Int. J. Engng Ed. Vol. 13, No. 4, p. 269–274, 1997
Printed in Great Britain.

0949-149X/91 $3.00+0.00

A New Approach for Teaching Microcontroller Courses*

T. K. HAMRITA
R. W. McCLENDON
Department of Biological and Agricultural Engineering, Driftmier Engineering Center, The University of Georgia, Athens, GA 30602-4435, USA. E-mail: takoi@bae.uga.edu

The microprocessor is playing an increasingly important role in a wide range of engineering applications. Engineers from all disciplines benefit from learning the power of the microprocessor in solving engineering problems. Within colleges or schools of engineering, microprocessor courses are currently taught by electrical engineering departments and in some cases by mechanical engineering departments. In this paper, we propose and discuss a new approach for teaching microprocessor courses. This approach allows all engineering students the opportunity to learn to use the microprocessor as a tool for solving engineering monitoring and control problems. The approach consists of shifting the focus of the course from the microprocessor itself to learning the design methodology by which the microprocessor could be used as a tool to solve practical engineering problems. We propose ideas to facilitate implementation of the approach and discuss its various benefits to engineering education.

INTRODUCTION

TRADITIONALLY, microprocessor courses have been taught within computer science departments or in electrical engineering departments. Focus within these courses has been primarily on computer software and hardware architecture. Recent advances in semiconductor electronics have changed the way industry solves manufacturing and process control problems. Many control problems which were previously solved using mechanical or electrical switching systems, can now be solved more effectively and reliably using electronic devices [1]. The heart of such an electronic monitoring and control system is a microprocessor. This increased use of microprocessors in solving monitoring and control problems has led to the appearance of microprocessors which are designed for the sole purpose of monitoring and control. These types of microprocessors are referred to as microcontrollers. The proliferation of low-cost microcontrollers and their development systems coupled with their increased use in solving industrial monitoring and control problems has led to new trends in microprocessor education.

Within electrical engineering departments, various educational objectives are beginning to be met by microprocessor courses. In particular, several schools have adopted microcontrollers in their courses instead of general-purpose microprocessors [2–4]. Consequently, a shift is being made from teaching the microprocessor architecture and the design of software to the use of microcontrollers in small monitoring and control exercises. An interesting example of this type of change is the one that took place at Pennsylvania State University where a microcontroller course was developed to integrate the electrical engineering curriculum [3]. Mechanical engineering departments have also started recognizing the important role of the microcontroller in solving monitoring and control problems in mechanical systems and started offering them for their students. An interesting example of this type of development is the one which took place at the School of Mechanical Engineering at the Georgia Institute of Technology [2].

These trends in teaching microprocessor courses in engineering education are placing electrical and mechanical engineers at a distinct advantage when it comes to using the microcontroller as a monitoring and control tool. We are aware of very few departments which are neither electrical engineering nor mechanical engineering which offer microcontroller courses in their curricula. As Wray *et al.* state in their book, 'Many non-electronic engineers are confused by the microprocessor. They often view it as a mysterious device that they will never understand [1]'. This lack of understanding regarding the use of microprocessors in monitoring and control among non-electrical engineers is inconsistent with ABET requirements in this regard which indicate that engineering students must demonstrate a knowledge of the use of computers in data acquisition and processing and process control. Moreover, at the industry level, applications of monitoring and control fall in all engineering areas and are not restricted to mechanical or electrical systems. For example,

* Accepted 14 July 1997.

chemical engineers must monitor and control many applications such as the conditions within a chemical reactor. Additionally, it is desirable for the engineer who designs and builds the control system to be knowledgeable about the application being monitored and controlled. Consequently, the chemical engineer who is knowledgeable about microcontrollers and their uses would likely be more qualified to control a chemical process than an electrical engineer.

As they are currently being taught, microcontroller courses are not very suitable for the non-electrical or non-mechanical engineer. Even with the recently documented new teaching directions [3, 5], students must still program in assembly language and spend a good portion of the term learning about programming and details of the microcontroller architecture. In this paper, we propose a new approach for teaching microcontroller courses. The primary objective of this approach is to present microcontrollers in a way that is appealing and applicable to all engineering disciplines. The approach is based on teaching the microcontroller as a design tool and emphasizing its use in solving practical and important engineering monitoring and control problems. In this way, the course is built around the application rather than the microcontroller. The focus is placed on the methodology of using the microcontroller to solve the problem rather than on details of the microcontroller. In the following, we first present the circumstances in our department which led to the development of this approach. Second, we discuss the proposed approach and present ideas to facilitate implementation. In the remainder of the paper, we discuss the various educational benefits of this approach.

MOTIVATION

The proposed approach for teaching microcontroller courses stemmed from our efforts in the Department of Biological and Agricultural Engineering at The University of Georgia to accommodate a unique engineering student body. Our department offers the only engineering degrees at the University. We offer a bachelor's degree in agricultural engineering which has evolved over the years from a traditional agricultural engineering degree to a 'general' engineering degree. Students must select an emphasis in one of five areas: electrical and electronic systems, mechanical systems, structural systems, processing, and natural resource management. In the past, we offered a course in microcontrollers with emphases on computer architecture and assembly programming. A simulator was also used to allow for the simulation of a few monitoring and control peripherals such as DIP switches and LED's. This course was required of students who selected an emphasis area in electrical and electronic systems but was optional for those in other areas. When it was time to mentor students in the senior engineering design courses, we noticed that several of our students, regardless of area of emphasis, were often interested in incorporating microcontrollers in their design. Those who were not in the electrical and electronic systems area of emphasis were at a disadvantage since they frequently did not take the microcontrollers course. As an effort to remedy this situation, we encouraged team design projects where the student who is interested in such an application would team up with someone who had taken the microcontroller course. This approach had limited success because the students within a team tended to work almost independently and only exchanged essential information.

Recently, we have restructured our curriculum for conversion from a quarter to a semester system. We decided to make the microcontroller course required of all of our agricultural engineering students regardless of their area of emphasis. Hence, we restructured the course to place less emphasis on the microcontroller (Motorola 68HC11) architecture and software and more on its use in sample monitoring and control applications. We also improved the laboratory by adding several development stations. Each station included an MC68HC11 evaluation board (EVB) and a target board which consists of devices such as LED's, DIP switches, stepper motors, key pads, and LCD displays. Student evaluations of the course indicated that although they were pleased with the course and the fact that it was required, they generally had a common complaint. The programming applications in assembly were tedious and it was difficult to program more than a few lines of code, therefore, the laboratory exercises were too simplified and lacked insight into the real world.

THE PROPOSED APPROACH

The proposed new approach for teaching microcontroller courses aims at teaching the microcontroller as a powerful design tool in solving various industrial monitoring and control problems. This is accomplished through teaching students a minimum base of information which is essential to using microcontrollers but more importantly teaching them a design methodology which they follow and build upon when they move on to industry. In this manner, the purpose of the course is to develop an essential framework for using the microcontroller as a design tool rather than learning every detail about the microcontroller itself and how to program it. The following is a discussion of the main building blocks to accomplish the new objectives of the microcontroller course.

The first and most important feature of the proposed method is to build the course around an actual and important engineering monitoring and control problem. This implies that at the beginning of the term, a problem is presented and its relevance to industry is illustrated with a few examples. This step is important in motivating the students and it also establishes practical and concrete goals to be reached. It may even be beneficial to demonstrate to the students a solution to the given problem to increase their interest in solving it. Focus is then placed on introducing the microcontroller in a way that highlights its different functions such as input/output, timing system, and interrupts. After students become familiar with what the microcontroller has to offer as a design tool, a solution strategy to the monitoring and control problem at hand should be developed with the different functions of the microcontroller as building blocks. The most relevant aspects of the solution are broken down into smaller tasks and assigned as laboratory exercises. The rest of the course is spent in developing the necessary skills to solve these tasks using the microcontroller while keeping the focus on the ultimate goal of solving the overall problem. An important point in this regard is that features of the microcontroller are not introduced before their relevance in solving the problem is demonstrated and no material is covered any deeper than is required to reach the solution to any given part of the problem.

An example which illustrates the proposed approach concerns teaching timer functions, interrupts, and interfacing with input/output peripherals. Instead of teaching these microcontroller functions independently, it is much more effective to teach them through a real-world engineering application. For example, monitoring and controlling pump flow rates is a common industrial application. A problem could be formulated to design a system which monitors a pump and maintains its flow rate at a user input value. A user interface to enter information specific to the pump and a display which shows the total volume pumped is needed. This is a problem which has practical value and its solution encompasses several important functions of the microcontroller.

The problem could be broken down into the following design steps:

1. how to sense and measure the flow rate of the pump;
2. how to control the pump flow rate;
3. how to design and build the user interface.

In addressing this problem, students first learn about the different sensing alternatives for detecting the speed of the rotating gear of the pump. In doing so, they learn that determining the speed of the pump is essentially equivalent to measuring the width of a pulse. In determining the speed of the pump, students learn that the timer input capture function is capable of measuring the duration of external events. Moreover, they learn that when the event is longer than a full 16-bit counter overflow period, they need to use timer overflow interrupts to keep track of the number of counter overflows. In controlling the speed of the motor, students learn about the concept of pulse width modulation (PWM) and how to achieve it using the timer output compare functions. They also learn about the concept of feedback when they use the measured speed of the pump together with PWM to control the pump flow rate. Finally, in designing the user interface, students learn about the input/output ports of the microcontroller and how to interface them with a key pad or an LCD display. Each component of the solution to this problem could be addressed in one or more labs as necessary.

Dealing with more complex applications requires more difficult assembly programming. With more assembly programming requirements there is less time in the term to focus on the application. Moreover, if computations involve floating point arithmetic such as trigonometric or logarithmic functions or even multiplication and division, coding becomes too difficult for the average student. Furthermore, if an assembly program subsequently must be modified so that a variable occupies more bytes than previously, this minor modification can affect hundreds of lines of code. Tracking down which lines of code need to be adjusted to compensate for this and modifying it could be a very time consuming task. In [7], it is suggested that 'C' is the ideal language for programming microcontrollers, and that 'C is a very powerful high-level language that allows the programmer access to the inner workings of the computer. Access to computer details, memory maps, register bits, and so forth, are not usually available with high level languages [7]'.

Making the switch from assembly programming to 'C' enables the students to write routines that would normally take hundreds of lines of assembly language instead of using just a few lines of 'C' code. This reduces the time required and the complexity of the coding, hence allowing for more complex and meaningful applications. Also, it is likely that 'C' will be the programming language they use to code applications in industry. However, it is important to emphasize that using 'C' in coding the assigned applications does not mean totally bypassing teaching assembly in microcontroller courses. On the contrary, we feel that teaching assembly is an essential part of learning about the microcontroller and is somewhat a prerequisite for using the 'C' language more effectively. As is stated in [7], 'If programming is only high-level, students will lose all sense of what is really happening'. It is through assembly programming that students learn the inner workings of the microcontroller. For example, through the assembly language, students develop an appreciation for bits and bytes, the different addressing modes, cycle-by-cycle execution of an instruction, complexity of floating point arithmetic, and memory allocation. Moreover, a student who learns assembly is better equipped to debug 'C' programs. Another reason why it is important to learn assembly in programming microcontrollers is that certain types of tasks are more easily and efficiently programmed in assembly. This is especially the case when memory is limited and code is to be optimized. Referring back to our pump example, students could program the timer functions using assembly, but the rest of the program, especially the computations used to determine the control actions, should be programmed using 'C'.

When significant course revisions are made, the faculty members initiating the changes must spend a considerable amount of time to make it happen. We realize that teaching a microcontroller course following the proposed approach is very demanding on the instructor during the initial stages. In particular the instructor must identify industry type applications which can be easily and economically duplicated in an academic laboratory environment, develop and test these applications, and present them in ways that are interesting to students and conducive to the learning environment. Much thought has to be given to selecting applications which involve pertinent functions of the microcontroller such as interrupts and timers, a good mix of hardware, and a broad range of required domain knowledge. The following is a discussion of resources and techniques which could be used to facilitate the transition to the new approach and to make the task of the instructor somewhat easier.

The instructor could rely on a number of sources for determining interesting applications for teaching microcontroller courses. For instance, if the instructor is involved in research projects which involve the use of microcontrollers in the design of monitoring and control systems, certain aspects of these projects may be easily duplicable in the teaching lab. In this case, even when the students do not duplicate the entire application in the lab, they have the opportunity to be exposed to the actual application through their instructor's research laboratory. Since we have several ongoing research projects in our department on applications of microprocessor-based control in biological and agricultural systems, this option is a viable one for us.

Another possible source of applications could result from involvement of the instructor in non-proprietary consulting jobs with industry. These types of assignments are usually less complex than those covered in research and do not usually last for long periods of time. Hence, they may be easier to duplicate in teaching labs.

A more viable source of applications could result from the involvement of senior students in design projects. Instructors could assign interesting applications which have industrial significance as design projects either to individuals or to groups of students. The ideal scenario would be to have companies sponsor these projects. Similarly, instructors could assign the same types of applications to motivated students as special topic courses. Subsequently, portions of the senior design projects could be used in the microcontroller courses.

Implementation of the proposed approach could be accomplished incrementally. Whatever the source of the application might be, the instructor should mentor motivated upper class students in building the components of the course. The instructor could gradually incorporate the important aspects of the application into the existing course until most of the labs have been developed. As a last step, the instructor could restructure the class lectures to follow the proposed approach and build the lectures around the application.

EDUCATIONAL BENEFITS

Our approach for teaching microcontroller courses opens up a wide range of opportunities for exposing students to applied engineering problems and to the unique solutions that microcontrollers offer in dealing with these problems. The following are educational benefits which would result from the proposed approach. Undergraduate students in engineering curricula are constantly searching for answers to the question 'what does an engineer do in the real world?' Often it is only in their senior year that they get a glimpse of the answer through a senior design experience. Prior to the senior year, most engineering courses are largely engineering science with some exposure to engineering design. Our approach would allow for an earlier exposure to engineering design related to monitoring and control. Some of the advantages of such an approach include:

- motivating the students for later classes;
- broadening their perspectives;
- enhancing their job opportunities.

The approach of building microcontroller courses around real-world applications provides the opportunity of tackling problems which would not be normally encountered in traditional microcontroller courses. Some of these problems stem from the fact that real-world applications come with all types of limitations, constraints, and lack of information. When teaching microcontroller applications, it is necessary to include sensor selection and associated characteristics as well as signal conditioning. For instance, going back to our pump example, students become aware of the basic types of proximity sensors (transducive, capacitive, photo-diode) which could be used to generate the pulse and of the limitations and advantages of each type. It is important to make the students aware of what is necessary to transform a signal so that it is compatible with the microcontroller. Another important issue is dealing with sensors which behave in an undesirable manner. Ideally, when we use a sensor, we would like for it to behave in a linear fashion. However, many sensors provide nonlinear signals thus complicating the problem. Students must learn how to deal with these nonlinearities.

Shifting the focus of microcontroller courses to real world applications provides an excellent opportunity for co-operative learning. In an industry-type engineering environment, engineers from various disciplines must interact and integrate their different skills to reach a solution to a given problem. When the applications in microcontroller courses progress beyond simple blinking of LED's or reading of DIP switches to a real-world engineering problem, the solution could become more than one individual could handle. This is the first of five criteria used by the ASEE's National Effective Teaching Institute in defining co-operative learning [8]. When students must rely on one another to reach the solution to a given problem, teamwork bmcomes positive. This is as opposed to a team of students getting together to solve a problem that each could solve individually. In the latter case, one talented student usually takes over the assignment while the others watch. Due to the diversity among the students in our department with different areas of emphasis, we have a unique opportunity for making the assigned projects overlap among various disciplines. While working on the assignments inside and outside the classroom, those students with a particular background become a resource for the rest of the students who do not have that background. With this type of approach, departments who teach microcontroller courses could open these courses for engineers from other disciplines. In addition to allowing those students from other departments to benefit from a microcontroller course, they can also implement co-operative learning among interdisciplinary teams, hence simulating the industry environment and broadening their students' vision.

As defined by ABET, 'engineering design is the process of designing a system, component, or process to meet desired needs' [9]. The design experience must allow for open-ended problems, creativity, problem statements, alternative solutions, and other considerations. Structuring microcontroller courses around a design problem allows the students to focus on the use of the microcontroller as a tool to accomplish the above objectives. ABET also requires appropriate computer-based experience in the engineering curriculum and specifically, 'data acquisition and processing, and process control' [9]. Traditionally, most engineering curricula have focused on the use of digital computer techniques to solve specific engineering problems. These problems have largely been involved with technical calculations and problem solving using a personal computer or a workstation. The areas of 'data acquisition and processing and process control' are often given less importance. It is these areas in which the microcontroller is especially suited yet most engineering disciplines neglect this methodology.

In [5], the author suggests that the way we currently teach microprocessors gives very low priority to interrupts. 'If interrupts are covered at all, it is only to describe them without saying what to do with them. The ideas behind one processor managing multiple tasks and juggling them around in ways which depend on the degree of urgency involved are quite unfamiliar to most students [5]'. Based on our own experience, we found out that was indeed the case. Previously, we tended to leave interrupts until the end of the quarter, and even when we covered them, there was insufficient time to give the students any meaningful experience. Since applied real-time events which must be monitored and controlled do not necessarily occur in sequence, it is very important to teach the students about interrupts and how to use them to manage the real-time flow of events. The proposed approach for teaching microcontroller courses gives us an opportunity to teach interrupts to a much greater extent. Since the focus of the course is on solving real world problems, inevitably interrupts become of high interest. Hence, learning about interrupts becomes part of the solution to the problem.

CONCLUSIONS

A new approach for teaching microcontroller courses has been developed in this paper. The approach presents the microcontroller as a viable tool for solving a wide range of industrial-type monitoring and control design problems in a way that is appealing and applicable to all engineering disciplines. The mechanism or vehicle which was proposed to accomplish this goal is to build the course around a real-world monitoring and control application. Designing the application would require knowledge of important functions of the microcontroller. The idea is to follow a step-by-step approach for solving the problem using the microcontroller and to cover the inner workings of the microcontroller only as needed to solve the problem. In this way, the focus originates at the application itself and goes all the way to the details of the microcontroller which are required to be able to use the microcontroller to solve the problem. Applications could be selected to match the students background; in this way microcontroller classes become more universal and less restricted to electrical engineering students.

The advantages of using such an approach include:

1. exposing students earlier in the curriculum to real-world engineering applications versus the simpler applications which are currently used in microcontroller classes;
2. enhancing co-operative learning which is essential in our curricula;
3. satisfying ABET design and computer knowledge requirements.

The proposed approach is not necessarily suited for every curriculum and may not be a substitute for microcontroller courses in electrical engineering curricula. Electrical engineers who are interested in designing microcontrollers would continue to focus on computer architecture and software. All engineering students, however, would benefit greatly from learning how the microcontroller could be used as a tool and how it fits in engineering design.

REFERENCES

1. Wray and B. Crawford, *What Every Engineer Should Know About Microcomputers: Systems Design and Debugging,* Marcel Dekker, Inc. (1984) p. iii.
2. Umeagukwu, J. McCormick, Microcomputers in the Mechanical Engineering Microprocessor Laboratory at Georgia Tech: Part I, *Int. J. Appl. Engrg. Ed.,* **6**, (1990) pp. 641–648.
3. Mayer, T. N. Jackson, and M. E. Lockley, A new role for microcontroller courses: integrating EE curricula, *Frontiers in Education,* 25th Annual Conference, November 1–4, 1995. Georgia Institute of Technology.
4. Hanson, A microprocessor laboratory for electrical engineering seniors, *IEEE Trans. Educ.* **E24,** (Feb. 1981) pp. 8–14.
5. Schultz, New directions in microcomputer education, *1991 Frontiers in Education Conference,* pp. 642–645.
6. Bennett, C. F. Evert Jr., and L. C. Lander, *What Every Engineer Should Know About Microcomputers: Hardware/Software Design,* Marcel Dekker, Inc., New York, (1991).
7. Sickle, *Programming Microcontrollers in C,* Hightext (1994).
8. Felder, J. E. Stice, National Effective Teaching Institute, Washington, DC June 20–22, 1996.
9. Engineering Accreditation Commission, Accreditation Board for Engineering and Technology, Inc., *Criteria For Accrediting Programs in Engineering In The United States,* 1996–1997, pp. 11–12.

Takoi Hamrita was born in Tunisia in 1965. She received a BS, MS, and Ph.D. in Electrical Engineering at the Georgia Institute of Technology. She has been an Assistant Professor of Biological and Agricultural Engineering at the University of Georgia since 1995. Her teaching interests are in the areas of microcontrollers and control theory. Her research interests are in applications of microprocessor-based control in biological and agricultural systems. Dr. Hamrita has recently been selected as a Lilly Teaching Fellow.

Ron McClendon received his BS and MS in Aerospace Engineering and Ph.D. in Engineering with a Composite Major at Mississippi State University. He is currently the Undergraduate Coordinator and Professor of Biological and Agricultural Engineering and Faulty Fellow of the Artificial Intelligence Center at the University of Georgia. Dr. McClendon's research interests are in developing decision-support systems incorporating artificial intelligence.