DEVELOPMENT OF LOW-LEVEL DIGITAL I/O EXPERIMENTS INVOLVING A HIGH-LEVEL PROGRAMMING LANGUAGE

Yanfei Liu
Department of Engineering
Indiana University - Purdue University Fort Wayne
Fort Wayne, IN 46805-1099

Abstract

In recent years there has been a shift on the type of programming language when students are taught low-level input/output (I/O) access. The shift has been from assembly language to a high-level language such as C or C++. To address this change and to prepare students well for their future career, a series of laboratory experiments covering I/O interfacing for an introductory microprocessor course has been developed. Ten laboratory experiments are now in place for this course. Five of them are software oriented assignments designed to have students become familiar with assembly language programming. The remaining five experiments are of hardware nature using a PCI DAQ board. The hardware experiments include basic digital I/O interfacing as well as A/D and D/A design. These experiments have been carried out by students that took the course in the fall semesters of 2005, 2006 and 2007. An assessment of the students' laboratory experience has been conducted and its results are summarized in this paper.

Introduction

Microprocessor architecture is a core subject in both electrical engineering and computer engineering curricula. Introductory microprocessor courses aim at providing students with enough background so they will be able to interface, design and program microprocessor based systems. The lectures complement the laboratory experiments; they introduce basic concepts that help students complete the laboratory projects. Thus a critical feature of an introductory microprocessor course is that it is a laboratory-oriented course. The laboratory experiments play a very important role in preparing students for work in industrial applications. With the rapid growth in the number of applications that use embedded microprocessor systems, there has been a shift on which programming language to use when students learn low-level input/output (I/O) interfacing. This shift has been from assembly language to a high-level language such as C or C++. Developing a series of meaningful laboratory experiments that meets this trend has then become necessary.

Since there is large variety of microprocessors in the market, faculty in charge of a microprocessor course have to pick the ones they believe will prepare best their students. Because of fast changes in microprocessor technology, laboratory courses associated with this topic have to be continuously updated. Engineering programs worldwide have investigated different microprocessors and then designed experiments that use the ones they pick [1]. Some of these efforts have been limited to assembler based software experiments e.g., Microsoft Macro Assembler (MASM) [2]. While MASM is a good tool for students to experiment with Intel X86 assembly instructions, it does not provide a meaningful hardware interfacing experience. Other institutions teach microprocessor architecture that lean heavily on hardware interfacing experiments [3]. Some other institutions, realizing that both software and hardware experiments are important for learning basic microprocessor architecture have shifted from a traditional microprocessor course, based on software experiments (e.g. using X86 assembly instructions), to one that uses a PIC microcontroller and then provided their students with software/hardware experiments [4]. While PIC microcontroller development boards have enough peripherals for hardware design, their instruction set is quite limited for programming complicated tasks. In this regard, Intel X86 based systems have a more powerful set of assembly instruction that facilitates working on more complicated tasks.

A few years ago the Department of Engineering at Indiana University – Purdue University Fort Wayne (IPFW) decided on using the Intel X86 family of microprocessor systems under the premise that Intel has been the leading manufacturer of microprocessors since 1970s. Prior to 2005, the digital I/O board used for the input/output operations in our experiments was designed for the ISA bus. Around the year of 2000 the PCI bus became dominant and squeezed out the ISA bus from most motherboards. By the year of 2005 it was almost impossible to find a computer equipped with an ISA expansion slot. To expose students to contemporary technology, in the fall of 2005 we started to develop a new set of experiments that use a PCI board on topics dealing with digital input/output operations, analog-to-digital (A/D), and digital-to-analog (D/A) conversions.

After searching through different digital PCI I/O boards we decided on a PCI DAQ board, the PCI-DAS1002, manufactured by Measurement Computing [5]. We picked this board so that students would get exposed to a commercial product that they will probably be using in the industry after they graduate. The Universal Library, included with the software that comes with the board, provides libraries for Windows Visual C/C++ programming.

Assembly language and a high level language, such as C/C++, are two different tools for embedded system design. Each has advantages and disadvantages. Without a “mysterious” intermediate compiler, assembly programming is faster and clearer. On the other hand a high level language accomplishes a given task in a more general way regardless of the differences between the assembly instruction sets of different microprocessors. Portability is of extreme importance for embedded systems. Thus using assembly language can be problematic. Due to fast changes in microprocessor technology, new designs may not use the same processor architecture and/or the same compiler as previous designs. Therefore, anytime one writes software in assembly language, the ability to reuse the code is greatly diminished. That’s why most industrial applications prefer to use C or C++ these days. By adopting the PCI DAQ board for this course our students have the opportunity to program the board in C instead of assembly language.

The remainder of the paper is organized as follows. First, the selected PCI DAQ board, its accessories, and the experimental workstation are introduced in detail. Then the newly developed experiments are described, followed by the course assessment process and its results. The conclusions are given at the end of the paper.

The PCI DAQ board and its accessories

The PCI-DAS1002 board, shown in Figure 1, is a multifunction analog and digital I/O board. This board embeds a programmable I/O device (82C55A) that has 24 I/O pins which can be individually programmed in two 8-bit ports and two 4-bit ports. Each port can be configured independently as input or output. This board also has A/D and D/A capabilities in addition to the I/O ports.

The PCI board is a plug-and-play board. After plugging it, the next step is to install InstaCal, which is the software used to conduct A/D and D/A calibration of the board. The PCI-DAS1002 comes with the Universal Library, which provides libraries for Windows Visual C/C++ programming. This library contains a variety of functions for the users to configure the board and conduct basic digital I/O operations as well as A/D and D/A conversions.

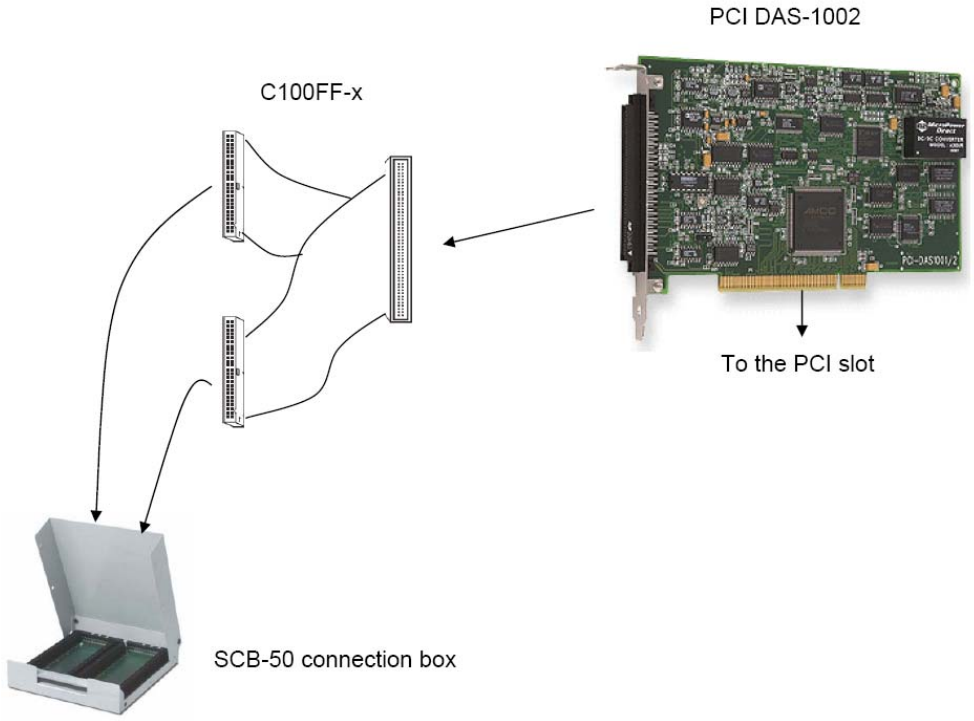


Figure 1: The PCI board and its accessories assembly.

For the students to be able to build their own circuits on a breadboard and interface with the PCI board, some extra components are still needed. With this PCI board, we also give students two other accessories: the SCB-50 connection box and the C100FF-x 100-pin high-density female connector to two 50-pin female IDC (Insulated Displacement Connector) as shown in Figure 1. Both of these accessories are made by Measurement Computing. The PCI board is connected to the SCB-50 through the C100FF-x cable. The SCB-50 connection box provides screw terminal connections for all 100 I/O pins of the PCI board. Hence, students can connect each specific pin to the breadboard and build the desired circuits for each individual experiment. Figure 2 shows the breadboard circuit that the students built for one of the experiments. Figure 3 shows the front and back view of one of the workstations used by the students. The computers are all equipped with Intel Pentium 4 processor (2.9 GHz) and 1GB RAM. The operating system is Windows XP.

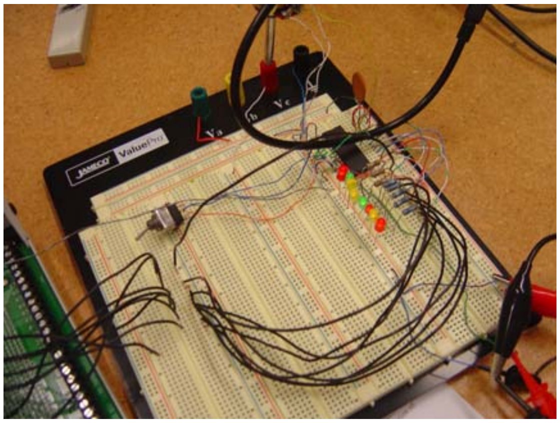
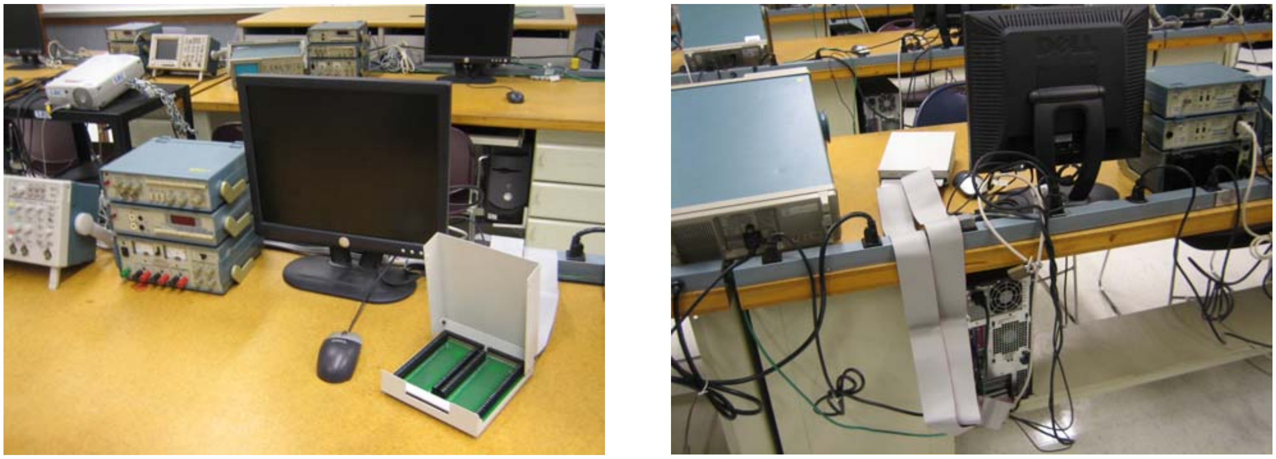


Figure 2: A student's breadboard.

Experiments

A total of ten experiments were designed for this introductory microprocessor course entitled "Microprocessor Systems and Interfacing." Each laboratory session takes 2 hours and 45 minutes. The first five experiments are software oriented and are focused on the Intel



(a) Front view

(b) Back view

Figure 3: A laboratory work station.

Table 1: Software laboratory experiments.

Lab Number	Experiment
1	Familiarity with MASM, Debug, and CodeView
2	Simple assembly language program design
3	Subroutine applications
4	Modular program design
5	Implementation of Complex Arithmetic Operations

assembly instructions under Microsoft Macro Assembler (MASM 6.xx). These five experiments are listed in Table 1. These experiments start with learning MASM and its corresponding debugging tools. Students then write assembly language programs from basic arithmetic operations such as addition and subtraction to modular program design. The last experiment requires students to write an X86 assembly program to implement a complex arithmetic operation: multiplication. Students are asked to implement the multiplication operation in two different ways. One way is to use the multiplication instructions provided in the X86 instructions set. The other way is to implement a multiplication algorithm by only using the shift and addition instructions.

The remaining five experiments are hardware oriented aimed to teach interfacing using the PCI DAS-1002 board. The titles of these five experiments are listed in Table 2. In this paper we focus on these five experiments. The first experiment of this set, learning how to use the PCI DAS-1002 board, is the foundation for the other four experiments. The relationship between experiment 1 and experiments 2-5 is illustrated in Figure 4. The primary goal of the first experiment is to learn how to program the PCI board and to understand the basic digital input/output operations. In this section, these five experiments are described in detail.

Table 2: Laboratory experiments using the PCI board.

Lab Number	Experiment
1	I/O interface design using a PCI DAS-1002 board
2	Programmable DC motor speed control system design
3	A/D, D/A interface design using a PCI DAS-1002 board
4	A/D interface design using an ADC0809 chip
5	D/A interface design using a DAC0808 chip

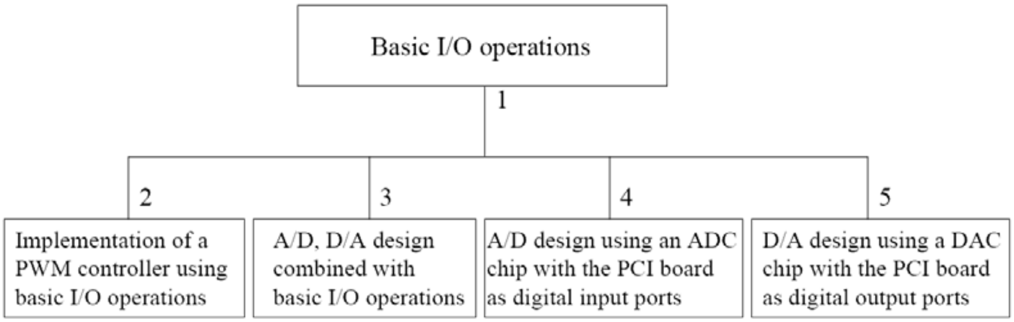

Basic I/O operations
1
2
3
4
5
Implementation of a PWM controller using basic I/O operations
A/D, D/A design combined with basic I/O operations
A/D design using an ADC chip with the PCI board as digital input ports
D/A design using a DAC chip with the PCI board as digital output ports

Figure 4: Relationship between experiments.

Experiment 1

The first experiment is an informative nature. It is about basic digital input/output (I/O) interface design using the 82C55A device built into the PCI DAS-1002 board. The objective is to have students become familiar with the process of programming, interfacing the PCI board, and understanding basic digital input/output operations. The 82C55A contains two 8-bit ports (PORTA and PORTB) and two 4-bit ports (PORTCL and PORTCH). In this experiment, students build two different circuits for digital input and output operations separately and then connect the circuits with the PCI board. The digital output circuit [2], shown in Figure 5, interfaces eight resistors and LEDs with PORTA of the 82C55A to represent the digital output data from PORTA. The digital input circuit [6], shown in Figure 6, connects the DC voltage source (5V) through the 8-bit DIP switch and resistors to the pins of PORTB of the 82C55A to generate a digital input signal to PORTB.

On the software side, the sample C programs provided by Measurement Computing for digital input/output are modified and given to the students. The digital input program, renamed as “in_bytes.c”, reads the digital input from the 8-bit PORTB and displays each bit and the corresponding decimal value on the monitor screen. There are two programs for the digital output operations: “send_bits.c” and “send_bytes.c”. The “send_bits.c” program enables the user to set/reset any specific pin of the 8-bit PORTA. The “send_bytes.c” asks the user to enter a decimal number from 0-255 and then sends the input decimal number to PORTA. Both programs can only receive input and output the signal once. After the signal is sent to the port, the program exits automatically.

Students follow a detailed procedure to create a new project in Visual C/C++, add the provided programs to the project, and change the configurations to add the directories of the header files and library files provided in the Measurement Computing software. Students then compile and build the project. After completing the building process, students run the executable files. For the “send_bytes.c” and “send_bits.c” programs, students observe the results using the circuit shown in Figure 5 and check if the LEDs lighten up as anticipated. For the “in_bytes.c” program, the students use the DIP switch in the circuit shown in Figure 6 to select different input values and observe if the results on the screen match the actual input.

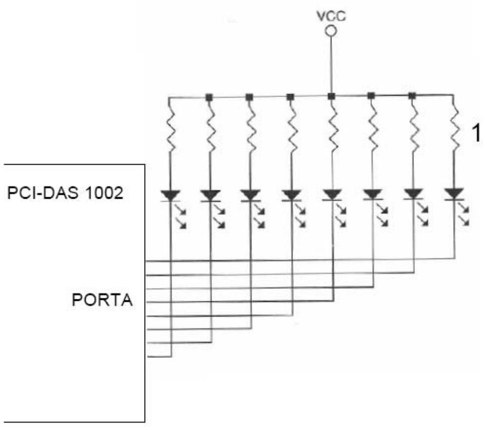


Figure 5: Basic output interface with LEDs.

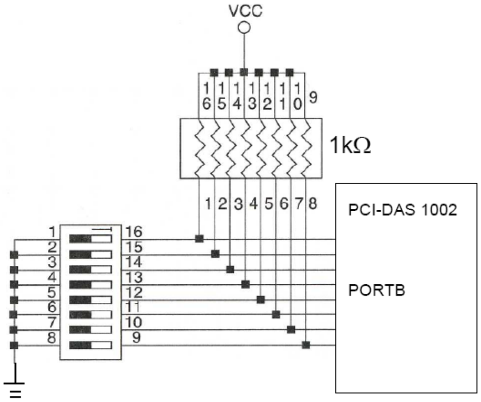


Figure 6: DIP switch circuit.

Once students understand how to program the PCI board for digital input/output operations, they are required to complete two assignments. The first assignment is to modify the “send_bytes.c” and “send_bits.c” programs so that the signal is only output for 2 seconds. The modified code should also allow the users to reconfigure the bits or bytes by prompting a message, such as “Do you want to exit or not?” The second assignment is to combine the “in_bytes.c” and “send_bytes.c” programs and modify the new program such that the resulting code reads the data from the DIP switch through PORTB, displays this data on the monitor screen and outputs the data through PORTA to turn on the appropriate LEDs.

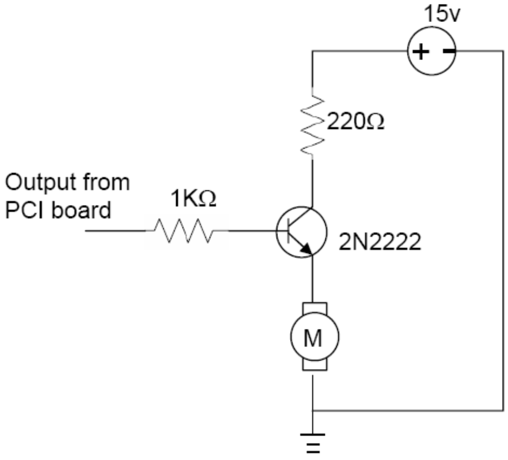


Figure 7: Motor driving circuit.

Experiment 2

The second experiment is to have students use their knowledge to implement a particular application. The application is to design a programmable DC motor speed control system using pulse-width modulation (PWM). PWM is a well known technique used to control the speed of a motor. The duty cycle of the PWM signal, the “on” time within a period divided by the period time (T), determines the motor speed. In this experiment, the driving circuit for the motor (as shown in Figure 7) is given to the students. Students are allowed to use the output from any of the digital ports of the PCI DAS1002 board to drive the motor. The students’ code should ask the user to input a number from 0 to 9. Each input corresponds to a different duty cycle of the PWM signal, i.e. different speed of the DC motor. Students then observe how the speed of the motor changes. In this experiment students are required to try three different periods T = 10ms, 100ms, 200ms with different duty cycles for each period and then observe the different responses.

This experiment also requires students to connect an 8-bit DIP switch to another digital port of the PCI board and configure this port as an input port. Students are asked to write a program to allow the user to use the DIP switch to select one of the 255 different possible speeds, and display the decimal and corresponding hex value on the monitor. For six different switch settings corresponding to the decimal values 1, 50, 100, 150, 200, and 255, students need to measure the voltage on the terminals of the DC motor and observe how the speed of the motor changes.

Experiment 3

This experiment is about A/D, D/A interfacing design using the A/D and D/A converter embedded in the PCI-DAS 1002 board. The PCI-DAS 1002 board has one 12-bit A/D converter (ADS7800) and one 12-bit D/A converter (AD7847). The PCI-DAS 1002 board provides either 16 single-ended or 8 differential analog inputs. The input ranges are software-selectable as either bipolar (±10V, ±5V, ±2.5V, and ±1.25V) or unipolar (0 -10V, 0 - 5V, 0 - 2.5V, and 0 - 1.25V). The PCI-DAS 1002 also provides two channels of 12-bit analog output. Analog output ranges are also software-selectable as either bipolar (±10V and ±5V) or unipolar (0-10V, 0-5V).

The first task is to conduct an analog test using InstaCal. Students are required to conduct analog tests under 8 differential and 16 single-ended analog input modes separately and observe the difference. The actual ranges of the test signals help students understand better about these two different modes.

The second task is the A/D design. The program "analog_in.c" that is used to read in the analog data is provided. The function of the "analog_in.c" program is to read-in an analog voltage signal, perform A/D conversion and display the analog voltage signal and the corresponding decimal value on the monitor screen. Even though the code is provided, students need to read through the code and figure out how the code works and how the "read analog data in" function works in order to accomplish the A/D design task described in the next paragraph. For the software task, students create a new project in Visual C/C++, add "analog_in.c" to the project and complete the necessary steps learned in *Experiment 1* to compile and build the project. For the hardware part, students are asked to connect a variable DC voltage source (5V) to the analog input channel 0 of PCI-DAS1002, run the program, increase the voltage at 0.25V intervals starting from 0 and record the corresponding decimal integer output from the monitor screen.

For the A/D design part, students are also required to show the converted digital data by lightening up the LEDs. In order to represent a 12-bit binary data, two digital ports from the 82C55A are used. Hence, continuing from the previous task, students modify the "analog_in.c" program to configure the 8-bit PORTA and 4-bit PORTCL as output ports, split the converted digital data (a 12-bit data) into an 8-bit one and a 4-bit one, and then output the 8-bit and 4-bit values through PORTA and PORTCL. A circuit that has the same structure as the one shown in Figure 2 except using 12 resistors and LEDs needs to be built and then connected to each pin of PORTA and PORTCL. Students then run their program, slowly adjusting a variable DC voltage source (5V) from 0 to 5V, and observe the different LEDs' patterns.

The third task is about a D/A design. Students start by modifying the digital input code "in_bytes.c**"** to write the digital input value to the D/A output channel 0. For the hardware part, students connect the DC voltage source (5V) through the DIP switch and resistors to the pins of PORTB as shown in Figure 3. Students then run their program, use the DIP switch to select six different settings corresponding to the decimal values 1, 50, 100, 150, 200, and 255, and record the output voltage signal from the D/A output channel 0. In the lab report students are required to plot the analog outputs as a function of the decimal value and explain their observations.

Experiment 4

This experiment is about interfacing an ADC0809 chip [7] with the microprocessor via an 82C55 (built in the PCI DAS-1002 board). The ADC0809 will be used to implement a positional sensor based on a potentiometer. For the hardware task, students wire up the circuits [8] as shown in Figure 8. For the software task, students write a program to configure PORTB as the input port, continuously check PORTB and display the value on the monitor in both binary and decimal numbers until the user enters a key to exit. When the students run the program, they are asked to increase the voltage at 0.5V intervals starting from 0 and record the corresponding decimal output from the monitor screen. In the lab report students are asked to plot the supply voltage as a function of the decimal output, explain what kind of relationship there is between them and explain their observations.

Experiment 5

This experiment is about D/A interface design using a DAC0808 chip [9]. Students are required to wire the DAC0808 as shown in Figure 9, construct a digital input circuit using a 5V power supply, a DIP switch and resistors, and connect the circuit to the digital inputs of the DAC0808. Then they adjust the inputs and measure the output voltage to verify that the D/A conversion circuits work properly.

After successfully constructing the DAC0808 circuit, students need to disconnect the digital input circuit from the DAC0808 and connect the 8 digital input pins of DAC0808 with PORTB in the PCI board. For the software task, students need to write a program to configure PORTB as the output port, send out a decimal value (0-255) through PORTB, and display the value on the monitor screen. Students need to record the output voltage for 0, 50, 100, 150, 200 and 255, plot the output voltage as a function of the decimal input, explain what kind of relationship is between them and explain their observations.

Laboratory assessment

These five experiments were carried out by students who took the microprocessor course in the fall semesters of 2005, 2006 and 2007. During the fall of 2007, a total of 12 students took this course. A survey was conducted at the end of the semester and the following (Table 3) Likert scaled questions were given [10]. The students were asked to rate their answers from 1 (not at all) to 5 (very much). There were a total of eight students who took the survey. Five of them majored in electrical engineering, two of them were in computer engineering, and one of them was in a dual degree program (electrical engineering and computer engineering). Table 3 lists the questions and the average scores for each individual question.

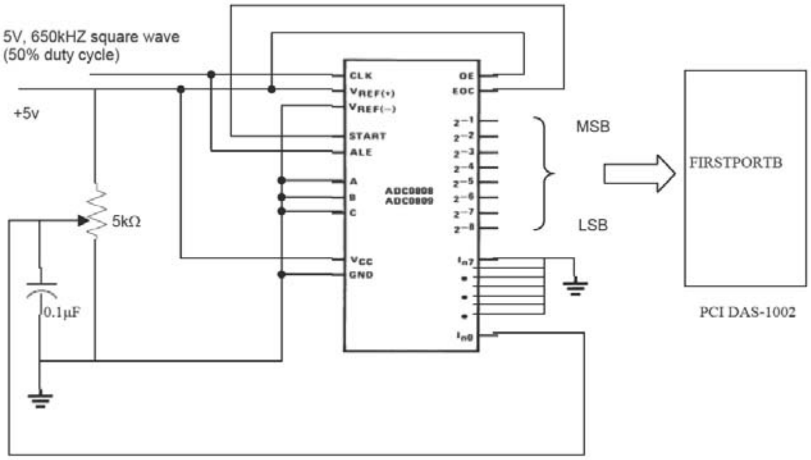


Figure 8: ADC0809 interfacing pin connections.

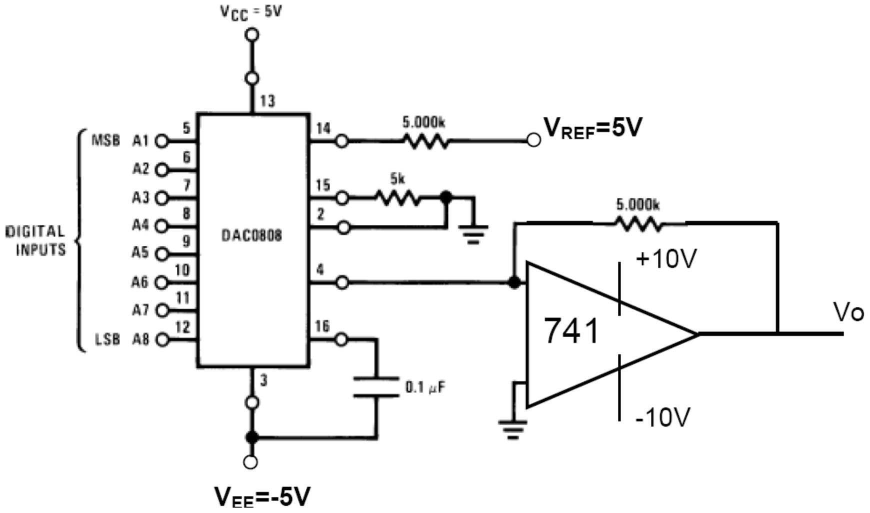


Figure 9: DAC0808 interfacing pin connections.

Table 3: Average scores of the survey questions.

Questions	Average scores
1. The PCI board is easy to work with. If not, please specify.	4
2. These experiments expose you to a way how data acquisition is done in real industrial applications. If not, please elaborate.	4.3
3. These experiments are reasonable in content and length. If not, how can we improve it?	4.6

For the first question, some students complained about Visual C++ and that one PCI board had a nonfunctional port A. Since there was a spare board, the students were able to be to relocate to another workstation and complete the experiments successfully.

From the assessment results, we can conclude that the students were very satisfied with these experiments. In addition to using these survey results, there is also one instance that validates our initial intention of using a commercial PCI board. One of the students who took this course in the fall of 2005 used a similar PCI DAQ board manufactured by Measurement Computing in his capstone project to design and build a Software Controlled Radio. The experiments in this course provided him with important knowledge for his senior design project.

Conclusions

This paper describes the efforts of introducing a PCI DAQ board into the laboratory experiments of an introductory microprocessor course. The PCI DAQ board chosen is the PCI-DAS 1002, manufactured by Measurement Computing. Based on this board, five experiments were designed. These five experiments involve basic digital input/output interfacing using 82C55A, A/D and D/A designs. The experiments are described in detail. This paper also summarizes a student survey that was conducted to assess their laboratory experience. The results of the survey showed that the students were very satisfied with the laboratory experiments.

References

1. H. Broberg and E. Thompson, "Selection of Processor, Language, and Labs in Introductory Microprocessor/Microcontroller Courses," in *Proceedings of the 2005 American Society of Engineering Educators (ASEE) Annual Conference,* June 12-15, 2005, Portland, OR.

2. S. He, "Laboratory Design for Introductory Course of Microprocessors," in *33rd ASEE/IEEE Frontiers In Education Conference,* November 5-8, 2003, Boulder, CO.

3. J. Jeon, "A Microprocessor Course: Designing and Implementing Personal Microcomputers," *IEEE Transactions on Education.*, vol. 43, no. 4, pp. 426-433, November 2000.

4. R. Reese, "Embedded System Emphasis in an Introductory Microprocessor Course," in *Proceedings of the 2005 American Society of Engineering Educators (ASEE) Annual Conference*, June 12-15, 2005, Portland, OR.

5. Measurement Computing, http://www.measurementcomputing.com/, (accessed October 2008)

6. B. Brey, *The Intel Microprocessors 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, and Pentium Pro Processor Architecture, Programming, and Interfacing,* 7th edition, Prentice Hall, 2005

7. ADC0808/0809 datasheet, http://www.national.com/ds/DC/ADC0808.pdf, (accessed October 2008)

8. H. Broberg, EET302 laboratory manual, Indiana University – Purdue University Fort Wayne.

9. DAC0808/0809 datasheet, http://www.national.com/ds/DA/DAC0808.pdf, (accessed October 2008)

10. Likert, Rensis, "A Technique for the Measurement of Attitudes," Archives of Psychology 140: pp. 1-55, 1932

Biographical Information

Yanfei Liu received the B.S. degree in Electrical Engineering from Shandong Institute of Architecture and Engineering, Jinan, China in July 1996. She then received the M.S. degree from Institute of Automation, Chinese Academy of Sciences, Beijing, China in July 1999, and the Ph.D. degree from Clemson University in August 2004. Dr. Liu has been a faculty member of the IPFW Department of Engineering since August 2005. Her research and teaching interests are robotics and microprocessor systems.