea was conceived when a local division of a multinational corporation contacted Northeastern University's Corporate Education regarding the teaching of a data acquisition course to technicians, technologists, and design engineers. Simultaneously with this inquiry, the division was considering the adoption of the Hewlett Packard (HP) software package, HPVEE, referred to as VEE in this paper.

The entire group (division members and corporate-education personnel) met and decided to develop and offer an on-site course that included the fundamentals of data acquisition systems and the application of VEE to data acquisition and analysis. In order to gain a greater understanding, Robert Angus and a senior design engineer from the company attended a four-day seminar on VEE and its applications.

The VEE seminar, plus the ongoing discussions with the corporation's division mentioned previously, led to interesting conclusions regarding course development. Both the data-acquisition theory and the VEE applications required course development to meet the company's requirements and the students' background. This paper compares more details of the project, compares two virtual software packages, and illustrates some applications of VEE. Finally, a description is presented regarding how to use the material for different groups of learners.

The Need

As mentioned, data acquisition and analysis problems are prevalent in manufacturing and product development areas. As such, many of the applications require trained professionals with a practical background who can work through a problem on-site. Our philosophy and the developing instructional package addresses this issue and the intended audiences.

Many of the training materials available are directed to a more highly trained professional with considerable programming background and experience. We address the technical student, and the experienced technician and designer. We successfully trained nearly 80 professionals with diverse educational and experience backgrounds. The need for step-by-step laboratory training and material was identified so it could be easily referenced when working on technical problems. Further, the programming specialist can use the material as a reference to write complex application packages and improve productivity for existing applications.

VEE is the software package that was selected since, from our perspective, it has several significant advantages. First, it is structured for relative ease of use. It is logical for the writing of unique programs. Other packages are more graphical which tend to work on complicated processes such as a chemical operation. The access to help within the program is extensive. This together with Appendix B in our book makes the debugging less painful! Second, MATLAB script and signal processing have been embedded into the latest version, known as VEE Pro, for enhanced graphics and data analysis. Finally, this version contains the ability to use any PC card or instrument from any vendor.

Overview of LabVIEW and Visual Basic

A qualitative summary of two other major software packages used for Test and Measurement applications is included below. They are LabVIEW and Visual Basic. The following summary is taken from the respective company's literature on the Internet. Refer to their web sites for more detail.

LabVIEW (National Instruments); Web Address: www.ni.com

Product Highlights and Overview

Since 1986, engineers, scientists, technicians, and researchers have successfully developed millions of LabVIEW applications. In particular, they classify the benefits of using LabVIEW into the following categories:

- productivity
- computer and instrumentation hardware
- unique solution development without complicated development tools
- powerful add-on tools for tasks such as data analysis and visualization, report generation, corporate database connectivity
- huge user network

LabVIEW is a high productivity graphical programming language for building instrumentation systems. With LabVIEW, you can quickly create front panel user interfaces, giving interactive control of your software system. To specify the functionality, you intuitively assemble block diagrams, a natural design notation for engineers and scientists. LabVIEW has all of the same development tools and language capabilities of a standard language such as C- looping and case structures, configuration management tools, and compiled performance.

Features:

- Has graphical programming, TestStand Test executive software, Measurement Studio for Visual C++, Visual Basic, and C (LabWindows/CVI)
- Has turnkey and configuration tools (Measure) and ready-to-run applications (Virtual Bench and Bio Bench)
- Gets job done quickly with integrated solutions, interactive tools, and wizards/code generation

Visual Basic Web Address; www.microsoft.com/vbasic

Product Highlights and Overview

The Visual Basic development system is the most productive tool for creating fast business solutions for Windows and the Web.

Features: Tap into productivity and performance by

- programming with the world's most popular programming language to quickly create high-performance applications and components in a RAD environment
- building fast, native code applications and components that use the technology in Visual C++® development system
- creating a broad range of reusable COM-based components including ActiveX® controls as well as middle-tier and server-side components

Description and Examples of the Instructional Package (Book)

Just-in-Time learning requires that objectives be developed for the material to be taught. The objectives are active verbs with a noun and a companion descriptive statement. These verb-noun statements are linked into a network with the objective in the node; connecting lines constrain the nodes. The final result is that educational information, devised via objectives, is delivered and learned by the student as needed and required. Therefore, it is retained longer by the learner[9] and allows the learner to acquire information more rapidly.

The concept and theory were used to develop and deliver three offerings of VEE. Labs were developed from the existing VEE documentation. This included a textbook by Robert Helsel[4] and the companion VEE User Manuals[5, 6, 7].

The referenced textbook and manuals assume that the user has extensive programming background and is educated as either a scientist or engineer. However, most users are practical-oriented, working as either technicians, technologists, or designers.

The package of labs, exercises, and projects were revised and expanded from the above references to fit into courses for technical-design personnel. The final and major feature of this package is that all material was developed and presented to the client's students who had little or no prior programming course or experience.

When Hewlett-Packard (HP) split into two corporations – HP and Agilent Technologies – the VEE software was assigned to Agilent. A new release was announced: VEE Pro, version 6[1, 2, 3]. Our latest package, which will be much more extensive than the original manual, is being developed and tested. This new package is also capable of being used with the previous release (VEE5.01).

The material is presented step-by-step so the learner has full comprehension before proceeding to the next lab. An example from the book is given below. It is based upon a virtual "vehicle radiator" whose temperature and pressure is to be monitored. The recorded data points are then processed statistically. Finally, an operator is allowed to initiate and terminate the program but cannot modify the program. (The virtual-radiator concept is initiated in Lesson 1 and is gradually developed throughout the entire book.)

Lab 8.4 – Building a Vehicle Radiator Operator Interface

This lab will show you how to compute, display, and store statistics for a variety of vehicle radiator test runs, and how to control the process and displays via an Operator Interface.

Open your VEE program and

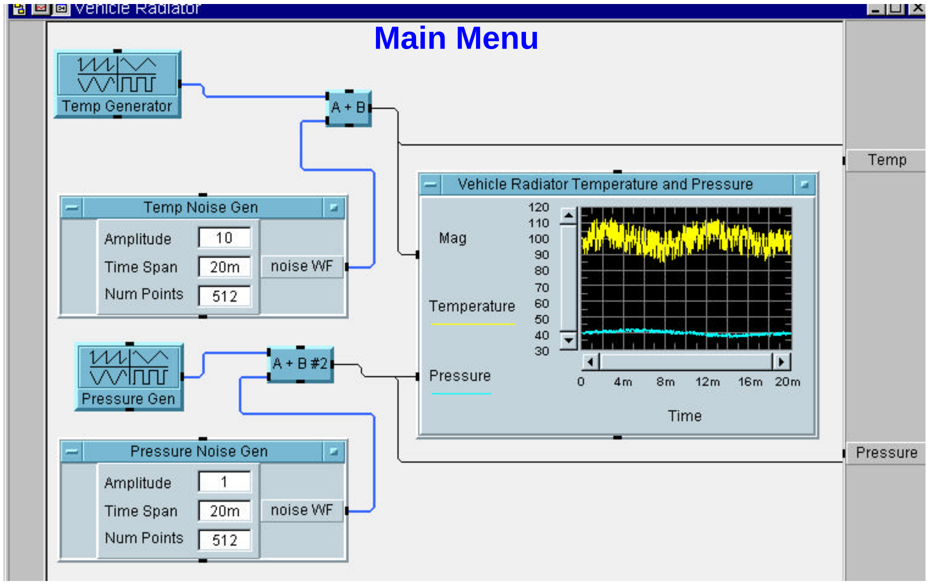
1. Clear your Work Area, remove the Program Explorer, and maximize your Work Area.

Building a vehicle radiator test database

2. Select File => LAB7-5.
3. Save As… LAB8-4 immediately.
4. Select Menu Bar =>Display => Note Pad; place it in the lower-right corner of your workspace.
5. Type in its descriptive (white) area:

   Control Vehicle Radiator
   temperature and pressure
   via an Operator Interface.

6. Size your Note Pad to be as small as possible.
7. Copy the waveform display to your clipboard.
8. Open the Vehicle Radiator UserObject; expand the UserObject to receive the display.
9. Paste the Vehicle Radiator display and center it between the Temp and Pressure output terminals as shown in Figure 8.6.



Temp Noise Gen: Amplitude 10, Time Span 20m, Num Points 512

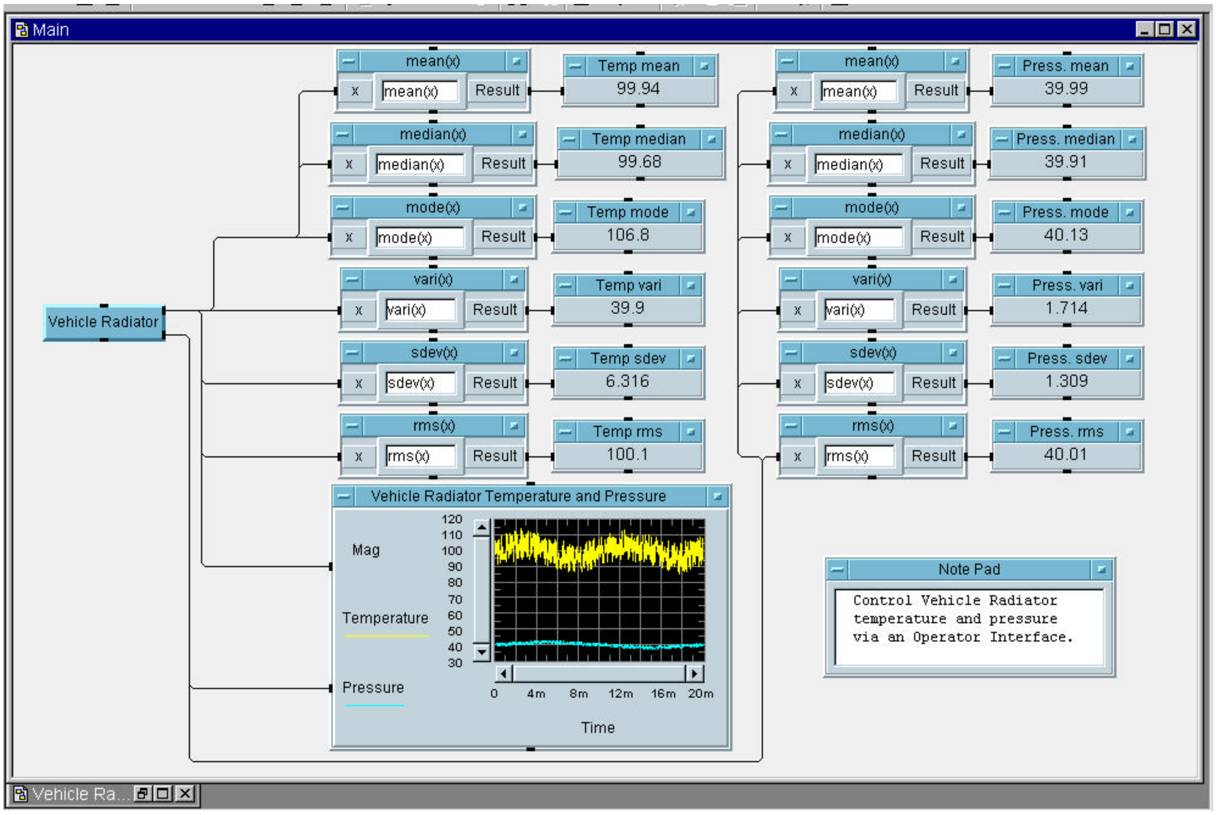
Pressure Noise Gen: Amplitude 1, Time Span 20m, Num Points 512

Figure 8.6 – The Modified Vehicle Radiator UserObject

10. Connect the Vehicle Radiator display input terminals as shown in Figure 8.6 above.
11. Save this program as LAB8-4 again.

? 12. Run this program several times with different amplitude values to be certain all portions are displaying properly.

13. Close the Vehicle Radiator UserObject via the upper-right "dash"; run this program. See Figure 8.7.



Temp mean	99.94
Temp median	99.68
Temp mode	106.8
Temp vari	39.9
Temp sdev	6.316
Temp rms	100.1
Press. mean	39.99
Press. median	39.91
Press. mode	40.13
Press. vari	1.714
Press. sdev	1.309
Press. rms	40.01

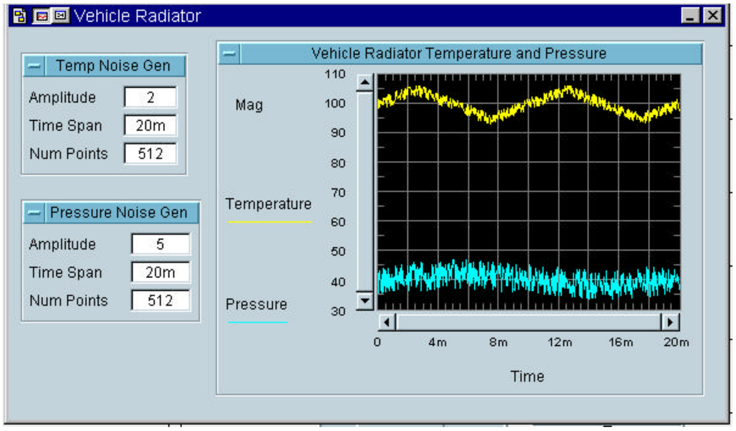
```
Control Vehicle Radiator
temperature and pressure
via an Operator Interface.
```

Figure 8.7 – Vehicle Radiator Temperature and Pressure Parameters Displayed

Creating a Vehicle Radiator Operator Interface

14. Open the Vehicle Radiator UserObject; depress and hold down Ctrl; click on
    Temp Noise Generator
    Pressure Noise Generator
    Vehicle Radiator Temperature and Pressure (the display)

15. Select Menu Bar => Edit => Add to Panel; the selected objects will appear in a Panel View; move these icons and size the interface box to appear as shown in Figure 8.8.



Vehicle Radiator

Temp Noise Gen

Amplitude 2

Time Span 20m

Num Points 512

Pressure Noise Gen

Amplitude 5

Time Span 20m

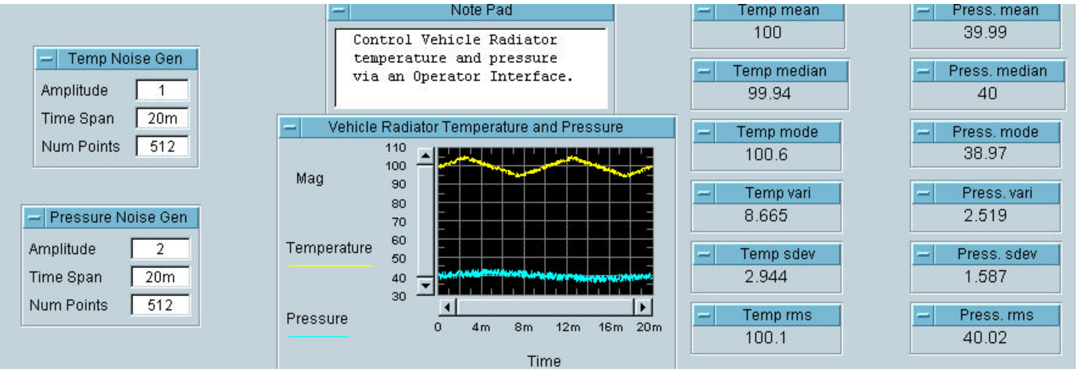
Num Points 512

Figure 8.8 – Vehicle Radiator Operator Interface

16. Save Lab 8-4 again; run this program to verify the displays in Figure 8.8.
17. Move the Main work area so there is room to its left; open the Vehicle Radiator object.
18. Copy, one at a time, the two noise generators and paste them in the space provided on the left portion of the Main work area.
19. Close the Vehicle Radiator UserObject via the upper-right "dash".
20. Hold down the Ctrl key; highlight the following icons:

Temp mean	Press. mean
Temp median	Press. median
Temp mode	Press. mode
Temp vari	Press. vari
Temp sdev	Press. sdev
Temp rms	Press. rms
Temp Noise Gen	Pressure Noise Gen
Note Pad	Vehicle Radiator Temperature and Pressure

21. Select Menu Bar => Edit => Add to Panel; arrange the icons as shown in Figure 8.9.



Temp Noise Gen

Amplitude 1

Time Span 20m

Num Points 512

Pressure Noise Gen

Amplitude 2

Time Span 20m

Num Points 512

Note Pad

Control Vehicle Radiator
temperature and pressure
via an Operator Interface.

Temp mean 100

Temp median 99.94

Temp mode 100.6

Temp vari 8.665

Temp sdev 2.944

Temp rms 100.1

Press. mean 39.99

Press. median 40

Press. mode 38.97

Press. vari 2.519

Press. sdev 1.587

Press. rms 40.02

Figure 8.9 – The Vehicle Radiator Operator Interface Displaying Statistics

22. Save this program again as Lab 8-4.

Securing an operator interface

23. Select Menu Bar => File => Save Secured Run Time Version …
24. Select Menu Bar => File => Open; return to the Secured Run Time version by first selecting LAB8-4.vee and changing the .vee extension to .vxe.

    Note: The Main title bar now will read: "Main (Secured).

? 25. Change the noise amplitudes by factors of 5, 10, or more; run the program several more times. What changes in the statistics have you observed? How do these changes compare with the waveform noise distortions?

Your notes:

____

____

? 26. For a given set of noise amplitudes, change the Num Points by factors of two, starting with values of 8 for both noise generators.

? 27. Increase the Num Point values by factors of two; observe the effect on the statistical values.

Your notes:

____

____

28. Exit LAB8-4.vxe.

The above example illustrates how students progress, step-by-step, so they will understand the capabilities and limitations of VEE Pro. (Step 2 of the above lab portion copies a previous laboratory so students can more efficiently incorporate the new material into the next material to be learned). The diamond (**?**) symbol is used to indicate in-depth exercises to be accomplished before proceeding to the next steps. The addition of the diamond steps improves the student's learning of VEE Pro and its flexibility, versatility, and power.

Starting with step 14, this lab demonstrates how the students explore the Operator Interface. An Operator Interface allows the devised material to be summarized for use by an individual (operator) who does not need to know the details of the design. However, that individual needs to be able to access and observe the results of data acquisition. (Only the program designer can modify the actual program.)

Instruction in VEE Pro can be accomplished in at least three different approaches. The first format presents and discusses the pre-lab in fifteen to thirty minutes, depending upon the depth and number of new concepts introduced. Students then proceed through each lab in the lesson, working individually at a station. Each lesson is designed to take the average student about two hours and typically consists of four to five related labs.

The second format is to use our package in a self-paced study mode. Students will study the pre-lab material and, on their own, schedule time on the provided computer network to complete the labs.

The last format combines the two formats above: The pre-lab is presented to the entire class; the labs are completed individually by each student. The only constraint on the student is that each previous lab must be completed before the next pre-lab can be presented.

The authors believe that the most powerful feature of this package will be its use as a reference. During the designing and running of real manufacturing processes, and designing-and-analyzing experiments, the six appendices become extremely valuable to the user.

The following items describe each appendix, its contents, and how it is used effectively.

- Appendix A – the VEE program, when opened, consists of a menu bar and a tool bar (buttons). This appendix is a list of all pull-down menus and buttons for the entire VEE Pro, with shortcut-keys (i.e. Ctrl + …) shown in bold and underlined. The names of the icons on the Tool Bar are also given.
- Appendix B is a list of the partial programming sequences developed in the lessons. A cross-index of these sequences is provided, arranged alphabetically to allow rapid reference by the program developer.
- Appendix C is a list of the virtual devices and instruments available within VEE. For each of these items, a summary of each applicable Help menu is provided.
- Appendix D contains the definition of the specialized technical terms used in the manual.
- Appendix E contains the twenty pre-labs for each of the lessons.
- Appendix F is an introductory 12-hour seminar for supervisors and engineering managers. It is designed for those who merely want to become familiar with the capabilities of VEE so they can better guide and evaluate their employees who are using this instructional material and the VEE programs[8].

Benefits to the Technical Community

Software such as the VEE family of programs dramatically reduces monitoring-and-control software development time. The desired software programs are developed off-line by trained technical personnel using virtual equipment and devices. Then the virtual devices are replaced by real equipment, each of which has a companion interface card. Thus, program development does not tie up laboratory, manufacturing, and process-control equipment.

This package provides the applicable tools for learning VEE so programs can be written for

- data acquisition,
- data processing, and
- process control.

The necessary detailed information is provided within the lessons.

Summary

The total instructional package described in this paper has been tested in draft form three times and has been updated based upon user input. The new VEE Pro release has been applied, which contains significantly more powerful refinements such as the addition of MATLAB® embedded script and graphics. This improvement enables one to model or simulate advanced mathematical functions or devices and display their output results. In fact, any process, device, or experiment can be designed, tested, and operated in either a real or virtual environment. Colleges, universities, and technical schools may prefer to offer the labs and training in a virtual environment[10]. Companies and associated laboratories may want to combine experiments and processes with real and simulated application packages.

This VEE Pro learning approach will be beneficial to educators and company trainers where preparation for later application to data acquisition problems is planned. This need is becoming more apparent as worldwide manufacturing and experimentation becomes more automated.

References

1. Agilent Technologies staff, Advanced Programming Techniques, March 2000.
2. Agilent Technologies staff, Agilent IO Libraries Installation and Configuration Guide for Windows, 4th edition, May 2000.
3. Agilent Technologies staff, User's Guide, March 2000.
4. Helsel, R., Visual Programming with HP VEE, third (1998) edition, Prentice Hall, Inc.
5. Hewlett Packard staff, Controlling Instruments with HP VEE, 1998 edition.
6. Hewlett Packard staff, Getting Started with HP VEE, 1998 edition.
7. Hewlett Packard staff, HP VEE Advanced Programming Techniques, 1998 edition.
8. Hulbert, T. E., and Angus, R. B., "A Survey Course on VEE Pro Software for Engineers and Technologists" ASEE Annual Conference Proceedings, June 2001.
9. Hulbert, T. E., Hansberry, E. W., and Angus, R. B. "Just-In-Time Education™: An Idea Whose Time is Overdue" ASEE Annual Conference Proceedings, June 1996.
10. Mannix, Margaret "The Virtues of Virtual Labs", ASEE Prism, September, 2000.

Biographies

THOMAS E. HULBERT is a Professor Emeritus at Northeastern University. For 35 years, he served as a faculty member and administrator in their College of Engineering. For the last 13 years, he was Director and Associate Dean of the School of Engineering Technology. Prior to joining Northeastern, he worked as a Senior Industrial Engineer for eight years. He has written numerous papers on educational innovation and has contributed to several textbooks. He is the Coordinator for the Professional Engineering Review courses at Northeastern.

ROBERT B. ANGUS is a Senior Lecturer at Northeastern University with 55 years of part-time and full-time teaching experience that includes mathematics, physics, and electrical-engineering courses. He has also been a design engineer, engineering manager, and senior engineering specialist for more than 20 years. For the past 20 years, he has been an engineering consultant specializing in circuit and system design, curriculum development, and technical-manual writing.