

A Practical Approach to Embedded Systems Engineering Workforce Development

Özgür Yürür¹ [John McLellan², Andy Mastronardi³, Ed Harrold⁴, Wilfrido Moreno⁵]

Abstract – It is common to find digital electronic devices at the source of controlling many aspects of a modern consumer's daily routine; these devices regularly influence our comfort, and provide a means of flexibility. Companies designing and manufacturing these devices use advanced hardware and software tools. Engineering graduates sometimes find it challenging to secure an entry-level position since there are major differences between what they have been learning in school, and the latest development tools used in industry. This paper discusses a solution to this dilemma in the context of modern embedded systems. Students learn about real world design through a state of the art embedded systems laboratory course. The course, besides expecting to teach theoretical and practical organizational and architectural concepts of microprocessors, provides students with the experience of designing and implementing applications in embedded systems using modern hardware and software development tools. An evaluation of the course indicates that most of objectives have been achieved.

Keywords: Embedded Systems Design, Microprocessors Based Systems, Electrical Engineering Laboratory

INTRODUCTION

Digital electronic devices are today an undisputed global necessity. Embedded control systems are found at the heart of every major industry including environmental, medical, security, defense, transportation, and communications, to name a few. Therefore, in the context of electronics engineering, innovation and design of these devices is crucial to fulfill consumer expectations. To that end, electronics device manufacturers strive to maintain the highest quality of engineering recruitments, and must rely on universities to provide new engineers with relevant and practical knowledge of embedded systems hardware and software implementations.

The Department of Electrical Engineering at University of South Florida (USF) recognized this necessity and developed a state of the art embedded systems laboratory course that enables its students to interact with the latest technology and development tools, and provides the environment for students to create their own projects. The lab aims to teach organizational and architectural concepts of microprocessors via introducing advanced engineering projects requiring students to integrate both hardware and software as they learn embedded systems skills.

When choosing a platform for the lab, the department collaborated with Freescale, one of the leading global semiconductor companies. The lab is based on Freescale's Tower System, which is one of the recent Freescale products that provide a total solution for embedded design. The Tower System is integrated with the Codewarrior computer aided design (CAD) tool for implementing software solutions to various development boards. Laboratory bench based on the Tower System setups are very flexible and allow students to conduct lab experiments and create their own projects which include variety of hardware and software modules. The developed lab experiments include all the typical aspects of embedded design including general purpose input and output (GPIO), analog to digital

¹ The Dept. of Electrical Eng., Univ. of South FL, Tampa, FL, 33620, oyurur@mail.usf.edu

² Freescale Semiconductor Inc, Austin, TX, 78735, R2AAKL@freescale.com

³ Freescale Semiconductor Inc, Austin, TX, 78735, RAT119@freescale.com

⁴ Green Technology, LLC, Istachatta, FL, 34636, ed@greentechnologyllc.us

⁵ The Dept. of Electrical Eng., Univ. of South FL, Tampa, FL, 33620, wmoreno@usf.edu

converters (ADCs), interrupt service mechanisms (ISRs), timers, serial communication interface units (SCIs), Ethernet, sensor management (potentiometer, accelerometer) to name a few.

In this paper, an overview of the philosophical approach on how to manage and create advanced engineering projects for guiding students through the many complex areas of embedded design will be described. Evaluation results will show the impact on how this new lab can extend students' engineering vision and support their senior design projects.

CONTENT OF THE LABORATORY COURSE

The lab course begins with an introduction to digital systems. The summary of organizational and architectural structure of microprocessor is introduced in this section. Especially some important concepts such as system clock, central processing unit, memory, system operation registers, buses, instructions, pipelining etc. are brought to the notice of students to allow them to visualize how microprocessors based systems are built and how they operate.

One of the byproducts of the developed lab course, and since the lab software implementations are done in the C programming language, is a tutorial on the C programming language which is given to the students. The Tower System has a real time operating system (RTOS) called MQX to run given applications. MQX has numerous libraries to operate its functionality correctly [Freescale Manuals, 1, 2, 3]. These libraries heavily use structs and pointers, which are seen as the most challenging topics by students, to form up their variable declarations and functions. Therefore this kind of programming technique in embedded systems is called as Advanced C programming. In this regard, besides general topics of C programming, students are exposed to new bringing of Advanced C by making connections between software and hardware organizations in such examples like how a struct can represent a hardware unit, and how a pointer can help to reach from one hardware unit to another.

Implementation of applications in embedded system programming was started with using Assembly language, and it continued with using co-existence of Assembly and Basic C. After creation of internet processors, which has brought the concept of real time TCP/IP communication suite (RTCS) providing variety of IP networking protocols, the model of software implementations in embedded systems has turned into today's high technological form. Therefore, the first experiment of the lab course consists of an application that is written in three different language formats mentioned above respectively. The intent of the first experiment is to show how embedded systems programming has been evolving in the time in terms of flexibility in designing and modularity.

For the instruction of each lab experiment, a threefold method is followed. The content of experiments based on each method is documented as lab manuals. Firstly, a new hardware unit such as GPIO, ADC etc. is introduced to students in each lab experiment. Theoretical information, hardware structure including the map of pins and registers, and hardware functionality of the corresponding new hardware unit are explained in this section. For example, suppose that a SCI device is used in an experiment, and it is acted as if a development board was being created in this experiment to communicate microprocessor with PC over the SCI device. Circuit schematic of the required design application is showed to students from device data sheets to establish necessary hardware connections [Freescale Manuals 4, 5, 6]. The intent of this method desires to let students be aware of how they design their own projects.

Secondly, how intellectual organization of the desired hardware unit is represented in the software aspect provided by MQX is taught. By helping of existing OS libraries for the unit, the essential ingredients such as types, functions, attributes, variables etc. for the software recipe of each lab experiment are enlightened.

Lastly, software implementation on The Tower System aimed in a lab experiment is carried out in the light of stated objectives in the lab manual.

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

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 and a few pop-quizzes throughout the semester are also given. Moreover, a few optional advanced

experiments are done to extend student's engineering visions more. 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. Courses entitled C for Engineers and Logic Design are prerequisites courses.

For lab experiments, TWR-MCF51CN-KIT is used as hardware development board. Lab experiments are designed to use on-board microprocessor, MCF51CN128, and connectors of peripheral unit extensions which are four LEDs, two push buttons, two DIP switches, a potentiometer, an accelerometer, a RS-232 port, and an Ethernet port.

A brief description of mandatory experiments is given below:

1. *Comparison of Assembly, Basic C and Advanced C with using General Purpose Input Output (GPIO):* The objective of this experiment is to enlighten students by introducing how embedded systems programming techniques evolved in the time. Blinking on-board LEDs with a specific time intervals is intended in this experiment, and related software implementation is done by writing three different codes in Assembly, Basic C and Advanced C. GPIO hardware unit is firstly introduced to students by exploring its architectural mechanism and register structures. Then, ColdFire V1 instruction set, address and data accumulators, program counter and code condition register are exposed to students for accomplishing the given task in Assembly. For writing code in Basic C, at this time, software structure of GPIO unit defined in board supporting package libraries of the microprocessor is explained to students. By accessing and adjusting related registers with simple value assigning commands, software implementation is done. Lastly, for embedded programming in Advanced C, operations in MQX is started to describe since remaining all lab experiments will be based on using MQX functionality. As a result, with all three different code implementations, a desired experiment objective is successfully accomplished.
2. *Exploring Real Time Operating System (RTOS), Freescale MQX, and more GPIO utilization:* Starting from this experiment, MQX will be explored more in detail by understanding task creation methods, pin struct assignments, IO device driving methods [Freescale Manuals, 7]. First experiment includes GPIO unit for usage of output purposes (blinking LEDs); however, in this experiment, GPIO unit is used for both input and output purposes. The objective of this experiment is to perceive user interventions by using either on-board switches or buttons, and depending on incoming commands, to blink LEDs in different pattern.
3. *More student involvement to drive GPIO:* This experiment aims to increase student excitement by letting them to create their own user-defined functions controlling MQX and GPIO unit. A model application is proposed which includes demonstration of a four bit binary counter over LEDs and utilizing push buttons to increment or decrement the counter by one.

After this experiment, first assignment is given to students. As objective of the assignment, it is expected from students to form their own application.
4. *Driving an Analog to Digital Converter (ADC) device, Potentiometer:* This experiment firstly starts with a brief explanation of principle of analog to digital conversion method and sampling theorem by putting importance on some design parameters such as resolution, sampling rate, quantization error etc. Secondly, hardware structure of ADC units is introduced to students especially by informing descriptions of some important pins like reference voltages and analog channels. Then, related section in MQX for driving ADC device is studied to implement the experiment. The objective of this experiment is to read analog inputs from on-board potentiometer in digital formats, to figure out how much the potentiometer is open, and then depending on the value (magnitude of readings) to set the number of alight LEDs.
5. *More ADC utilization, Accelerometer:* The objective of this experiment is creation of a motion detector. Besides same instructions given for reading analog channels in previous experiment, hardware structure and working principle of an accelerometer will be introduced. The experiment is based on obtaining acceleration information of three coordinate spaces from on-board accelerometer, and using them for detection algorithm. Returning digital values from analog readings are compared to an experimentally found threshold value, which states the existence of motion. A sufficient acceleration variation over any coordinate space triggers LEDs blinking.

After this experiment, it is expected from students to form their own application as second assignment including ADC and GPIO units.

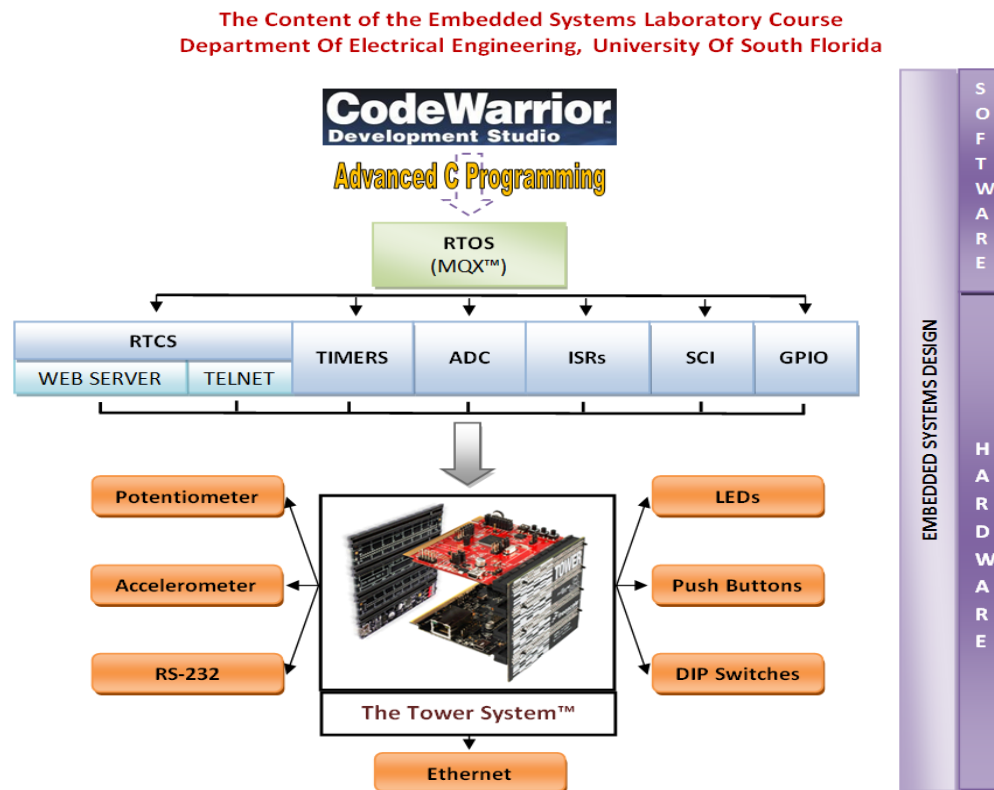


Figure 1: The content of the lab course

6. *Driving a Serial Communication Interface (SCI) device, RS-232 port:* This experiment studies establishment of serial communication between microprocessor and PC. For this purpose, SCI device is introduced to students in terms of its hardware structure embedded in microprocessor and its software structure involved in MQX OS. Some important key terms such as synchronous/asynchronous transmission, half/full duplex transmission, baud rate, parity check etc. are also carefully given to students. Such communication protocols like Universal Asynchronous Receiver/Transmitter (UART), Serial Peripheral Interface Bus (SPI) and Inter-Integrated Circuit Bus (I²C) are also briefly explained in comparison to SCI. The objective of experiment is to transmit keyboard characters from PC to microprocessor; on the other hand, microprocessor basically bounces characters back to PC. All transactions can be visible on HyperTerminal.
7. *Setting up Timers:* The objective of this experiment 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 turning LEDs on or off. As a result, LEDs blink constantly every a second. For the experiment, hardware instructions of timers are given by explaining their three different utilization methods which are counter, compare and capture. In software perspective, the method of counter using one-shot and periodic timers in MQX are introduced and implemented.
8. *Understanding of Interrupts:* One of important features brought by MQX is multi-tasking. Up to this experiment, thanks to the multi-tasking feature, polling methodology, if necessary, has been used instead of

triggering interrupts. In this experiment, a user-defined interrupt service routine (ISR) is designed to take 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 that after setting the new ISR, to wait for a while and then to check how many interrupts has occurred. Moreover, in the beginning of the experiment, some important issues such as working mechanism of interrupts, interrupt vectors, and what happens to normal program flow, system registers and stack memory while interrupts are being served etc. are explained to students.

For third assignment, it is expected from students to build up their projects which include one of topics stated in experiment 6, 7, or 8.

A brief description of optional advanced experiments is mentioned below:

9. *Trivial File System (TFS) and Memory Management*: This experiment intends to form a file directory in the flash memory, and to create some text files in this directory. Content of the files can be seen over PC screen by driving a SCI device. Before software implementation, related topics in MQX are introduced to students.
10. *Real Time TCP/IP Communication Suite (RTCS) and Telnet*: This experiment is prepared to show a piece of advanced applications that today's embedded system technology can reach. A communication is established between microprocessor and PC via Internet. Some important steps are explained to establish the network such as creating RTCS, setting local IP address, obtaining Ethernet MAC address, initializing the Ethernet device and binding IP address to the network over Telnet-ported sockets. Shell console programming is used to recognize user-defined commands and execute them over Telnet connection is applied. The user-defined commands are turning on/off LEDs, reading a file in memory, and querying counter for obtaining 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. Web page shows active status of LEDs, push buttons and potentiometer, and updates these statuses dynamically whenever any of them changes. For this experiment, usage of RTCS in MQX is studied further [Freescale Manuals, 8].

Pop-quizzes generally are given a week after whenever a new hardware device is successfully introduced.

For the final project idea, it is expected from lab teams to bring forward their own application, and implement it.

EVALUATION OF THE COURSE

The evaluation of the Embedded Systems Laboratory course includes responses of questionnaire asked to students for stating their opinions and experiences. The evaluation results are very important to the future improvement of the course which will help to respond totally students' needs.

First of all, students feel very excited to see examples of state of the art applications in embedded systems design; therefore, they find the lab course very interesting. According to their opinions, while taking the lab course, their visions and the way of looking on today's cell phones, mp3 players and the other popular electronics devices start to change from consumer perspective to designer perspective. They enjoy seeing how embedded systems work in real life by stating that their interest has arisen in engineering and technology after taking this lab course.

Moreover, even if they have difficult times to understand and write C code for applications, they agree with the idea of Advanced C supported with a rich set of MQX functionalities is a powerful tool for embedded systems applications especially when they see a really complex instruction set and implementation method of assembly language comparing to Advanced C. Students also propose the extension of the course entitled as C for engineers, and want it to be attached to the learning of The Tower System since they really like seeing on real time hardware changes such as blinking LEDs etc. whenever they set necessary adjustments by understandable software commands.

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. Thus, they state that they might think of the tools to design their senior year projects.

Besides its contribution to student improvement, students think that this lab course is where the theory meets real applications. Especially by looking at the different topics that experiments cover, the diversity of theoretical information ranging from digital communication to logical design, and from networks to circuit organizations are introduced concisely to be used in the platform of embedded systems.

Furthermore, most students think that course manuals are complementary to explanation of the subjects covered in the lab course and experiments. They find lab manuals very useful since it walks through the whole process step-by-step to help their understanding what is being done.

With respect to the negative feedback from the students, a few students complain about the workload due to writing individual reports for mandatory reports, and the week long time given to submit them. In addition to that, most of students criticize the lab needing to know C programming way beyond than their programming skill learnt on their C for Engineers sophomore course. However, in the beginning of the semester, a well detailed tutorial of C programming is lectured, and if necessary, some hints are also given throughout semester. Since the lab course does not force students to learn the programming language strictly, experiments, assignments and projects are allowed to be done in teams composed of up to three students. Therefore, grading of any course work does not only depend on software implementation. Last thing can be added to students' concern is that while taking this course; they see really challenging face of the engineering world. Thus, sometimes it makes them think that they are not ready for this challenge.

CONCLUSION

The embedded system laboratory course, developed in the department of Electrical Engineering at USF in collaboration with Freescale and Green Technology, intends to provide a systems approach throughout the learning experience. One goal 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 and placing them in high demand due to their competencies in embedded design. Evaluation results show the impact on how this new lab course extends students' engineering vision, and respond to enhancements of mentioned challenges. Other institutes that wish to establish an embedded system based laboratory course can benefit from the experiences discussed in this paper. The content and the methodology of the lab course that inspires students and keeps them motivated in the embedded systems world can be useful for other institutions. Moreover, since this lab covers concepts of some traditional senior core courses such as microprocessors, communications and controls, it can serve as a resource for students to develop senior design projects that encompass applications from any of these areas.

REFERENCES

- [1] Freescale Manuals, *Freescale MQX™ Real-Time Operating System*,
http://cache.freescale.com/files/32bit/doc/user_guide/MQXUG.pdf
- [2] Freescale Manuals, *Freescale MQX™ RTOS Reference Manual*
http://cache.freescale.com/files/32bit/doc/ref_manual/MQXRM.pdf
- [3] Freescale Manuals, *Using MQX Libraries*
http://cache.freescale.com/files/32bit/doc/app_note/AN3907.pdf
- [4] Freescale Manuals, *ColdFire® Family Programmer's Reference Manual*
http://www.freescale.com/files/dsp/doc/ref_manual/CFPRM.pdf
- [5] Freescale Manuals, *MCF51CN128 ColdFire® Microcontroller*
http://cache.freescale.com/files/32bit/doc/data_sheet/MCF51CN128.pdf
- [6] Freescale Manuals, *MCF51CN128 ColdFire® Integrated Microcontroller Reference Manual*
http://cache.freescale.com/files/32bit/doc/ref_manual/MCF51CN128RM.pdf
- [7] Freescale Manuals, *Freescale MQX™ I/O Drivers*
http://www.freescale.com/files/32bit/doc/user_guide/MQXIUG.pdf
- [8] Freescale Manuals, *Freescale MQX™ RTCS™ User's Guide*
http://cache.freescale.com/files/32bit/doc/user_guide/MQXRTCSUG.pdf

Özgür Yürür, M.S.E.E.

Özgür Yürür was born in Istanbul, Turkey in 1985. He received dual major degrees from the department of electronics engineering and the department of computer engineering at Gebze Institute of Technology, Kocaeli, Turkey in 2008, and M.Sc. degrees in electrical engineering from the University of South Florida (USF), Tampa, FL, USA in 2010. He is currently pursuing the Ph.D. degree in Electrical Engineering at USF.

Andy Mastronardi

Andy Mastronardi joined Freescale Semiconductor in September of 1999 and is Global Director of the Freescale University Programs. Prior to Freescale, Andy spent 26 years in the education industry, both as a teacher and in educational publishing. Andy completed his undergraduate studies at SUNY Potsdam and graduate studies at Fordham University and Long Island University. His background is in Mathematics and Mathematics Education.

Edward J. Harrold Jr., M.B.A., B.S.E.E.

Ed Harrold has worked in high-tech defense and aerospace industry since he served in the U.S. Army Security Agency. Besides his military training in electronics and crypto code interception, Ed has worked in engineering and management for companies like SmithKline, United Technologies, and Honeywell, and has earned numerous technical certificates and awards in electronics, software, robotics, and automation. Ed also earned his B.Sc. degree in Electrical Engineering from USF, and a MBA from Regis University where he currently teaches in the Master of Science in Computer Information Technology Program.

Wilfrido Moreno, Ph.D, P.E.

Dr. Wilfrido A. Moreno received his B.S.E.E., M.S.E.E. and Ph.D. degrees from the University of South Florida in Tampa, Florida. Dr. Moreno is a Professor in the Department of Electrical Engineering at the University of South Florida and is the director for the Ibero- American Science and Technology Educational Consortium at USF, (ISTEC-USF). Dr. Moreno's current areas of interest are oriented towards system integration by providing "off-the-shelf" hardware/software solutions to industrial applications. Dr. Moreno's broad educational background in communication, controls and microelectronics has allowed him to work on multidisciplinary research projects where concepts and expertise in all of these areas comes together for system design, fabrication and test.