AC 2011-1234: INCORPORATING LABVIEW(R) IN JUNIOR ELECTRICAL ENGINEERING LABS

Bill Yang, Western Carolina University

Dr. Yang is currently assistant professor at Western Carolina University. He holds Ph.D. degree in Electrical Engineering from Princeton University. Prior joining WCU he has worked more than 7 years at Bell Laboratories, Lucent Technology, Inc. as Member of Technical Staff and Ciena Corp. as Principal Engineer, doing research in photonic networks and optoelectronics. His teaching interest focuses on the project-based learning (PBL) model of engineering education with self-directed learner as enhanced educational outcome. His research area focuses on optoelectronics, semiconductor lasers and metamaterials.

Robert D. Adams, Western Carolina University

ROBERT ADAMS is an Associate Professor and Program Director of Electrical Engineering at Western Carolina University. His research interests include digital signal processing, bioelectromagnetics, and engineering education. Dr. Adams is a senior member of IEEE and a member of ASEE.

Aaron K. Ball, Western Carolina University

Dr. Aaron K. Ball is a full professor in the Engineering and Technology Department at Western Carolina University in Cullowhee, North Carolina. He holds a B.S. and an M.S. from Appalachian State University, and earned his doctorate from Virginia Polytechnic Institute and State University. His areas of interest include automation, fluid power, advanced machining, prototyping systems, and applied research.

Incorporating LabVIEW® in Junior Electrical Engineering Labs

Abstract

LabVIEW® is a popular instrument control and automation software tool that is widely used in industry. Sufficient LabVIEW programming skills greatly improve engineers' productivity, and these skills are in high demand. On the other hand, it is usually a common weakness in Electrical Engineering (EE) as well as in general engineering programs that the students are lacking in the application of appropriate software to solve engineering problems. As a part of the continuous curriculum improvement strategy, we are implementing a LabVIEW programming thread in the EE curriculum to enhance the students' real-world ready and workplace applicable skills. Built upon the successful incorporation of LabVIEW programming into engineering technology programs within the same department, this paper will discuss the general plan of the implementation of the LabVIEW programming thread in the EE program and outline in detail the designed LabVIEW programming activities in junior EE Labs. The LabVIEW series sessions in junior EE Labs starts with simple activities of instrument communications and file manipulation for data collection, progresses into more systematical activities of instrument control and data trace collection for specific instruments (such as a digital oscilloscope), and gradually integrates into other Lab sessions. Student feedback in the form of standard Student Assessment of Instruction (SAI) as well as Small Group Analysis (SGA) focused on the learning experience enhancement will be used as the formal assessment of the incorporation of LabVIEW programming activities within the EE curriculum. This paper will emphasize the curriculum integration strategies, educational methodology and merit of this approach.

Introduction

Laboratory Virtual Instrumentation Engineering Workbench (LabVIEW®) is a popular software platform of graphic programming environment developed by the National Instruments and specifically targeted for automation of processing, test, and measurement equipment. Since its modest start in late 1980s, LabVIEW has become a widely accepted engineering software tool for industrial automation, scientific data acquisition, and instrument control in almost every engineering related industry and research labs. Consequently, LabVIEW programming skills greatly improve engineers' productivity and these skills are in high demand. Early exposure to LabVIEW programming and its applications are therefore very much beneficial and desirable to undergraduate engineering students[1-8]. On the other hand, as both our employer and alumni surveys indicated, our graduates are weak in the application of appropriate software to solve engineering problems. This is indeed usually a common weakness for Electrical Engineering (EE) programs[9-12]. In general, a fine line of balance has to be maintained between theory and practice in all engineering programs and the issue of the application of appropriate engineering software is only a part of the bigger picture of the well-rounded engineering education[11,13,14]. Institutionally, our engineering programs, including Engineering Technology (ET), Electrical and Computer Engineering Technology (ECET), and EE, are implementing a cross-program and inter-disciplinary approach based on the concept of project-based learning (PBL) to enhance the student outcomes. The central idea of PBL enhanced curricula is to use hands-on problems to drive students to internalize the principles of what they have learned and in doing so to bring about a deeper understanding of the content, a respect for lifelong learning and enhanced problem-solving skills. As a part of this continuous improvement effort, the EE program is implementing both MATLAB® and LabVIEW® threads in the curriculum. Both threads involve a couple of appropriate courses with course activities focused on the popular engineering software. Incorporation of LabVIEW activities in EE 311 Systems and Electronics Lab reported here is a pilot of this greater and more integrated effort.

LabVIEW programming and its applications have been incorporated into the other two engineering programs (ET and ECET) within the Engineering and Technology department previously[8]. There, the issue was focused on addressing the demand of rapid time-to-market design, prototyping, testing and production. LabVIEW by the National Instruments has been identified as the engineering software package that provides the ability to merge the virtual and real worlds, which is ideal for the programs' focus on the applied scientific knowledge and engineering principles. LabVIEW is also being used in the Master of Science in Technology (MST) graduate program at Western Carolina University to support graduate courses in automation systems and directed projects. The integration of LabVIEW activities into the EE curriculum therefore fulfills the requirement of the cross-program and inter-disciplinary integration of the PBL enhanced engineering curricula. It is also believed to be a more effective instructional approach when compared to the conventional approaches with little use of any engineering software. The integration of theory, hands-on learning experience, and the application of the theory and skills is also in harmony with the core concept of the applied approach of teaching and learning promoted at Western Carolina University. Our objective here is to enhance the student's theoretical understanding through the using of the common engineering software packages, and, on the same token, to improve the student's ability of application through applied projects in common industrial systems and research laboratory environment.

Built upon the successful incorporation of the LabVIEW programming into the engineering technology programs within the same department, this paper will discuss the general plan of the implementation of the LabVIEW programming thread in the EE program and outline in detail the designed LabVIEW programming activities in a junior EE Lab.

LabVIEW Programming Thread in the EE program

As outlined in the previous section, LabVIEW has several primary strengths which make it attractive and suitable for integration into the PBL enhanced EE program. First, LabVIEW is widely used in industry and academia for automation, data acquisition, and instrument control. These contemporary real-world working skills are important for EE students and are also among the top priorities of student outcomes in our engineering programs. Second, LabVIEW provides the capability to merge the virtual and real worlds, which is ideal for our program's focus on applied scientific knowledge and engineering principles. Last but not least, the graphical programming environment lends itself very well to the intuitive and interactive learning, which allows students to engage the tool rapidly in their design process and experimentation on computer based automation and data acquisition systems. Table 1 shows our eight-semester suggested EE curriculum with the potential PBL core and LabVIEW thread courses identified. The eight-semester curriculum looks very typical as many other EE programs and still has an emphasis on mathematics and science, as well as other traditional EE courses. However, it indeed has a heavy hands-on learning component comprising the core PBL courses.

Table 1: Eight-semester course plan for B.S.E.E degree

Electrical Engineering - B. S. Degree – Western Carolina University					
Suggested Course Plan					
Fall Semester	**Hours**			**Spring Semester**	**Hours**
ENGL 101 Composition I	3			ENGL 102 Composition II	3
MATH 153 Calculus I	4			MATH 255 Calculus II	4
ENGR 199 Eng/Proc/Prin. I (**PBL**)	3			ENGR 200 Eng/Prac/Prin. II (**PBL**)	2
CHEM 139 General Chemistry	4			PHYS 230 Calculus Based Physics I	4
Liberal Study Perspective	3			Wellness	3
	17				**16**
Fall Semester				**Spring Semester**	
EE 201 network Theory I	3			EE 202 Network Theory II	3
EE 200 Computer Utilization in C++	3			EE 222 Electrical Engineering Design I (**PBL**)	2
EE 221 Logic Systems Design I	3			EE 212 Instrumentation & Networks Lab (**LV**)	1
EE211 Logic and Networks Lab (**LV**)	1			CMHC 201 Communications	3
MATH 320 Ord. Diff. Equations	3			MATH 253 Engineering Calculus III	3
PHYS 231 Calculus Based Physics II	3			PHYS 310 Modern Physics	3
	16				**15**
Fall Semester				**Spring Semester**	
EE 351 System Analysis I	3			EE 322 Electromagnetic Waves	3
EE 341 Electrical Engineering Design II (**PBL**)	2			EE 332 Electronics	3
EE 331 Fund. Electronics & Semiconductors	3			EE 342 Solid State Electronic Devices	3
EE321 Electromagnetic Fields	3			EE312 E-M and Electronics Lab	1
MATH 370 Probability and Statistics I	3			ENGR 300 Prof. Development (**PBL**)	1
EE 311 Systems and Electronics Lab (**LV**)	1			Liberal Study Perspective	3
				PHYS 322 Optics and Materials	3
	15				**17**
Fall Semester				**Spring Semester**	
EE 401 Senior Design I (**PBL, LV**)	2			EE 402 Senior Design II (**PBL, LV**)	2
EE 411 Analog and Digital Communications	3			Liberal Study Perspective	3
Technical Elective	3			Liberal Study Perspective	3
Technical Elective	3			Liberal Study Perspective	3
EE 412 EE Professional Practice (**PBL**)	2			Technical Elective	3
Liberal Study Perspective	3			Technical Elective	3
	16				**17**
Total program requires 127 hours. Liberal Studies 42 hours. Major Requirements 85 hours					
PBL: Project-Based Learning Core; **LV:** LabVIEW thread.					

The selection of PBL core courses is a year-long rigorous process across the programs including the departmental special task force on PBL and many hours of faculty discussions. First, the desired PBL outcomes are articulated by the PBL task force and vetted by the faculty. Then the needed credit hours are distributed through the curriculum. And finally the individual courses are identified to be the PBL core courses. The PBL enhancement, however, will be implemented within the same amount of credit hours and will greatly enhance the experiential learning and benefit from the hands-on features of the programs.

The selection of LabVIEW thread courses in EE program, however, is driven by both the PBL objectives as well as the ABET continuous improvement requirements. Rather than offering a stand-alone course on LabVIEW, we have chosen to integrate the LabVIEW content into the EE curriculum. This practice is encouraged by the favorable experience and student outcomes obtained in ET and ECET programs[8], where the LabVIEW content is introduced in various courses and reinforced there by hands-on learning in the disciplinary context. Accordingly, the engineering lab courses and the senior design courses are naturally chosen as the targeted courses to integrate the LabVIEW thread. It is in the curricular design that the virtual instruments (VIs) built in previous semesters will be used and/or leveraged in the courses later on, including the senior design projects. As shown in Table 1, the LabVIEW thread, accordingly, will be integrated in the curriculum at a couple of places including: EE211 Logic and Networks Lab, EE 212 Instrumentation & Networks Lab, and EE 311 Systems and Electronics Lab. The senior capstone project, which consists of the two-semester sequence EE 401 Senior Design I and EE 402 Senior Design II, will most likely, depending on the projects, provide opportunities for the students to apply the LabVIEW programming skills they acquired earlier in the curriculum. As can be clearly seen from this suggested course plan and the plan for the LabVIEW thread integration, a complete cycle of introduction, cultivation, and application is implemented in this PBL enhanced EE curriculum for the LabVIEW thread. Upon completion of the program, therefore, the students should be adequately familiar with the LabVIEW programming and equipped with basics of this important working skill.

Pilot LabVIEW activities in EE junior Lab

The first course that has piloted the integration of LabVIEW programming thread is a junior lab, EE 311 Systems and Electronics Lab. The junior lab is focused on the electronic circuits utilizing semiconductor devices such as diodes, MOSFETs, BJTs, as well as the op-amps. Several lab sessions involve measurements of frequency response curves, which require students to set up the function generator to a set of fixed frequency points and measure the circuit responses such as voltage gain and input impedance. Most of these measurements are carried out by comparing voltage responses at two different places in a circuit (e.g., input and output nodes). Sometimes the data traces also need to be recorded and reported. While mastering the basic measurement concept and techniques is one of the major goals for the lab, implementing at a few properly chosen points in the course the instrument automation and data acquisition is a nice and natural add-on and hopefully can enhance the student learning outcomes in both the basic skills and the LabVIEW instrument control, data acquisition, and automation. Table 2 shows a typical course schedule for the EE 311 System and Electronics Lab with LabVIEW activities incorporated.

The designed LabVIEW activities are arranged purposefully throughout the course of the whole semester with experimentation and instrument control and data acquisition developing in depth gradually. LabVIEW activities from simple examples of single function block file I/O to a moderately structured automation process measuring the frequency dependence of voltage gain or input and output impedance. The LabVIEW activities integrated here are more focused on the instrument control and data acquisition rather than utilizing the built-in data processing capability of current LabVIEW. The data processing and analysis are carried out after the lab sessions using other spreadsheet applications such as Microsoft Excel® and MATLAB®, with which students are generating figures and plots for their lab reports.

Table 2: Course schedule for EE 311 System and Electronics Lab incorporating LabVIEW programming activities

Week	Dates	Laboratory
1	8/23	Course matters, general electronic lab issues
2	8/30	Lab 1 Op amps
3	9/6	LabVIEW(R) session 1: Introduction, File I/O, and Digital Multi-meter
4	9/13	PSpice introduction
5	9/20	PSpice introduction to op amps
6	9/27	Lab 2 Diodes
7	10/4	Lab 3 Diode Rectifiers
8	10/11	Lab 4 Zener Diode Regulators
9	10/18	LabVIEW(R) session 2: Instrument Control (Digital Oscilloscope)
10	10/25	Lab 5 MOSFET Transistor Basics
11	11/1	Lab 6 MOSFET Transistor: Amplifier Configurations
12	11/8	Lab 7 MOSFET Amplifier: Input/Output Impedances LabVIEW(R) session 3: Automation (Frequency Dependence)
13	11/15	Lab 8 BJT Transistor Basics
14	11/22	Lab 9 BJT Transistor: Amplifier Configurations LabVIEW(R) Automation: Frequency Dependence
15	11/29	Lab 10 BJT Amplifier: Input/Output Impedances LabVIEW(R) Automation: Frequency Dependence
16	12/6	Last Lab report due

Introductory LabVIEW session:

In the introductory session the students engage in LabVIEW exercises involving different data formats, 1D and 2D arrays, and saving data in spreadsheets. The purpose of these exercises is to allow the students to gain confidence in designing and implementing their own user interfaces.

The first LabVIEW session is at the week 3 early in the semester. This session is one of two dedicated LabVIEW sessions in the whole semester. Other LabVIEW activities are integrated with other main experiment lab sessions such as recording data sets for a frequency response analysis of BJT amplifier voltage gain. The main purpose of this first introductory session was to help the students to become familiar with the user interface of the LabVIEW programming as well as the concept of graphical programming. Different data types represented by different color lines are illustrated and a simple program of saving an input text into a file is built step by step together with the students. This simple program is shown in Figure 1. The program takes an

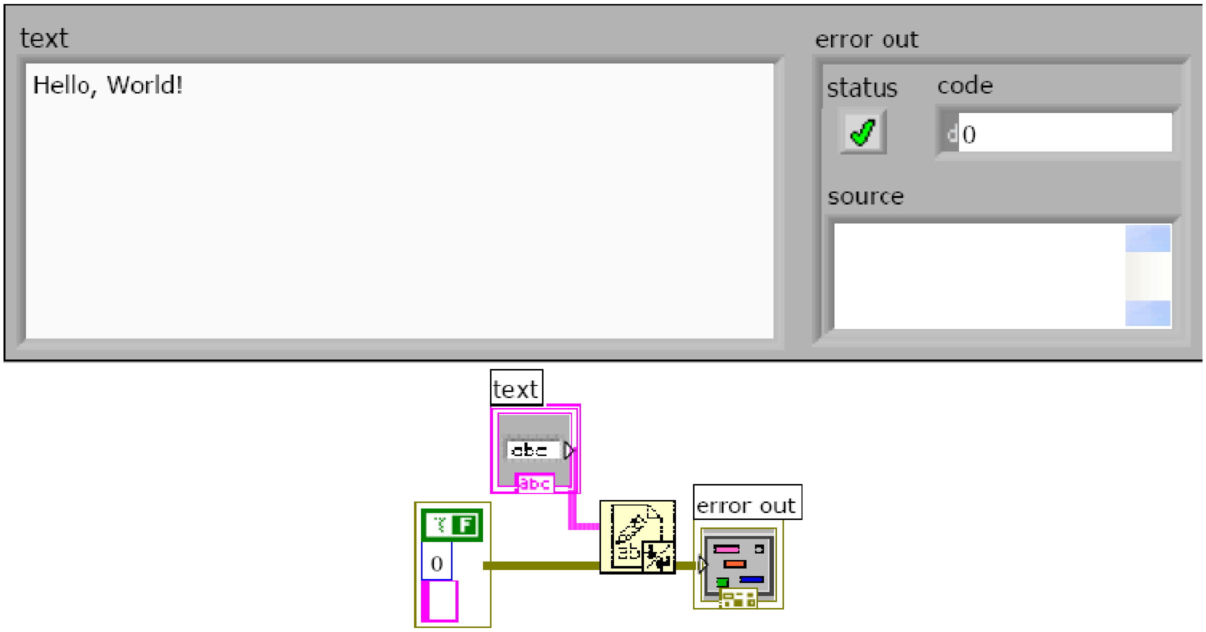

Figure 1: “Hello, World” LabVIEW® program that saves a text field into a file. Once running, the program opens the native Windows® file selection dialogue box and saves the input text to the specified file. Top: The front panel. Bottom: The block diagram.

input text from the input text field and saves it to a text file, which is specified by opening the native Microsoft Windows® file selection dialogue box. This first “Hello, World” program is an eye-opening experience for many students who do not have extensive programming experiences. The built-in file dialogue box and the simple graphical programming interface help to build considerable confidence for the students, and the tool is well accepted by the students instantaneously.

(a) (b)

Figure 2: LabVIEW® program that saves a 2D numerical array into a spreadsheet file. Both array data types and format strings are introduced. (a) The front panel. (b) The block diagram.

The second example of the introductory session involves saving a 2D numerical array into a spreadsheet file. This again can be simply implemented by one function block as shown in Figure 2. However, this second example introduces a critical data type, 2D numerical array, and a few array operations. Format strings for numerical data are also covered. The saved spreadsheet file can then be readily read by any spreadsheet applications such as Microsoft Excel or MATLAB. Further data processing can be performed in these other applications. The data process capability of the LabVIEW is intentionally avoided here in the designed LabVIEW activities to keep the focus on the instrument control and data acquisition. Again, the program is built step by step together with the students and this exercise keeps building the confidence for the students and raises their comfortable level with different data types especially the array operations.

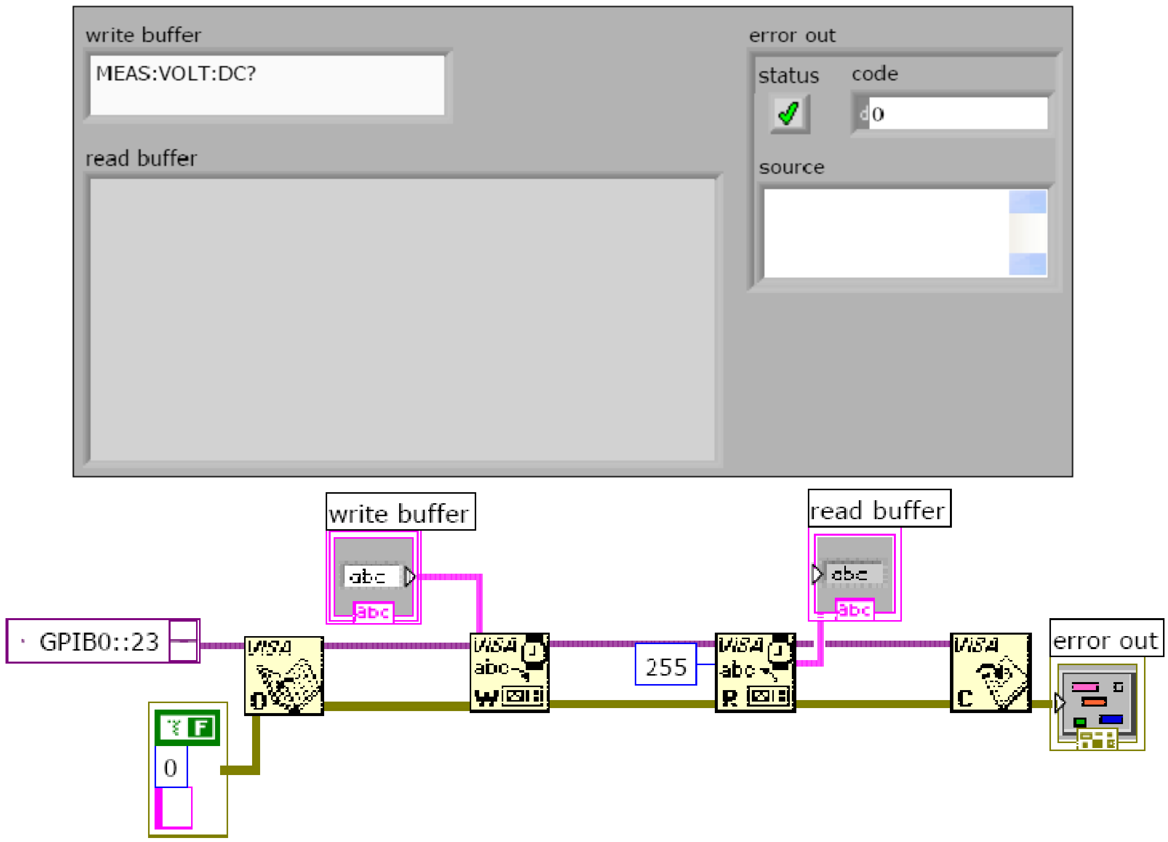

Figure 3: Interactive control and communication with Agilent 34401A Digital Multimeter via GPIB. Top: The front panel. Bottom: The block diagram.

After these introductory and confidence building exercises, we dive into the instrument control by first introducing the GPIB interface and then interactively communicating with a digital multimeter (Agilent 34401A). This can be implemented by a minimum of four function blocks involving VISA instrumentation interface. The program is shown in Figure 3. Although this is still a fairly simple example, it nonetheless emphasis on two critical graphical programming concepts: data flow and instrumentation interface. The Students were given plenty of opportunities and experiment until the satisfactory results are obtained. While not thoroughly covered, the exercise builds the students' familiarity with the important PC instrument interface GPIB. After the success of this first instrumentation example, other individual instruments with simple controls such as Agilent 33250A Function Generator are introduced. These exercises conclude the first dedicated LabVIEW session and build good confidence level for the students.

Virtual Instrumentation (VI) Example: Digital Oscilloscope

The second and also the last dedicated LabVIEW session takes place at the week 9 in the middle of the semester. This time a fairly complicated example of data acquisition using a digital oscilloscope (Agilent 54621A) is tackled. The complication of this project comes from several aspects. First, a new important instrument interface, namely RS-232 serial communication is used instead of the GPIB. Second, Agilent 54621A transmits large amount of oscilloscope trace data most efficiently in a binary data format instead of the ASCII text format. This means one needs to code their own data type converter to interpret correctly each byte send over the communication link. This is one of typical core tasks of any data acquisition applications. Third, we also added the data presentation, namely the graphical representation of the collected data in the user interface to build a more real-sense virtual instrument. While the instrument control and data acquisition become significantly more sophisticated in this second LabVIEW session, we restrained from using any built-in programming structures such as loops or frames. This helps us to focus our effort on the instrument control and data acquisition side. Figure 4 shows the

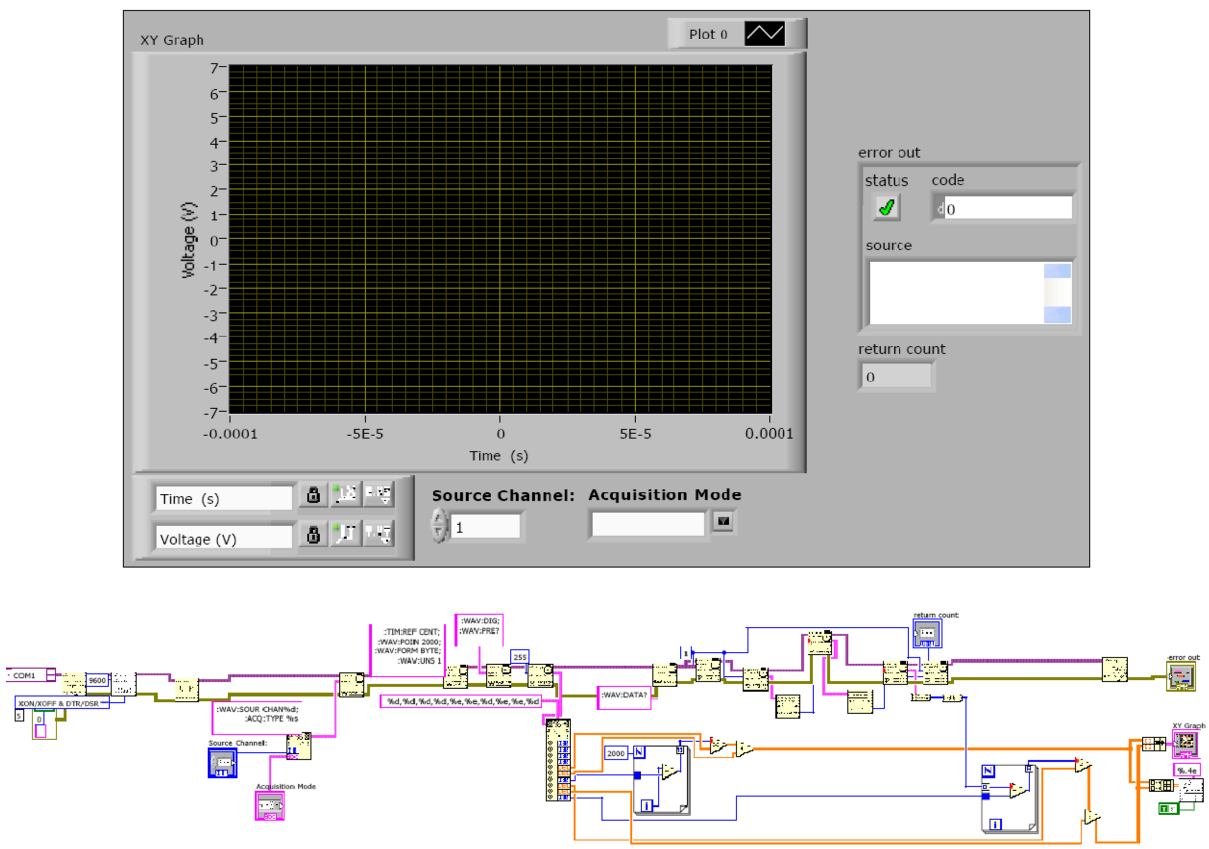

Figure 4: Digital oscilloscope virtual instrument (VI). The VI gives a graphical interface representing the collected data trace while providing the basic control of channel and acquisition mode selections. Top: The front panel. Bottom: The block diagram.

finished program. The program selects the channel and acquisition modes, records the time and voltage data, graphically represents the data on screen, and saves the data to a spreadsheet file. Several issues peculiar to the RS-232 serial communication of large amount of data were encountered and data acquisition programming remedies were devised and implemented. The student programming is helped by the instructor with a basic block diagram showing the necessary steps of a typical data acquisition application.

Integration of Vis and Automation:

The third LabVIEW session follows the second one fairly quickly in the week 12 when the main experiment of the session involves taking frequency dependence of input and output impedance of a MOSFET amplification circuit. In the previous week, the students have manually taken the frequency dependence of the voltage gain of MOSFET amplification circuits by adjusting the frequency of the function generator and recording the input and output voltages. This LabVIEW session provides an opportunity for the students to taste the integrated instrumentation control and automation. The previous example of the digital oscilloscope virtual instrument is modified to have input and output connectors which controls the channel number and acquisition mode selection on the input side and yields the trace data on the output side. In addition, error in and error out are connected to assist the operation flow. Now with the help of the programming structures such as for loops and frame sequences, an integrated process of taking frequency response or frequency dependence can be automated. The front panel of the program is shown in

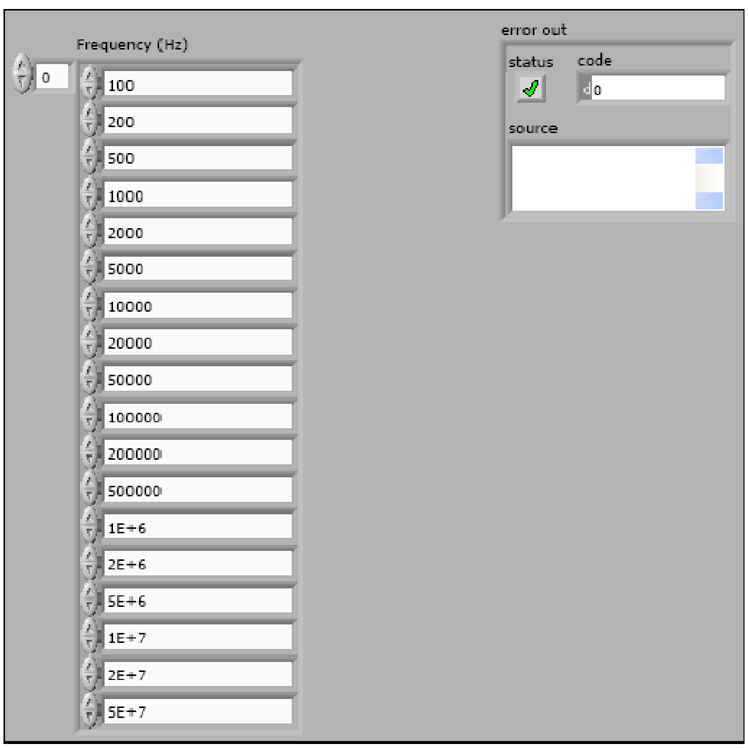

Figure 5: Front panel of frequency response measurement virtual instrumentation. Data traces are available at each digital oscilloscope VI's user interface. The program saves voltage traces at two places in a circuit to files named after the frequency settings.

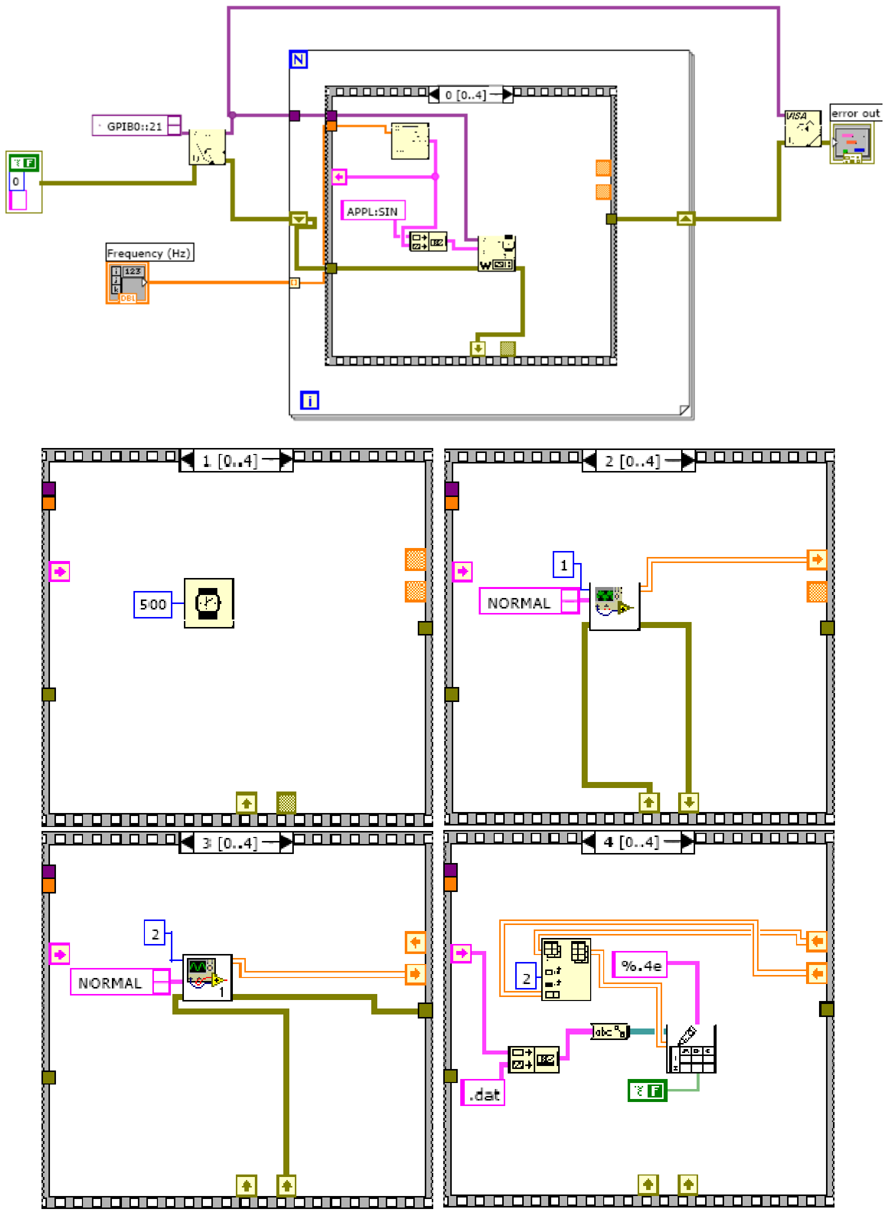

Figure 6: Block diagrams of the frequency response measurement virtual instrumentation. Both loop structure and frame sequence structures are utilized. Digital oscilloscope VI is also integrated into the process.

Figure 5. Individual data traces are available at the digital oscilloscope VI's user interface. The program saves the trace data in separate spreadsheet files according to their corresponding frequency settings for students to process. Figure 6 shows the structured block diagram of the program

The similar automation programs are then later used in the lab sessions that follow this LabVIEW integration session to collect data for frequency dependence of gain and input and output impedance of the BJT amplification circuits. These exercises enhance the application of the LabVIEW instrumentation and automation programs. Again, the built-in data processing capability of the LabVIEW programming is not explored here. The purpose here is to focus on introduction to the tool, and the instrument control and data acquisition features of LabVIEW.

Student Learning Outcomes

The pilot integration of LabVIEW in EE 311 System and Electronics Lab has a goal of gauging the students' perception of learning LabVIEW graphical programming for instrument control and data acquisition. The integrated LabVIEW activities are also carefully designed to build the students' confidence and ability to use the software. Accordingly, the students' response to the designed LabVIEW activities was enthusiastic. Through the integrated LabVIEW instrument control and data acquisition exercises, the students achieved a high level of confidence of using the software tool and were eager to try many new things with LabVIEW. The success, as also pointed out by other authors and previous studies, at least partially attributes to a more visual learning environment this generation of students has grown up with, for example, through the exposure to interactive video games. Abstract concepts such as various data types, file I/Os, instrument interfaces are graphically represented in the LabVIEW "G" language by data lines of different colors and line shapes, icons and blocks. This strongly supports the bridging between the abstract theory and the real world applications, which was clearly demonstrated in the students' response, their perception of learning, and the sense of accomplishment. Implementing the LabVIEW programming thread in the PBL enhanced EE curriculum is therefore very appropriate and promising in order to address the student outcome weakness in the application of appropriate software to solve engineering problems. In addition, the integrated LabVIEW activities can also greatly strengthen students' real-world ready and workplace applicable skills. The student perception of learning score on the Student Assessment of Instruction (SAI) is 3.4 with very positive feedbacks. This is a definitive improvement from 3.3 of a previous lab course with the same cohort of students and instructor. The positive v.s. negative comments ratio also improved from 1.17 for the previous course to 1.5. One student commented that "the best aspect of the course was the introduction to LabVIEW and the ability to run experiments entirely using the computer. Even if it is tedious at first, it seems like something that would be very useful later". Other feedbacks collected from the students are strongly in favor of the LabVIEW programming learning experience, even though the students typically spent more time in the lab as well as in a couple cases outside and beyond the lab sessions trying to debug and make the program working properly. In terms of LabVIEW programming competence at the end of the semester, the instructor evaluated that out of 12 students, 4 exceed the expectation, 4 are satisfactory, and 4 are marginal. The enthusiastic attitude toward learning LabVIEW programming, however, was across the board and all students gained a genuine sense of accomplishment using LabVIEW to control the computer and computer interfaces with the instrument. The entire pilot practice was a success.

Educational Merit

The educational merits of the integrated LabVIEW programming thread in the PBL enhanced EE curriculum are apparent and significant, even in light of the limited assessment data at hand. First of all, given the popularity of LabVIEW programming in instrument control, data acquisition, and industrial automation, the introduction of this popular engineering software to the undergraduate EE students is extremely valuable for the students. Secondly, the hands-on and interactive approach of the LabVIEW programming, for example, for instrument control, greatly enhances the students' understanding of the abstract concepts of data transfers and procedure analysis. When a program is not functioning as expected, students can visually and interactively follow the data lines and the status of each function block in the debug mode and this experience can greatly enhance their problem solving and analysis skills. Thirdly, the enthusiasm of learning transpired by the ability of the interactive LabVIEW programs can bring about the best aspects of active and intentional learning experiences for students. In the pilot effort, many students developed their competence of the software tool and started to apply it in other areas of their study and projects, achieving the integrative learning. Last but not least, the integration of LabVIEW programming in EE curriculum blends well with the existing practices in the other engineering programs within the department and this truly builds a real sense of integrated, cross-program, and inter-disciplinary engineering PBL core, especially because that the common central idea of the PBL enhanced curricula is to use hands-on problems to drive students to internalize the principles of what they have learned. Pedagogically, the integration of LabVIEW into the curriculum instead of being offered in one stand-alone course also provided a conducive learning environment for the students to acquire this important content in a highly effective fashion of learning by doing in a context.

Educational merit also lies in the novel approach to the well-rounded engineering education centralized on the concepts of the PBL enhanced engineering curricula. The fine balance between the engineering theory and engineering practice skills is at the center of many engineering education forums. Real world applicable and inter-disciplinary skills such as LabVIEW programming are not always an emphasis nor explicitly taught in engineering curricula where the focus has been on theoretical content and analytical skills of the specific engineering disciplines. While demonstrating the ability of application of appropriate software to solve engineering problems is an explicit requirement from both industry and ABET, how to seamlessly integrate such activities with the student outcomes of competence within the already crowed four-year engineering curricula is an open question. Our pilot program carefully designed such activities and achieved the goal of promoting the confidence and competence of the students in respect to the application of appropriate software to solve engineering problems. The approach implemented across the engineering disciplines in all three engineering programs here at Western Carolina University provides a logical and systematic method for building on theory and developing essential real world software application skills.

Conclusions and Future Work

In conclusion, the pilot program integrating LabVIEW in the junior EE electronics lab is largely a success. It strongly demonstrated the students' enthusiasm to learn in such interactive and hands-on environment as well as provided many valuable lessons on the instructional approach. Positive results of the applied approach have been confirmed by student activities in the lab and through the student feedback and performance. It is also found out that the dedicated LabVIEW sessions had provided a much more focused teaching and learning stage than the integrated sessions where the students had to deal with obstacles from hardware, software, as well as the interface realms. Although it was a valuable learning experience that may even be closer to the real world situation, the compounded complexity inevitably prolonged the lab sessions and in one group case the standard three-hour lab session became an over six-hour adventuring ordeal. This complication can be largely reduced in the future when a full LabVIEW thread is implemented in the PBL enhanced curriculum. There we will have sufficient dedicated LabVIEW sessions as well as integrated lab and LabVIEW sessions on a learning curve with much reduced slope throughout the curriculum. Additionally, integrating some basic requirements of the LabVIEW certifications is also another possibility considered favorably.

The pilot effort of incorporating LabVIEW® in EE labs reported here only collected limited amount of assessment data that are specific to LabVIEW learning activities in the course of normal course evaluation. A more rigorous and specific assessment plan for the LabVIEW thread in the EE curriculum is currently being built as a part of the comprehensive assessment plan for the PBL enhanced curricula across the ET, ECET, and EE programs. The closed loop of assessment and continuous improvement will ultimately guide us in this long-term but valuable adventure to the new paradigm of engineering education for new generations to come.

Bibliography

1. Wang, J.Y.-Z., "LabView in engineering laboratory courses," *Frontiers in Education, Annual*, pp. F2E-F13, 33rd Annual Frontiers in Education (FIE'03), 2003.
2. Faraco, G. and Gabriele, L., "Using LabVIEW for applying mathematical models in representing phenomena," *Computers & Education*, vol. 49, no. 3, pp. 856-872, 2007.
3. Higa, M.L., Tawy, D.M., and Lord, S.M., "An introduction to LabVIEW exercise for an electronics class," *Frontiers in Education, Annual*, vol. 1, pp.T1D13-16, 32nd Annual Frontiers in Education (FIE'02), 2002.
4. Stouffer, W.B., Russell, J. S., and Oliva, M. G., "Making The Strange Familiar: Creativity and the Future of Engineering Education," paper 1615, *American Society for Engineering Education Annual Conference & Exposition (ASEE'04)*, 2004.
5. Winstead, V., "Applied Engineering With LabVIEW: Experiences From a Plug-in Hybrid Project," paper AC 2008-2707, *American Society for Engineering Education Annual Conference & Exposition (ASEE'08)*, 2008.
6. Backer, P. and Garcia J., "Assessment of LabVIEW and MULTISIM in the Delivery of Electronics Laboratory Content," paper AC 2007-235, *American Society for Engineering Education Annual Conference & Exposition (ASEE'07)*, 2007.
7. Zhang, Y., Akujuobi, C., Wang, Y., and Cui, S., "Engineering Technology Laboratory Enhancement With LabVIEW," paper AC 2010-1646, *American Society for Engineering Education Annual Conference & Exposition (ASEE'10)*, 2010.
8. Stone, W., Ball, A., and Howell, B., "Integrating LabVIEW® into Engineering Technology Curricula," *ASEE Southeast Section Conference*, 2008.
9. Diefes-Dux, H.A., Moore, T., Zawojewski, J., Imbrie, P.K., and Follman, D., "A framework for posing open-ended engineering problems: model-eliciting activities," *Frontiers in Education, 2004. FIE 2004*. 34th Annual, pp. F1A- 3-8, 2004.
10. Hughes, J.L.A. , "Incorporating project engineering and professional practice into the major design experience," *Frontiers in Education Conference, 2001. 31st Annual* , vol.3, pp.F3G-16-2, 2001.
11. Soundarajan, N. , "Engineering Criteria 2000: the impact on engineering education," *Frontiers in Education Conference, 1999. FIE '99. 29th Annual* , vol.1, pp.11A1/25-11A1/30, 1999.
12. Prince,M., and Hoyt, B., "Helping students make the transition from novice to expert problem-solvers," *Frontiers in Education, Annual*, vol. 1, pp. F2A7-11, 32nd Annual Frontiers in Education (FIE'02), 2002.
13. Besterfield-Sacre, M.E., Wolfe, H., Atman, C.J., and Shuman, L., "Development of Customer-Based Outcome Measures for an Engineering Program" *American Society for Engineering Education Annual Conference & Exposition (ASEE'97)*, 1997.
14. Bandyopadhyay, A., "Technology Education in the Next Century: Is the Proposed TAC/ABET Criteria Compatible?" *American Society for Engineering Education Annual Conference & Exposition (ASEE'98)*, 1998.