AC 2008-748: A MICROCONTROLLER APPLICATIONS COURSE AND THE FREESCALE'S MICROCONTROLLER STUDENT LEARNING KIT

Chiu Choi, University of North Florida

A Microcontroller Applications Course and Freescale's Microcontroller Student Learning Kit

Abstract – This paper describes the improvement of a microcontroller applications course by the adaptation of a new microcontroller development tool- Freescale's microcontroller student learning kit with CSM12C32 module for the course. This paper also describes the pedagogy for this course and the student evaluation results. The microcontroller applications course is a junior/senior level course offered in the electrical engineering department. The objective of the course is to teach the students how to design microcontroller-based systems. The particular microcontroller used in this course at the present time is the Freescale MC9S12C32. The integrated development environment used is CodeWarrior Development Studio for HCS12. It supports assembly, C and C++ programming. The major course outcomes were: first, with the adaptation of the Freescale kit the design flow has become simpler and smoother than before. Elements of the design flow such as editing of programs, compiling, flashing, debugging and re-flashing of the microcontrollers are done in one development platform- CodeWarrior. Second, the form factor of the CSM12C32 module is small enough that our microcontroller projects have become truly embedded systems. Third, the skills and the tools covered in this microcontroller course are useful to the students that they prefer to use them in their capstone design projects. Fourth, the student evaluation results of this course taught by the author have been among the highest in our department and the college. This microcontroller course with the Freescale student learning kits is a viable approach for equipping students with the skills and tools that they need for prototyping embedded systems and for preparing them for their future design projects.

I. INTRODUCTION

The microcontroller course is one of the most demanding courses in our undergraduate electrical engineering curriculum. The prerequisite for this course is an introductory course on digital design. The microcontroller course covers the fundamentals of microcontrollers with emphasis on hardware interfacing, software design, and applications. Topics include microcontroller software architecture, assembly instruction set, addressing modes, memory map, general purpose inputs/outputs (GPIO), analog-to-digital converters (ADC) , timers, input capture, output compare, pulse-width modulators (PWM), serial communication interfaces, and interrupts. This course also gives students the training necessary to effectively use an integrated development environment (IDE) for developing their application programs in assembly language and C. Many of these topics are discussed in a number of textbooks[1,2,3,4,5] and data books[6,7,8].

A number of different microcontrollers were evaluated for meeting the objectives of this course. The particular microcontroller selected was the MC9S12C32 from Freescale Semiconductors. This is a general purpose microcontroller that is simple enough for the students to learn its operation quickly and with on-chip peripherals that are sophisticated enough for covering a wide range of applications. The microcontroller is supported by a powerful software development tool-Metrowerks CodeWarrior IDE. A project board for flashing the microcontroller is available. The project board supports CodeWarrior and has basic electronic components such as switches, LED's, dc voltage supplies, and a breadboard for prototyping. The project board, the microcontroller and the CodeWarrior IDE are packaged as a Student Learning Kit (SLK) offered by Freescale Semiconductors. The SLK's were first used in Spring 2006 in our microcontroller applications course.

Before the adaptation of the Freescale kit for this course, we had used for four years the Motorola MC68HC912B32 microcontrollers (evaluation board model # CME-12BC from Axiom Manufacturing). The software development tools used were DBUG12 Monitor, AXIDE (for programming the flash and other functions), ImageCraft C compiler for developing applications in C, and other software support and utility tools. This set of tools was very helpful but were not integrated into one environment. This might cause extra effort in the development process occasionally. For example, handling of interrupts on DBUG12 required extra effort such as pseudo interrupt vectors. In the programming of the microcontroller EEPROM, the pseudo vectors must be replaced by the actual interrupt vectors. This is an example of extra effort that is not required if Code Warrior is used.

The rest of this paper is organized as follows: Section II covers briefly the pedagogical approach for the course. Section III outlines the lecture materials. Section IV describes the labs briefly. Section V discusses the class projects. Section VI discusses the advantages of using the Freescale's microcontroller experimentation kit. Section VII provides the student feedback. Section VIII describes the lesson learned and recommendation. Section IX provides concluding remarks.

II. PEDAGOGICAL APPROACH

The pedagogical approach to achieve the objective of this course consists of several stages. The objective is to teach the students to be able to design microcontroller-based systems. The first stage is to educate the students on CPU programming model, common addressing modes, and assembly language programming. This is done by lectures and by demonstration of the assembly program execution in the CodeWarrior debugger. The debugger allows the users to see the source assembly program, the assembled program, the CPU registers, the contents of RAM and Flash, the data of the program, and other information in a number of windows on one screen. The effects of assembly instructions on the CPU registers and memories are clearly demonstrated to the students in the debugger by executing sample programs line by line. The students gain further understanding of programming model, addressing modes and assembly language programming by performing a set of labs with guidance.

The second stage is to educate the students on the microcontroller peripherals. This is done by lectures and labs. The particular peripherals covered include general purpose input/output, timers, analog-to-digital converters, pulse-width modulators, serial communication interface, and interrupts. In the lectures, the coverage of each of these on-chip resources begins with their functional description followed by detail explanation of the control, status, and data registers (i.e., how to configure and use them). The Processor Expert of CodeWarrior is not used so that the students will gain the experience of programming these registers on their own. Programs were developed for showing the students the applications of the peripherals. Execution of these programs in the debugger is shown to the students in the lectures. The students can see clearly the benefits of these microcontroller peripherals. The application programs were written in both assembly and C, which are often shown side by side to the students so that they will appreciate the differences and similarities in developing application programs in both languages. These programs are distributed to the students and they use them in their class projects later. Further, the students are required to configure and use the peripherals in the labs by writing their own programs in assembly and C.

The third stage is to educate the students on the development of microcontroller-based applications. This is done by guiding the students through their class projects and by teaching them how to interface microcontrollers to sensors, displays, and actuators in the lectures and demonstrations. Students are required to submit proposals for their projects. The proposals were evaluated and approved by the instructor based on the project complexity and cost. The students received help in selecting electronic parts, sensors, actuators, and other components from the instructor. Periodic project updates and final project reports were required in addition to PowerPoint presentation of the projects. A good number of projects were successfully built for this course. The project reports document that the students used the skills that they had learned in the lectures and the labs.

III. Outlines of the Lectures

The documentation from Freescale[6,7] were used as the text books for this course. They were selected because back in spring 2006 there were very few text books written particularly for the HCS12 microcontrollers. The materials covered in the Freescale documentation can now be commonly found in microcontroller textbooks[2,3,4].

The lectures began with an overview of the S12 CPU of the HCS12 microcontrollers. The purposes of the internal registers and flags, data types, memory map are clarified.

The second topic was the addressing modes. All modes of addressing supported by the CPU were covered. Extra examples on the indexed modes were given to the students to help them understand the various pre/post decrement/increment operations, the accumulator offsets, and the differences and applications of the 5-bit, 9-bit, 16-bit versions of indexed modes.

The third topic in the lectures covered the common assembler instructions including load and store instructions, transfer and exchange instructions, addition and subtraction instructions, multiplication and division instructions, decrement and increment instructions, compare and test instructions, Boolean logic instructions, complement and negate instructions, bit test and manipulation instruction, shift and rotate instructions, branch instructions, jump and subroutine instructions, interrupt instructions, index manipulation instructions, stacking instructions, pointer and index calculation instructions, and stop and wait instructions. The operations of some of these instructions with various addressing modes were demonstrated to the students through the execution of assembler programs on Code Warrior in class. The students saw the effects of these instructions on the CPU registers, the flags, and the memory.

The fourth topic covered the format of an assembler program and the procedures for using CodeWarrior for developing and debugging assembler programs and for downloading into the flash of the microcontrollers. Demonstrations were given to the students in the lectures showing students the steps involved in using CodeWarrior for such applications. The focus was then shifted from the CPU to the peripherals.

The fifth topic covered the GPIO, which was our first topic for the peripherals. The students were taught how to configure various ports on the microcontrollers to function as input and output pins. Demonstrations of GPIO for sending signals out and reading signals in through execution of programs in assembly and C on CodeWarrior were given to the students. These programs were also sent to the students for them to use in their labs and projects.

The sixth topic covered the serial communication interface (SCI). The students were taught how to configure the control registers for setting various baud rates, modes of operation, and data formats. They also learned how to use the status registers for checking the state of communication between the microcontrollers and the external device. Two demonstrations were given to the students in class. The first was using the SCI to communicate with a PC through a com port on the PC. The second was showing how to display strings on a serial LCD. The demonstration programs were developed in both assembler and C and were executed on CodeWarrior. They were given to the students for their use in the labs and projects.

The seventh topic covered the ADC. This topic was included in the syllabus so that the students would be able to use it for reading sensors such as semiconductor temperature sensors, potentiometers, accelerometers and other sensors with analog outputs. This has enlarged the set of components that the student could use for their projects. In the lectures, the students learned how to configure the ADC control registers for various modes of operation such as single or continuous conversion, single channel or multi-channels operations, sampling rate, resolution, conversion time, and other features. Application programs for ADC in both assembler and C were demonstrated on CodeWarrior for the students in the lectures. These programs were also given to the students to help them do the labs and their projects.

The eighth topic covered the PWM. This topic was included so as to give the students the skills necessary for driving actuators such as dc motors in their labs and projects. In the lectures the students learned how to configure the control registers for setting frequencies and duty cycles of the PWM signals. Various features such as pulse polarity, alignment of PWM signals, resolution, and clock sources were also covered. Like the previous topics, PWM application programs in assembler and C were executed on CodeWarrior for demonstration purposes for students to see the benefits of this peripheral. These programs were also given to the students for their use in the labs and projects.

The ninth topic covered logic analyzer. The topic was useful for debugging of the hardware. The students were taught how to use Tektronix TLA611 logic analyzers. The students used them for analyzing the timing of PWM signals and for debugging their application programs.

The tenth topic covered interrupts. Real time interrupt, IRQ interrupt, SCI interrupt, and other peripheral interrupts were covered. The students learned how to configure the control register to operate such interrupts. Demonstration on CodeWarrior of interrupt programs were given to the students.

Topics such as input capture and output compare were also covered when time permitted. There are other useful features on the microcontrollers such as controller area network, background debug mode, and flash modules. There was not enough time for this first course on microcontrollers to include them.

IV. Labs

Students learned practical skills by doing the labs. They applied the lab skills to the construction of their projects. There were twelve labs for the students to do. The students had one week to complete every lab. They were required to turn in a weekly lab report. The contents of the labs are briefly described below.

In the first lab the students were given a guided tour of the CodeWarrior integrated development environment. The students learned how to create and build their assembly and C programs, to flash the program into the MC9S12C32 microcontrollers (on the CSM12C32 module from Axiom Manufacturing) and to execute the program.

In the second lab, the students built their own MC9S12C32 modules. They soldered the electronic components onto unpopulated PCBs designed for this course. Soldering instructions were given to them. They tested the modules that they built and used them in subsequent labs. The purpose of this lab was to give the students experience of building their own modules and to learn about different packaging such as surface mount and DIP for various electronic parts through the soldering process.

In order for the students to understand the internal operations of a microcontroller and to be able to develop their own programs in assembly, the next three labs covered intensive assembly language programming and debugging assembly language programs in CodeWarrior. The students also wrote many assembler subroutines that they would use later on in their labs and projects.

The sixth lab was on the serial communication interface. The students learned how to develop a set of input-output utility subroutines for sending data to and receiving data from a PC through the SCI port of the microcontroller. The techniques learned by the students in this lab were used later in their projects in which data was displayed on serial LCDs and for reading sensors with ASCII outputs.

In the seventh lab the students learned how to use GPIO for reading incoming digital signals to the microcontrollers and for sending signals out of the microcontrollers. Lab exercises included generating a sequence of pulses of varying duty cycles, driving LEDs and reading the states of switches. The skills learned were applied to turning on and off dc motors and other actuators.

The students were taught in the eighth lab how to use a logic analyzer. The particular logic analyzers used in the lab were Tektronix TLA611, which is a 34-channel logic analyzer with 500 picoseconds timing resolution on all channels. Students learned how to operate the equipment and to select the parameters for viewing signals of different pulse width and frequencies. They learned the effect of the sampling time and memory depth on the captured waveforms. The waveforms could be easily saved into USB flash drives and were later inserted into their lab reports.

The ninth lab was to learn how to configure and use the on-chip analog-to-digital converters. Students learned how to design a microcontroller-based on-off control system in this lab. The microcontroller read analog sensor inputs through the ADC. Depending on the analog value, the microcontroller generated digital outputs on the GPIO pins for turning on or off a dc motor.

The students also learned how to configure the on-chip PWM for generating pulse width modulated signals in the tenth lab. They used the PWM for motor control. They built a microcontroller-based system that read an analog sensor input and proportionally controlled the duty cycle of a PWM signal. The PWM signal drove a motor driver chip that controlled the speed of a dc motor. The motor driver used was A3953 made by Allegro MicroSystems.

The students then moved on to learning internal and external interrupts and were shown how to do it in both C and assembly in the eleventh lab. Students also learned how to configure the timer system for real-time interrupt.

V. Class Projects

The purpose of the class project was to give an opportunity for the students to apply the skills learned in the lectures and the labs to built meaningful embedded systems. Each project used the MC9S12C32 microcontroller, sensors and actuators. The most used on-chip peripherals in the projects were GPIO, ADC, PWM, and the serial communication interface.

There were twenty projects built in the spring 2006 semester. That term was the first term that the Freescale microcontroller kit was used. Each of the projects was done by a group of two students except a few that were done individually. The titles of projects were: interfacing accelerometer to HCS12 microcontroller, obstacle avoidance robot, collision warning system, bicycle speedometer, credit card reader, digital automobile display, SPI communication between two microcontrollers, entry control system, fire protection system, HCS12-based home security system, interfacing a keypad and a LCD to a HCS12 microcontroller, Katakana quiz hand-held game, 9S12E128 printed circuit board design, data storage using an EEPROM, proximity parking sensor, monitoring of a water pump for a drinking well, smart fan, temperature warning system, and an electronic glow bug.

Some of these projects were challenging. For example, in the water pump monitoring project, the microcontroller-based system could detect and register water pump activity. Such a system could be used to detect a hidden leak in a water pipe indicated by registered water pump activity when there should be none. In the credit card reader project, a portable, microcontroller-based magnetic card reader was built. It could read the magnetic strip of a credit card, process the data received, and display the contents of the magnetic stripe on an LCD display. In the Katakana quiz hand-held game project, the microcontroller displayed Japanese katakana characters on a LCD with multiple choice questions and could receive the answers through the push buttons.

In the spring 2007 semester, seventeen projects were built with the Freescale kit. The projects were: digital compass, autonomous robot, a prototype of an attic cooling system, monitoring of fuel, a prototype of a home automation system, monitoring of ac voltage on the internet, interfacing a microcontroller to a PDA using Bluetooth, interfacing to thermostats, obstacle avoidance robotic vehicle, a smart parking monitoring system, pet food dispenser, piggyback air fuel correction system, proximity sensor, robotic ant, a smart alarm clock, valve regulation controller, and water level controller.

Some of the projects were interesting, for instance, the project of interfacing the 9S12C32 microcontroller to a PDA using Bluetooth. In the project, a prototype was built that commands sent from a PDA via Bluetooth was received by the microcontroller. The microcontroller then performed a specified action for that command. Another interesting project was a smart garage monitoring system. The system could monitor the number of vehicles entered into and out of the prototype garage and display the number of available parking spaces on a LCD.

In the summer 2007 term, twelve class projects were built. The projects were: access control using RFID, elevation and azimuth control using stepper motors, automatic stabilizing platform, automatic motion sensor based sliding door opener, color detector, digital thermostat, motor controller, multifunction GPS, parking warning system, robot charger, temperature sensor and fan control, touch screen scramble pad, and ultrasonic scanner. The project of access control using RFID was interesting. In this project, the microcontroller read the RFID tag through the RFID reader, checked the validity of the tag, and gave or denied access accordingly. Another interesting project was an ultrasonic scanner. The microcontroller drove an ultrasonic distance sensor mounted on a servo motor from zero to 180 degrees and read the output of the sensor. When an object was detected by the sensor, the microcontroller displayed the object's azimuth location and distance on a LCD.

VI. Advantages of Using the Freescale Experimentation Kit

The Freescale's microcontroller experimentation kit came into the market about three or four years ago and has gone through two upgrades. The kit contains a project board that can program a wide selection of microcontrollers and can be used in conjunction with CodeWarrior for assembly, C, and C++ program development, debugging, and other capabilities. The kit also contains a microcontroller evaluation module and the CodeWarrior software package in one box.

With the microcontroller module, programmer, and the software development platform in one place, this allows the students to develop microcontroller-based systems without using additional tools. CodeWarrior is a powerful tool for program development and debugging. The basic features of it can be learned in a short time. The microcontroller module is a complete single board computer that the students can use directly in their projects. The form factor for the module is smaller enough that the projects become truly embedded systems. The project board comes with a programmer that can flash the microcontroller (9S12C32), in addition to its prototyping capabilities. This kit is an inexpensive tool for learning the fundamentals of microcontrollers and for building microcontroller-based projects. It is also useful for experimentation with the peripherals because of the prototyping resources available on the project board. Many of our students used this kit in their capstone design projects.

VII. STUDENTS FEEDBACK

The Freescale kits with CodeWarrior have been used in the three semesters of spring 2006, spring 2007, and summer 2007. In the lectures and the labs, the Freescale kits were used as the tool for the facilitation of learning. Much of the materials covered in the lectures were delivered to the students through the Freescale kits and CodeWarrior. University administered student surveys were conducted for the microcontroller course for these three semesters. Spring 2006 was the first semester that this tool was used. The raw data of these evaluations are shown in the Appendix. A summary of the data for these three semesters is shown in Table 1. For comparison purposes, the evaluations of the same course taught in summer 2003, spring 2004, and fall 2004 semesters before the adaptation of the Freescale kits are also shown in Table 1. The tools used in these three semesters were Axiom's CME-12BC evaluation boards, DBUG12 Monitor, AXIDE, and ImageCraft C compiler. These tools were used as the means for the facilitation of learning for the course. Much of the materials delivered to the students in the lectures were done through these tools. Further the students used them extensively in their labs for learning purposes.

It is noticed that the evaluations of the facilitation of learning for the course using the Freescale kits and CodeWarrior are higher than that of the same course using the other tools. There are several reasons for the improvement. First, the students commented that they preferred to use the module came with the Freescale kit because it was smaller (1.5"X2") and could be easily encapsulated with other electronics inside one small box for their projects. The form factor of the previous module (Axiom CME-12BC evaluation board) is about 6"X6.5" and is consider as big for their embedded projects.

Second, the debugging process is made simple by CodeWarrior and the MC9S12C32 module. The downloading process is done by just one click. Using the former software tools (DBUG12 Monitor, AXIDE, and ImageCraft C compiler), it took longer time for debugging because the tools are not integrated together like CodeWarrior. The downloading process is lengthier than CodeWarrior.

Third, the use of pseudo interrupt vectors is necessary by the former tools. There is no such requirement for the Freescale tools.

Table 1: evaluation on the facilitation of learning

Term	Summer 2003	Spring 2004	Fall 2004	Spring 2006	Spring 2006	Summer 2007
Rating for facilitation of learning (5 is the highest score)	4.13	4.36	4.35	4.79	4.82	4.67

Another observation on the data is that the overall rating of the instructor was improved, as indicated in Table 2. This may have a correlation with the adaptation of the Freescale tools. The course materials and the labs for this course were mostly the same before and after the adaptation of the Freescale kits. It was the same instructor teaching this course in the six semesters. The only thing changed was the tools.

Table 2: evaluation of instructor

Term	Summer 2003	Spring 2004	Fall 2004	Spring 2006	Spring 2006	Summer 2007
Overall rating of instructor (5 is the highest score)	4.13	4.53	4.67	4.93	5	4.78

Interestingly the numerical ratings for this course were among the top in the department and the college for those semesters that the Freescale tools were used. The written comments from the students were also very positive for the course with the new tools.

VIII. Lessons Learned and Recommendation

Even though most of the students learned the operations of CodeWarrior quickly, a few students needed a well written tutorial on the development tool. It is recommended that the instructor prepare a CodeWarrior tutorial that is pertinent to the labs and projects covered in the course.

Many students needed help on understanding the operations of assembly instructions. Help was provided to them by showing them the effects of the execution of each assembly instruction in the debugger on CodeWarrior. The students saw how assembly instructions modified the internal registers and the flags. This helped them visualize the operations carried out by the assembly instructions.

The data books[6,7] for the microcontroller were used as the text. This provided the students the opportunity for reading data books from the manufacturer and for learning specialized information about the particular microcontroller. However, additional teaching materials were needed and developed since the data books were not written specifically for students.

IX. Concluding Remarks

The Freescale kits were useful as a learning tool for microcontrollers and as a means for building microcontroller-based projects. They are easier to learn as compared to the previous tools. Most of the senior students preferred to use the Freescale kits in their capstone design projects. The student evaluations on the facilitation of learning and the overall rating of the instructor were improved after the adaptation of the new tools. The ratings were among the highest in our College. It appears that the Freescale kits are beneficial for a first course on microcontrollers.

REFERENCES

[1] Almy, T., *Designing with Microcontrollers the 68HCS12*, Rev. 1A, 2005.
[2] Cady, F., *Software and Hardware Engineering Assembly and C Programming for the Freescale HCS12 Microcontroller*, 2nd Ed., Oxford, 2008.
[3] Huang, H.-W., *The HCS12/9S12: An Introduction, Software & Hardware Interfacing*, Thomson Delmar Learning, 2006.
[4] Pack, S.F. and D.J. Barrett, *Embedded Systems*, Pearson Prentice Hall, 2005
[5] Todd Morton, *Embedded Microcontrollers*, Prentice Hall, 2001.
[6] *CPU12 Reference Manual*, Freescale Semiconductors, 2000
[7] *68HCS12 Data Sheet*, Freescale Semiconductors, 2005.
[8] *Freescale SLK user manuals*, Freescale Semiconductors, 2005.

APPENDIX

Table 3: Evaluation of the microcontroller course adapted with the Freescale student learning kit for spring 2006

Item ID	Item	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	NR / NA	Mean	
R01	Communicates effectively with students	78.57	21.43	0.00	0.00	0.00	0.00	4.79	
R02	Enthusiasm for course material and teaching	92.86	7.14	0.00	0.00	0.00	0.00	4.93	
R03	Mastery of the course content	100.00	0.00	0.00	0.00	0.00	0.00	5	
R04	Relates course material to current examples	78.57	21.43	0.00	0.00	0.00	0.00	4.79	
R05	Clearly explains complex concepts and ideas	71.43	28.57	0.00	0.00	0.00	0.00	4.71	
R06	Lectures organized and provide framework for learning	92.86	7.14	0.00	0.00	0.00	0.00	4.93	
R07	Course syllabus accurately described the course	92.86	7.14	0.00	0.00	0.00	0.00	4.93	
R08	Course instructional materials used effectively	78.57	21.43	0.00	0.00	0.00	0.00	4.79	
R09	Involves students in class activities	78.57	21.43	0.00	0.00	0.00	0.00	4.79	
R10	Uses class time well	85.71	14.29	0.00	0.00	0.00	0.00	4.86	
R11	Fosters environment conducive to critical thinking	78.57	21.43	0.00	0.00	0.00	0.00	4.79	
R12	Treats all students in a consistent manner	92.86	7.14	0.00	0.00	0.00	0.00	4.93	
R13	Exams reflect the material covered	85.71	14.29	0.00	0.00	0.00	0.00	4.86	
R14	Willingly assists students outside of class	78.57	21.43	0.00	0.00	0.00	0.00	4.79	
R15	I found this class to be challenging	64.29	28.57	0.00	0.00	7.14	0.00	4.43	
Item ID	**Item**	**Excellent (5)**	**Very Good (4)**	**Good (3)**	**Fair (2)**	**Poor (1)**	**NR / NA**	**Mean**	
S01	Description of course objectives and assignments	78.57	21.43	0.00	0.00	0.00	0.00	4.79	
S02	Communication of ideas and information	85.71	0.00	14.29	0.00	0.00	0.00	4.71	
S03	Expression of expectations for this class	85.71	7.14	7.14	0.00	0.00	0.00	4.79	
S04	Availability to assist students in or out of class	78.57	14.29	7.14	0.00	0.00	0.00	4.71	
S05	Respect and concern for students	78.57	21.43	0.00	0.00	0.00	0.00	4.79	
S06	Stimulation of interest in course	71.43	28.57	0.00	0.00	0.00	0.00	4.71	
S07	Facilitation of learning	78.57	21.43	0.00	0.00	0.00	0.00	4.79	
S08	Overall rating of instructor	92.86	7.14	0.00	0.00	0.00	0.00	4.93	
Item ID	**Item**	**Male**	**Female**						
Z01	Gender	85.71	7.14						
Item ID	**Item**	**< 2.00**	**2.00-2.49**	**2.50-2.99**	**3.00-3.49**	**3.50-4.00**			
Z02	Current Cumulative GPA	0.00	0.00	7.14	57.14	28.57			
Item ID	**Item**	**Full-Time**	**Part-Time**	**Unemployed**	**Retired**				
Z03	Current Employment	28.57	57.14	7.14	0.00				
Item ID	**Item**	**Freshman**	**Sophomore**	**Junior**	**Senior**	**Postbac**	**Masters**	**Doctoral**	**Other**
Z04	Classification	0.00	0.00	0.00	92.86	0.00	0.00		0.00

Table 4: Evaluation of the microcontroller course adapted with the Freescale student learning kit for spring 2007

Item ID	Item	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	NR / NA	Mean	
R01	Communicates effectively with students	88.24	11.76	0.00	0.00	0.00	0.00	4.88	
R02	Enthusiasm for course material and teaching	82.35	17.65	0.00	0.00	0.00	0.00	4.82	
R03	Mastery of the course content	94.12	5.88	0.00	0.00	0.00	0.00	4.94	
R04	Relates course material to current examples	88.24	11.76	0.00	0.00	0.00	0.00	4.88	
R05	Clearly explains complex concepts and ideas	76.47	23.53	0.00	0.00	0.00	0.00	4.76	
R06	Lectures organized and provide framework for learning	88.24	5.88	5.88	0.00	0.00	0.00	4.82	
R07	Course syllabus accurately described the course	88.24	11.76	0.00	0.00	0.00	0.00	4.88	
R08	Course instructional materials used effectively	94.12	5.88	0.00	0.00	0.00	0.00	4.94	
R09	Involves students in class activities	70.59	11.76	11.76	0.00	0.00	5.88	4.63	
R10	Uses class time well	76.47	23.53	0.00	0.00	0.00	0.00	4.76	
R11	Fosters environment conducive to critical thinking	76.47	23.53	0.00	0.00	0.00	0.00	4.76	
R12	Treats all students in a consistent manner	82.35	11.76	0.00	0.00	0.00	5.88	4.88	
R13	Exams reflect the material covered	88.24	11.76	0.00	0.00	0.00	0.00	4.88	
R14	Willingly assists students outside of class	76.47	17.65	5.88	0.00	0.00	0.00	4.71	
R15	I found this class to be challenging	76.47	17.65	5.88	0.00	0.00	0.00	4.71	
Item ID	**Item**	**Excellent (5)**	**Very Good (4)**	**Good (3)**	**Fair (2)**	**Poor (1)**	**NR / NA**	**Mean**	
S01	Description of course objectives and assignments	47.06	29.41	5.88	0.00	0.00	17.65	4.5	
S02	Communication of ideas and information	76.47	17.65	0.00	0.00	0.00	5.88	4.81	
S03	Expression of expectations for this class	76.47	23.53	0.00	0.00	0.00	0.00	4.76	
S04	Availability to assist students in or out of class	76.47	17.65	5.88	0.00	0.00	0.00	4.71	
S05	Respect and concern for students	70.59	29.41	0.00	0.00	0.00	0.00	4.71	
S06	Stimulation of interest in course	82.35	11.76	5.88	0.00	0.00	0.00	4.76	
S07	Facilitation of learning	82.35	17.65	0.00	0.00	0.00	0.00	4.82	
S08	Overall rating of instructor	100.00	0.00	0.00	0.00	0.00	0.00	5	
Item ID	**Item**	**Male**	**Female**						
Z01	Gender	82.35	5.88						
Item ID	**Item**	**< 2.00**	**2.00-2.49**	**2.50-2.99**	**3.00-3.49**	**3.50-4.00**			
Z02	Current Cumulative GPA	0.00	0.00	11.76	64.71	23.53			
Item ID	**Item**	**Full-Time**	**Part-Time**	**Unemployed**	**Retired**				
Z03	Current Employment	29.41	47.06	17.65	5.88				
Item ID	**Item**	**Freshman**	**Sophomore**	**Junior**	**Senior**	**Postbac**	**Masters**	**Doctoral**	**Other**
Z04	Classification	0.00	0.00	41.18	47.06	0.00	0.00	0	5.88

Table 5: Evaluation of the microcontroller course adapted with the Freescale student learning kit for summer 2007

Item ID	Item	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	NR / NA	Mean	
R01	Communicates effectively with students	61.11	38.89	0.00	0.00	0.00	0.00	4.61	
R02	Enthusiasm for course material and teaching	72.22	27.78	0.00	0.00	0.00	0.00	4.72	
R03	Mastery of the course content	94.44	5.56	0.00	0.00	0.00	0.00	4.94	
R04	Relates course material to current examples	83.33	16.67	0.00	0.00	0.00	0.00	4.83	
R05	Clearly explains complex concepts and ideas	77.78	22.22	0.00	0.00	0.00	0.00	4.78	
R06	Lectures organized and provide framework for learning	77.78	22.22	0.00	0.00	0.00	0.00	4.78	
R07	Course syllabus accurately described the course	77.78	22.22	0.00	0.00	0.00	0.00	4.78	
R08	Course instructional materials used effectively	77.78	22.22	0.00	0.00	0.00	0.00	4.78	
R09	Involves students in class activities	61.11	33.33	5.56	0.00	0.00	0.00	4.56	
R10	Uses class time well	55.56	33.33	11.11	0.00	0.00	0.00	4.44	
R11	Fosters environment conducive to critical thinking	77.78	22.22	0.00	0.00	0.00	0.00	4.78	
R12	Treats all students in a consistent manner	77.78	16.67	5.56	0.00	0.00	0.00	4.72	
R13	Exams reflect the material covered	72.22	27.78	0.00	0.00	0.00	0.00	4.72	
R14	Willingly assists students outside of class	61.11	22.22	11.11	5.56	0.00	0.00	4.39	
R15	I found this class to be challenging	50.00	50.00	0.00	0.00	0.00	0.00	4.5	
Item ID	**Item**	**Excellent (5)**	**Very Good (4)**	**Good (3)**	**Fair (2)**	**Poor (1)**	**NR / NA**	**Mean**	
S01	Description of course objectives and assignments	66.67	33.33	0.00	0.00	0.00	0.00	4.67	
S02	Communication of ideas and information	66.67	33.33	0.00	0.00	0.00	0.00	4.67	
S03	Expression of expectations for this class	77.78	11.11	11.11	0.00	0.00	0.00	4.67	
S04	Availability to assist students in or out of class	55.56	27.78	11.11	5.56	0.00	0.00	4.33	
S05	Respect and concern for students	83.33	5.56	5.56	5.56	0.00	0.00	4.67	
S06	Stimulation of interest in course	77.78	16.67	5.56	0.00	0.00	0.00	4.72	
S07	Facilitation of learning	72.22	22.22	5.56	0.00	0.00	0.00	4.67	
S08	Overall rating of instructor	77.78	22.22	0.00	0.00	0.00	0.00	4.78	
Item ID	**Item**	**Male**	**Female**						
Z01	Gender	94.44	5.56						
Item ID	**Item**	**< 2.00**	**2.00-2.49**	**2.50-2.99**	**3.00-3.49**	**3.50-4.00**			
Z02	Current Cumulative GPA	0.00	0.00	16.67	55.56	22.22			
Item ID	**Item**	**Full-Time**	**Part-Time**	**Unemployed**	**Retired**				
Z03	Current Employment	27.78	44.44	22.22	0.00				
Item ID	**Item**	**Freshman**	**Sophomore**	**Junior**	**Senior**	**Postbac**	**Masters**	**Doctoral**	**Other**
Z04	Classification	0.00	0.00	16.67	77.78	0.00	0.00		0.00