# A Practical Approach to Education of Embedded Systems Engineering

**Özgür Yürür**
*Department of Electrical Engineering*
*University of South Florida*
*Tampa, Florida, 33620*
*oyurur@mail.usf.edu*

**Wilfrido Moreno**
*Department of Electrical Engineering*
*University of South Florida*
*Tampa, Florida, 33620*
*wmoreno@usf.edu*

***Abstract*** *— It is very common to see that in almost every daily routine, digital electronic devices provide vast comfort and flexibility for consumers. Companies designing and manufacturing these devices use advanced hardware and software tools. Engineering graduates sometimes have difficulty securing an entry-level industry job since there are major differences between what they have learned in school and the latest technological development tools used in the industry. In this paper, a state of the art embedded systems laboratory course is introduced. This lab course helps students with the experience of designing and implementing applications in embedded systems by using modern hardware and software development tools. It aims to teach theoretical and practical, organizational and architectural concepts regarding microprocessors via advanced engineering projects. The evaluation of the course indicates that most of the objectives have been achieved.*

***Keywords****:* Electrical Engineering Laboratory, Embedded Systems Design, Engineering Workforce Development Introduction Microprocessors Based Systems.

Digital electronics devices have been a critical part of human lives over the last two decades. It is easy to see that these devices are being heavily used in every environment where humans interact such as medical, security, and communications to name a few. Therefore, the design and enhancement of these devices are crucial to fulfill the future consumer requirements. In this regard, electronics device manufacturers have to maintain the quality of their engineer recruitments in such a way that not only responds to the consumer needs but also creates new developments and innovations. To be able to do that, the educational methodology of future engineers at universities plays a core role in preparing their students for jobs in the industry by introducing and teaching current technological advancements in the context of embedded systems hardware and software implementations.

The Department of Electrical Engineering at the University of South Florida (USF) has seen this necessity and has consequently developed a state of the art embedded systems laboratory course that enables its students to interact with the latest technological development tools, and provides creative environments to design their own projects. The lab aims to teach organizational and architectural concepts of microprocessors via introducing advanced engineering projects which help students to clarify the idea of how to integrate both hardware and software embedded systems skills.

For the required hardware and software development tools, the department collaborated with Freescale, one of the leading global semiconductor companies, to establish the laboratory course. The Tower System [1], which is one of the recent Freescale products, was chosen as the hardware development environment. CodeWarrior was also chosen as a computer aided design (CAD) tool for implementing software solutions. These tools are the integral parts of the laboratory setup. Equipped laboratory benches are very flexible, and allow students to conduct lab experiments and to create their own projects consisting of hardware and software modules

This work was supported by Freescale Semiconductor, Inc.

supported by the Tower System. These modules ranges from general purpose input and output (GPIO) units, analog to digital converters (ADCs), interrupt service mechanisms (ISRs), timers, serial communication interface units (SCIs), Ethernet, sensor managements (potentiometer, accelerometer) to some other peripheral units such as memory units, etc. which are common among most microprocessors.

In this paper, an overview of the philosophical approaches on how to manage and create advanced engineering projects is described in detail. Evaluation results show how this new lab course can extend students' engineering visions and support their senior-year projects.

## I. CONTENT OF THE LABORATORY COURSE

The lab course relies mostly on the introduction of hardware structures and the explanation of software implementations. Besides these clarifications, the general concepts of digital systems and microprocessors are also given at the beginning of the semester. Some important concepts of architectural organizations of microprocessors are also explained in detail to help students visualize how microprocessors are designed and how they operate.

Introduction to embedded systems programming is also presented to students in the beginning of the lab course. Since today's microprocessors are constructed with very powerful hardware units, the software programming of these devices has become more challenging. One of the important byproducts of the developed lab course is to let students learn C programming language with real time hardware applications. The development board has a real time operating system (RTOS) called MQX to run given applications. This sophisticated RTOS has numerous libraries to enable proper functionality operation. In this regard, the tutorial of the C programming is given in the light of introduction to MQX libraries and its functionality. Students are exposed in order to grasp software mentality and then expected to make reasonable connections between software and hardware organizations in such examples as how a hardware unit is represented as a structure in software manner.

Embedded systems programming started with the implementation of applications written in Assembly language, and then continued with the coexistence of Assembly and the C programming. Especially after the Internet processors came out, software implementation of embedded systems has become more modular and increasingly complex. This programming technique called advanced C programming is the primary method of software implementation in the lab course. All lab experiments are based on this technique with MQX functionality. However, the first experiment is presented to students in three different programming language formats, which are Assembly, Basic C and Advanced C respectively. The first experiment is designed to show how embedded systems programming has evolved in terms of design flexibility and modularity.

A threefold method is applied for the instruction of each lab experiment. The content of lab experiments are based on this method and each experiment step is documented in the lab manual [2]. First, a new hardware unit is introduced to students. The corresponding hardware structure, its register organization and pin mapping are explained in detail. Then, an application idea is put forward for the students' consideration. Over a selected hardware unit, the application is put into effect. During this process, the instructions that must be followed during the hardware implementation are explained, and then related pin assignments and other external connections are demonstrated Secondly, intellectual software organization is as depicted in MQX libraries representing the related hardware unit is shown. The essential variable declaration, functions and programming method are described. Lastly, the software implementation of the lab experiment on The Tower System is carried out in light of objectives stated in the lab manuals.

From the explanations of the issues discussed above, as illustrated in Fig.1, the ultimate objective of this lab course is to let students use hardware and software development tools effectively to create state-of-the-art engineering applications.

## II. Course Structure and Laboratory Experiments

The course is based on the successful completion of all the mandatory experiments and a design project. A few assignments as well as pop-quizzes throughout the semester are given. Moreover, a few optional advanced experiments are done if there is no time limitation. All mandatory experiments, assignments and a design project are carried out by lab teams, which are set to include no more than three students; however, writing a report for each application is an individual effort.

The lab course is offered to senior year students. C for Engineers and Logic Design are chosen as prerequisite courses.

For lab experiments, TWR-MCF51CN-KIT is used as the hardware development board. Lab experiments are designed to employ an on-board microprocessor, MCF51CN128, and external hardware extensions which are four LEDs, two push buttons, two DIP switches, a potentiometer, an accelerometer, a RS232 port, and an Ethernet port.

A brief description of mandatory experiments is given below:

*1) Comparison of Assembly, Basic C and Advanced C with General Purpose Input Output (GPIO)*

The objective of this experiment is to create an application which basically programs on-board LEDs to blink within specific time intervals, and to write related source code in Assembly, Basic C and Advanced C. This is a first experiment, to show how embedded systems programming techniques have evolved in time.

*2) Exploring Real Time Operating System (RTOS), Freescale MQX, and more GPIO utilization*

Starting from this experiment, MQX and its functionality will be further emphasized, and used for software implementations. The usage of the GPIO unit is explored in depth throughout this experiment, especially by focusing on its different data direction selections (Input/Output). The planned application for this experiment is that by using either onboard switches or buttons, external user interventions are perceived, and LEDs start to blink in different formats.

*3) More student involvement into driving GPIO*

This experiment aims to increase student excitement by letting them create their own user-defined functions to control MQX and GPIO unit. As a model application, a four bit binary counter is implemented by using LEDs and push buttons. The application includes using push buttons to increase or decrease the counter by one, and showing it over LEDs.

*4) Driving an Analog to Digital Converter (ADC) device, Potentiometer*

The experiment starts with a brief explanation of the principle of analog-to-digital conversion method and sampling theorem. Some important design parameters such as resolution, sampling rate, and quantization error are explained in detail. The desired application of this experiment is to read analog inputs from on-board potentiometer in a digital format, and then depending on its rotation level (magnitude of readings) is to set the number of alight LEDs.

*5) More ADC utilization, Accelerometer*

The objective of this experiment is the implementation of a motion detection algorithm on the development board. On-board accelerometer is used for this purpose. Therefore, besides similar instructions given for reading analog channels in the previous experiment, the hardware structure and working principle of an accelerometer is exposed to students.

*6) Driving a Serial Communication Interface (SCI) device, RS-232 port*

This experiment establishes a connection between the development board and PC over a serial communication interface. Characters typed through the keyboard are sent to the board, and bounced back to the PC. All transactions can be visible over HyperTerminal.

*7) Setting up Timers*

The objective is to set up two different timers to trigger two different user-defined functions. Each timer runs for a second, and then sleeps for a second. One of timers is forced to start running a second before the other starts. Therefore, two user-defined functions are triggered a second after another. User-defined functions are responsible for turning LEDs on or off. As a result, LEDs blink constantly every second.

*8) Understanding of Interrupts*

Multi-tasking is one of the important features that are brought by MQX. Until this experiment, software implementations have been based on polling methodology when needed. Triggering interrupts are not used for prompting different tasks. In this experiment, a user-defined interrupt service routine (ISR) is designed to take the place of a timer interrupt service module. The new ISR basically has a static counter value, and whenever called, it increments the counter by one. The objective of this experiment is after setting the new ISR, to wait for a while and then to check how many interrupts has occurred.

After the 3rd, 5th and 8th experiments, assignments are given to students. They have to create any application which consists of topics that been covered up to that time.

A brief description of optional (more advanced) experiments is given below:

*9) Trivial File System (TFS) and Memory Management*

This experiment intends to create a file directory in the flash memory, and to put some text files in this directory.

*10) Real Time TCP/IP Communication Suite (RTCS) and Telnet*

This experiment is prepared to show a piece of advanced application which shows how far embedded system technology can reach nowadays. A communication is established between a microprocessor and PC via the Internet. In able to do that, some important steps are explained in advance such as creating RTCS, setting local IP addresses, obtaining Ethernet MAC addresses, initializing the Ethernet device and binding IP addresses to the network over Telnet-ported sockets. On the other hand, shell console programming is used to recognize user-defined commands and execute them over Telnet connection. User-defined commands are chosen by turning on/off LEDs, reading a file in memory, and querying a counter for getting the number interrupt occurrences for a given time duration.

*11) Web Server, Dynamic Course Web Page Design*

The experiment includes creating a HTTP server and running a dynamic web page in the microprocessor. The web page shows the active status of LEDs, push buttons and potentiometer, and updates these statuses dynamically whenever any of them changes physically.

Pop-quizzes are usually given a week after a new hardware device has been successfully introduced.

For the final project, lab teams have to create a unique project idea which shows their interest in embedded systems.

TABLE I. LAB COURSE PLAN

| Weeks | Labs | Topics |
|---|---|---|
| #1 | Intro to Digital Systems | |
| #2 | Intro to C Programming | |
| #3 | Week #2 Cont. | |
| #4 | Intro to The Tower System | |
| #5 | Lab #1 | Assembly Language (GPIO) |
| #6 | Lab #1 Cont. | Basic C & Advanced C (GPIO) |
| #7 | Lab #2 | GPIO (Gen. Purpose Input/Output) |
| #8 | Lab #3 | GPIO |
| #9 | Student Exercise #1 | GPIO |
| #10 | Lab #4 | ADC (Analog to Digital Conv.) |
| #11 | Lab #5 | ADC |
| #12 | Student Exercise #2 | ADC |
| #13 | Lab #6 | SCI (Serial Comm.) |
| #14 | Lab #7 | Timers |
| #15 | Lab #8 | Interrupts |
| #16 | FINAL PROJECT DUE | |

## III. APPLIED METHODOLOGY

This lab course has been offered for the last three semesters. Due to time limitation during the semester and overall student assessment, the context of the lab course has been shortened to achieve proper and effective understanding of the course by students. Table I shows the course plan followed during the semester. As shown above, the lab course is student-oriented. The advanced labs are omitted, and they are reserved as an optional course of study for senior year project topics.

Covered topics include the most important sections in embedded systems. Students are exposed to the learning process of embedded systems from scratch for them to become capable of creating advanced projects.

The weeks where student exercises are held are reserved for creating an original project proposed by students instantly. The intention is to show how a project idea is turned into a real design and its implementation.

## IV. Evaluation of the Course

The evaluation of the Embedded Systems Laboratory course is made with the outcomes of questionnaires asked to students. The questions are prepared to collect students' opinions and experiences after taking the lab course. The evaluation results stand as a key point for the future improvement of the lab course to respond to students' needs and meet the desired objectives.

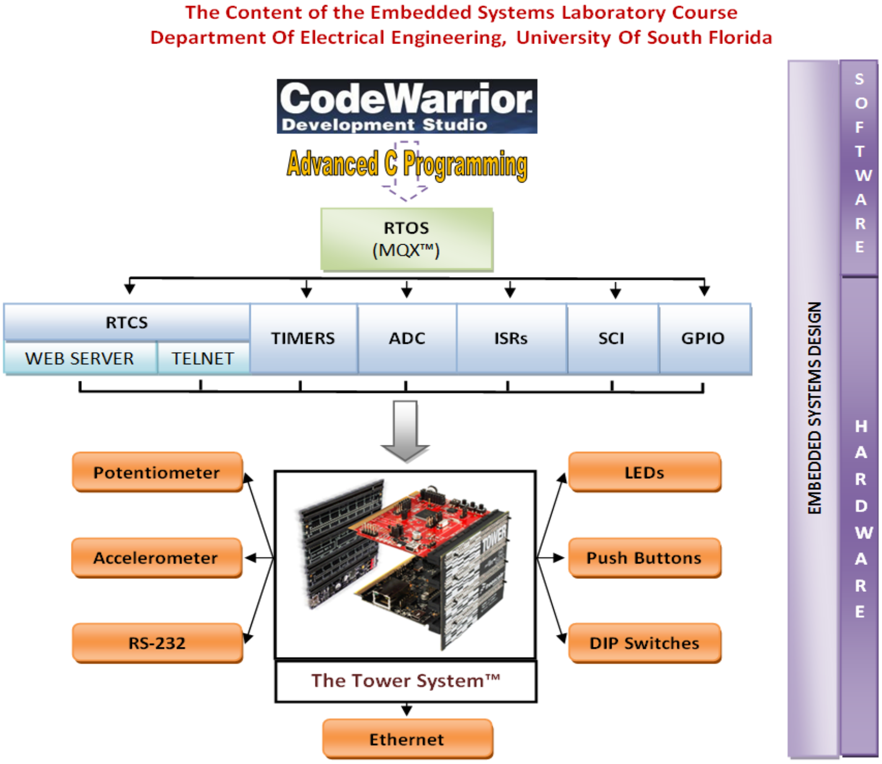


Figure 1. Content of the Lab Course

First of all, students feel excitement to create interesting projects. They find the lab course very interesting thanks to seeing some examples of state of the art applications in embedded systems design. They state that their interest into engineering and development of the technology has arisen after taking this lab course. According to their opinions, the way that they look at the consumer electronics devices, such as cell phones and mp3 players, have been evolving from the consumer perspective to the engineer perspective, especially thinking of how these devices are designed as a combination of both hardware and software.

Moreover, even if they have difficulty understanding and writing C programming code for applications, they agree with the idea that Advanced C supported with a rich set of MQX functionalities is a powerful tool for embedded systems applications. They confirm that learning a high level language is very helpful to design and improve the engineering project with the supporting of real implementations and simulations.

Another important issue which is pointed out by students is that the lab course helps them ~~to~~ understand organizational and architectural concepts of microprocessors better, and improves their knowledge of using hardware and software development tools in embedded systems design. Students think that this lab course is where theory meets practical engineering. The lab course covers enough theoretical information ranging from digital communication to logical design, from networks to circuit organizations concisely to be implemented practically in the platform of embedded systems.

Furthermore, students find the lab manual very useful since it walks them through the whole process step-by-step and helps them understand what is being done.

On the other hand, some students complain about the workload for writing reports after each lab or assignment, and find the time duration to submit them limited. Besides that, a few students criticize C programming requirements for the lab course. They believe it is beyond their programming skill obtained from the course entitled C for Engineers. However, before carrying out experiments, a well detailed tutorial of C programming is lectured and this process continues throughout the semester when it is needed. In addition, the lab course does not force them to learn the programming language strictly. Therefore, experiments, assignments, and projects are allowed to be done with team work. Also, grading policy does not only depend on the accuracy of software implementation. It actually relies on student effort and how seriously the lab course is taken.

The last thing that can be added to students' concern is that while taking this course they encounter a challenging face of the engineering world. Some of them think that they are not ready for this challenge.

## V. CONCLUSION

The embedded system laboratory course, developed in the department of electrical engineering at the USF, intends to provide an engineering system approach throughout the learning experience. One of the goals is to create a culture among the students to properly meet the challenges encountered in embedded systems hardware and software design; but most importantly, the lab serves as a vehicle to prepare new graduates for the engineering workforce, placing them in high demand due to their competencies in embedded design. Evaluation results show how this new lab course extends students' engineering vision, and respond to enhancements of the mentioned challenges. Other institutes that wish to establish an embedded system based laboratory course can benefit from the content and the methodology ~~of~~ that the lab course offers. The proposed lab course inspires students and keeps them motivated in the embedded systems world. As a last point, since it is complimentary to the traditional theoretical courses, it can serve as a resource for students to create design projects.


ACKNOWLEDGMENTS

The authors would like to thank Andy Mastronardi and John McLellan from Freescale Semiconductors, and Ed Harrold from Green Technology.


REFERENCES


[1] Freescale Techincal Manuals, http://www.freescale.com/tower

[2] O. Yurur, E. Harrold, and W. Moreno, "MCU and embedded systems course manual (featuring TWR-MCF51CN)," http://www.freescale.com/webapp/sps/site/overview.jsp?nodeId =06258 A2580257B55A0