AC 2010-198: NEW APPROACH FOR TEACHING A MICROCONTROLLERS SYSTEM DESIGN COURSE FOR ENGINEERING TECHNOLOGY

Fernando Rios-Gutierrez, Georgia Southern University

New Approach for Teaching a Microcontrollers Systems Design Course for Engineering Technology

Abstract

The Electrical Engineering Technology (EET) program at our University (XXX) offers a Microcontrollers System Design course which is a four-credit course that all students are required to take for the EET program. The main goal of this course is for the students to learn basic programming techniques (in assembly language) and practice their implementation in practical projects for the lab component. However, although the department has offered the microcontrollers course for many years and students have been successful in learning assembly programming and basic project's testing and implementation, the course has been not easy for the students to understand and the lab projects were too simple that they did not afford for the students to truly demonstrate their full capabilities.

Due to the above, the microcontrollers course has been modified and the contents updated to have the latest technology available and its contents made more accesible for the students and to develop lab experiments in which students learn to implement actual engineering applications.

In the new course format the course outcomes have been upgraded to include:

- Learning basic structure programming techniques and the use of simple data structures.
- Learning the description of a program structure using flow diagrams.
- Updating the course contents so that besides learning to program in assembly language, students also learn how the same program can be implemented in a high level language such as C.
- Updating lab projects that show the use of microcontrollers in real engineering applications
- Developing of team projects
- Writing of formal project reports.

The new format has elevated the level of student achievement in the course and their projects implementation are now typically more challenging and professionally manufactured, so that they allow the interactions of the students as a team.

The paper describes the faculty effort required to make the appropriate changes in preparation for the new format of this course, the particulars of the implementation, how the course evolved and the improvement in student performances are described. We also present the methods, mechanisms, and lessons learned during the process and how they can be helpful to others contemplating a similar course, or those anticipating a revision to an existing engineering technology program.

Introduction

The EET program at XXX's College of Science and Technology (COST) requires all students to take a microcontrollers system design and assembly language programming course, to learn their application in the implementation of an embedded system. The purpose of this course requirement is both to enhance the technological knowledge in the electrical engineering field and to better prepare GSU students to be competitive in the work world. In particular, GSU students need to be prepared for a job market that requires programming skills in low level and high level languages programming. Most digital systems today are programmable and require knowledge in cutting edge technology, and microcontrollers have become an important part of most electronic devices and appliances in engineering, industry, business, and information technology.

The main motivation to update this course was to provide better programming experience and real engineering applications experiments to the students taking this course, and at the same time, making the material easier to understand. To make the learning experience for the students more interesting we implemented the following policies:

a) Increasing interactions between instructor and students in class and lab
b) Enhancing communication with students during class and lab sessions
c) Supporting students software learning and project development while using microcontrollers for the first time
d) Enhancing active learning and instruction
e) Practicing program debugging and documentation
f) Improving class assessment and evaluation of students
g) Motivating collaborative and interactive learning

With the adoption of the above policies in our classes we have made the learning experience more interesting and dynamic for the students, made difficult topics more accessible and easier to understand, since we provide students with immediate access to debugging and simulation tools and the use of online course material such as lecture slides and program examples. With the use of in-lab simulators and interactive debugging tools students feel more confident and are more willing to participate in implementation of the lab projects.

We take advantage of the lab's equipment, work stations capabilities and internet connection for students to have access to databases, design and programming tools, spreadsheet use and image capture display. Also, students are able to generate or acquire data files during labs, return to a desktop environment, and continue to work. Software demonstrations and interactive exercises can be downloaded and run in the lab and then carried home for future reference. As their programming skills progress, students are able to write software on their desktops that can be downloaded and run from the prototyping cards.

The lab environment made possible efficient and accurate interaction with course material and data. Students are able to use the work stations as a microcontroller productivity tool. They also are able to take advantage of the university's expanding wireless network, to access the Internet without having to go into a computer lab. All class-related documents (lecture notes, assignments, syllabi, and software packages) are available through the wireless network.

Lab Environment Characteristics

The laboratory for this class counts with 20 workstations that can be used by the students to write, test and implement their projects. The workstation setting is shown in Figure 1.

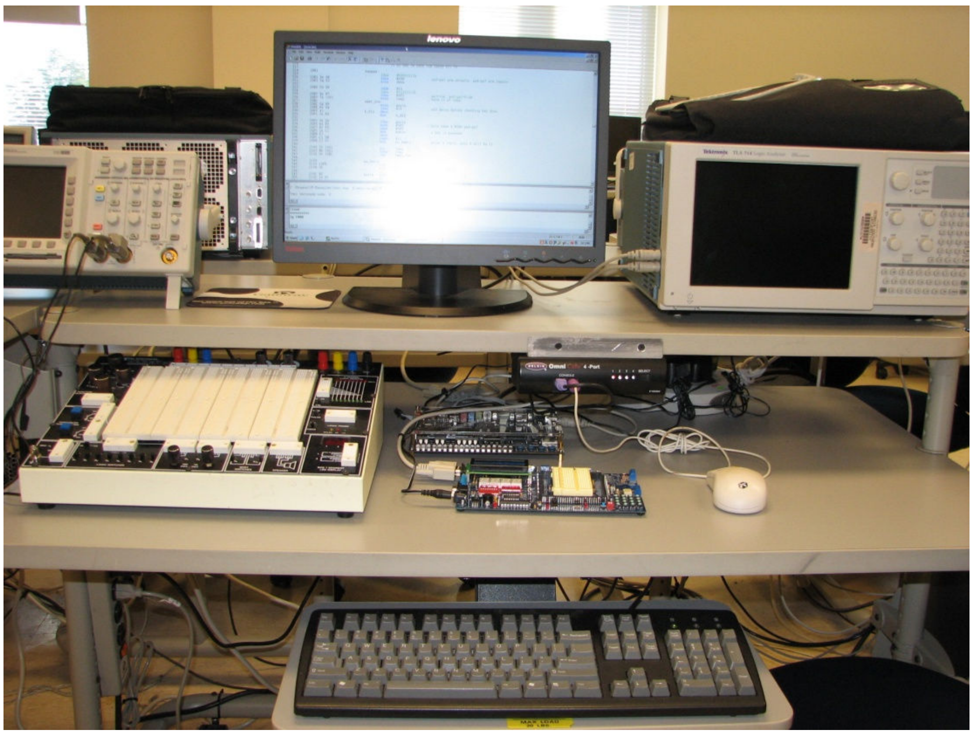

Figure1. Microcontrollers Workstation

As can be seen from this figure, each workstation counts with all the hardware and testing equipment that students need to implement their projects. The equipment available on each station includes a microcontroller prototyping card, a desktop computer, an oscilloscope, a logic analyzer, an FPGA card (that is not used in this course) and a general purpose prototyping board.

The main equipment that students use in the implementation of their projects is the prototyping microcontroller card (Dragon12 MC68HCS12 EVB board from Wytec[1]), as the one shown in Figure 2. To write, compile and download their programs students use the MiniIDE software from MGTEK[2] and for simulation they use Almi's 68HCS12 simulator[3], which are available on each computer. After their programs are free of errors, they download it to the prototyping card. The microcontroller card is cable connected to the host computer. Once the program is downloaded, it can be run, debugged and tested. If needed, the card can be disconnected from the main computer and will retain their functionality unless the power is removed. An EPROM is also available in case the program needs to be left permanent. This means that the board can be used as a stand-alone system if required.

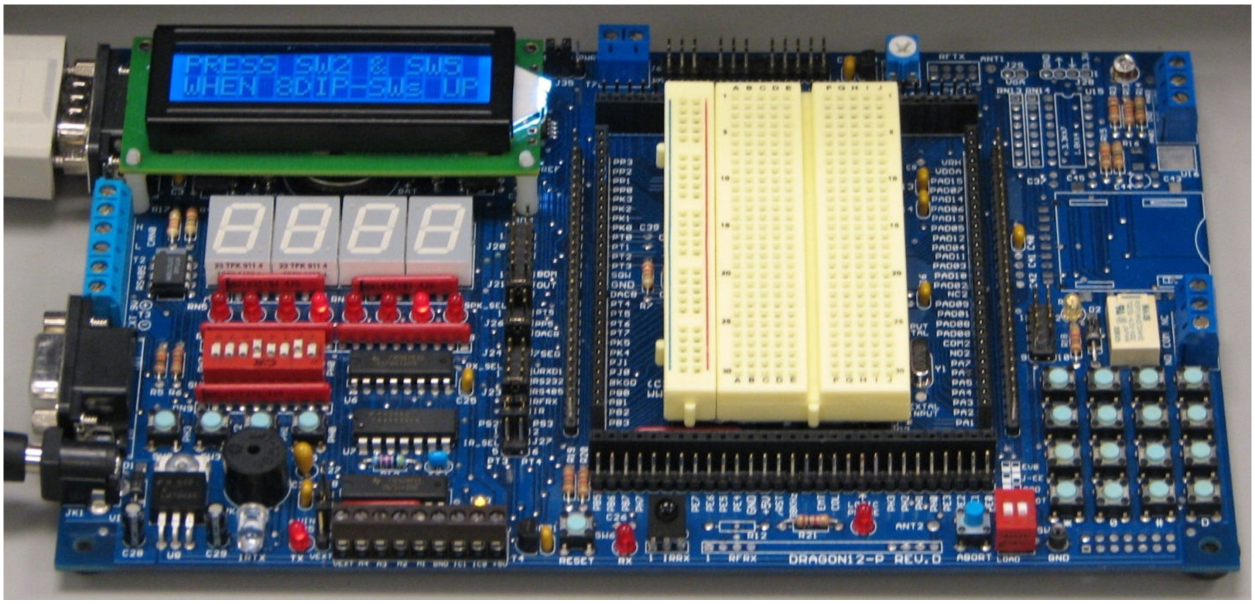

Figure 2. Dragon 12 Prototyping Card

Class Organization and Teams Interactions.

In order to take this course, students are required to have already taken at least a basic course in digital systems and another course on computer programming. The class meets three times a week for a 50 minutes lecture and has a two hour lab scheduled for the students to work on their projects. The class typically consists of a group of 40 students that are separated into independent design teams of two students each. Each team's goal is to design and test the corresponding project assigned for the week that has to be tested and implemented on the prototyping card.

All teams do the same project each week, so in order to reduce cheating, students need to prepare a pre-lab assignment that consists of:

a) Having all the theoretical work required for the implementation of the corresponding project,
b) show that they performed all the calculations required to implement the project, and
c) show a flow diagram of their programs.

All this material is reviewed at the beginning of each lab session. The two students on each team are equally responsible for the writing, testing and debugging of their projects. Students have freedom in selecting a different partner for each project and the only condition is that every student has to write an individual project report. The membership selection to a particular team is assigned by the instructor only when they are unwilling to find a partner on their own or in case of conflict. During the lab sessions the design teams receive continuous supervision from the faculty advisor and a TA, to which they need to show their projects working correctly.

Laboratory Projects

As mentioned above, students taking this class are given a different project each week and although all teams do the same project, they are encouraged to develop their own implementation ideas. To make the lab projects more attractive, all of the projects are based on actual automobile applications. We decided to do this because all students are familiar with cars and the information devices that are available in the cars. The list of the projects assigned include: a gas tank fuel meter, an intelligent tail-gate lights control, a car alarm system, a tachometer, a wireless car entry system, a speedometer, an odometer, an intelligent horn, an engine temperature meter and oil pressure meter.

To implement these projects, students need to use different input and output (I/O) devices already available in the prototyping card. These I/O devices include switches, Push Buttons, LEDs, seven-segment displays, LCD display, numeric keyboard, speaker, infrared emitter/receiver and analog to digital converter. Other external devices and equipment available for their projects include resistances, potentiometers, TTL gates, temperature and pressure sensors, power sources and signal generators. Each team works on its own design and implementation of each project. As part of their projects implementation, students are requested to write a comprehensive report that includes information about the description of the project, all design process followed, flow diagram, program listing and the results obtained in the lab. At the end of the semester, each team is also required to do an oral presentation in front of the class, for one of their projects implemented during the semester.

Projects Design and Implementation

With all the devices and capabilities available in the Dragon card, and the compiling and simulation software tools available, students are able to implement their application projects in the one week period of time assigned for each experiment. In order to achieve the completion of their projects, students take advantage of the useful set of I/O devices and system's subroutines that the microcontroller's operating system provides. These system's subroutines include programs that can be used to read a command from the keyboard or to send information to the different displays, etc. To use these subroutines students need to understand the use and programming characteristics for each of the I/O devices.

Class Assessment

The lab projects are evaluated in several stages, in a gradual and continuous way. First students are given a comprehensive lab experiment description that includes all the required technical and theoretical information that students need to develop their projects[4]. Based on the lab description students prepare the pre-lab assignment that is due before the starting of the lab session in which they show the design of their projects and the code of their programs which is graded according to their contents. Since they have only two hours in the actual lab to test and demonstrate that their projects are working, students need to come to the lab with their programs already simulated, to make sure their program will work as expected, using the simulator for the 68HS12 microcontroller. Next, during the lab sessions students need to demonstrate to the instructors their implementation of the project actually working, from which they receive a grade according to de completion of the project, and finally they need to submit a formal written report for each project that is due on the next lab session, so they have a week after the completion of the lab session to write the formal report.

Conclusions

Throughout the process of designing and building the projects, usually the teams make programming errors or make some mistakes on their implementation. The top challenge for students and faculty members is to be able to finish the projects in the two hours available in the lab. This situation is more critical at the beginning of the semester, but as they get more experience the number of mistakes decreases greatly. Some students suggested that it would be desirable to have more time to develop their projects. Based on this information, we are thinking on having, for the more complex projects, a two- week session time frame in order for the students to have more time for testing and debugging their projects. However, even with the time constraints most students are usually able to finish the project on time and with results as expected.

At the end of the semester we ask each team to make an oral presentation for one of their projects, and make a survey asking the students their opinion about the projects and the class in general. Based on survey responses from students, and our own in class and lab interactions with them, we found that students enjoyed the type the projects assigned and the overall response to the class assigned and the choice of microcontroller card is generally acceptable. With the use of the automobile application based projects we have detected an increment in class participation for part of the students and a faster dynamic in the class environment. The main complaints we get from students is about the time it takes to learn the assembly language, how to use all the tools available from the software and the large amount of information they have to read. Although the software package is available in all the computers located in this lab, the software can also be run with no problem in any laptop or desktop computer with the help of a free downloaded student version available from the microcontroller's manufacturer.

Since the cost of the microcontroller card we use is getting closer to the price of a textbook, in the near future a good option for this course would be to ask each student to purchase their own microcontroller card so that they could use it in other courses in the electrical engineering technology program. However, the main attraction of using the new microcontroller is the improvement in the learning experience and the optimization of the design process that makes it more appealing to the students.

As more technological advances are developed, the microcontroller cards will add more capabilities that will make their use more attractive. As the technology becomes more popular, the price of the cards will tend to decrease, so that there will be more attractive and universal tools to use in general, which will give our students an advantage in the long run.
Overall, this course provided the students with a way to experience the design and implementation of complex digital systems and develop practical applications.

Bibliography

1. Wytec Co, Dragon12 Trainer Boards, http://www.evbplus.com/hcs12.html
2. MGTEK Co, Downloads, HC11/12 Integrated Development Environment, http://www.mgtek.com/miniide/download/
3. Tom Almi, Designing with Microcontrollers, The 68HCS12 Simulator, http://hcs12text.com/hcs12textbook.html
4. Fernando Rios-Gutierrez, Course Material, TEET 2433 Microcontrollers, METEET Department GSU, http://personal.georgiasouthern.edu/~frios