

## Embedded Systems Education for Multiple Disciplines

Ali H. El-Mousa and Ashraf Al-Suyyagh

Department of Computer Engineering, University of Jordan, Amman 11942, Jordan

---

**Abstract: Problem statement:** This study described the process and experience of designing and developing an embedded systems first course that serves different disciplines at the University of Jordan together with its associated laboratory. The course serves three different departments with varying requirements and varying student initial background. The stages of the design process of the course and associated labs are explored. Emphasis is given throughout to the notion of systems design without losing track of the details involved in the design process. The design methodology used for the laboratory facilities and the learning outcomes are explored. **Approach:** The methodology followed in the design process was based on defining clear objectives taking into consideration the different backgrounds of students and departmental requirements. This was followed by defining the course content, designing the lab facilities and choosing the right hardware and software tools. We followed the Demonstrate, Work-to-Know and Design methodology for lab design **Results:** The course had been running successfully for several years now. Successful innovative projects undertaken by students and their results are reported. Due to its success, numerous senior capstone projects in many departments are now based on embedded systems ideas and notions. **Conclusion:** Embedded system teaching for multiple disciplines requires diligence and clear understanding of the varying requirements and backgrounds of students. Careful consideration should be given to the objectives, choice of tools and the timeline of the course and associated labs. Projects that tie the concepts together are very important for the success of the course.

**Key words:** Embedded systems, educational systems, microcontrollers, course design, lab facilities

---

### INTRODUCTION

An embedded system is defined as a “system whose principal function is not computational, but which is controlled by a computer embedded within it” (Wilmshurst, 2001). These types of systems have become very widely spread in all types of equipment whether consumer, industrial or military products. They can be found in washing machines, microwave ovens, cellular phones. They are the main subsystem in Programmable Logic Controllers (PLC) used to control industrial assembly lines and machinery. Their use has become prevalent in all aspects of our daily lives. The suggested guidelines for curricula for a computer engineering program produced by the joint task force of the IEEE Computer Society and the ACM recommends a minimum of 20 hours of core lectures in embedded systems. It also suggests that courses also cover elective topics in embedded systems (IEEE Computer Society/ACM Task Force on Computing Curriculum, 2004). Also, in recognition of the importance of the topic in education, for the first time a workshop on Embedded Systems education was organized by the ACM at the end of 2005 (Jackson and Caspi, 2005).

Other disciplines such as Electrical, Software and Mechatronic Engineering in many universities have also discovered the importance of the topic and have incorporated it into their curriculum.

Towards this end, the Computer Engineering Department of the Faculty of Engineering and Technology at the University of Jordan has developed a course with a design laboratory on embedded systems.

The offered course and its accompanying laboratory are designed as introductory suitable for all engineering students from the computer, electrical and mechatronic departments.

The course and associated laboratory are designed such that they cover concepts and topics on the use of embedded systems in different domains including control, instrumentation and embedded processing. This will appeal to the different backgrounds of the students. The laboratory experiments are quite intensive and are uniquely designed to reduce the learning curve and provide an enjoyable experience to the students.

The objectives of introducing the course and its accompanying laboratory in the curriculum of computer, electrical and mechatronic engineering are:

- Introducing the students to the growing importance of the field of embedded systems; its techniques, applications, concepts and practical implementation
- Helping the students grasp the concepts involved in engineering system design. This is particularly visible during the project phase at the end of the lab. This particular objective is unique as it is rarely possible in an undergraduate setting for the students to be able to start and finish a complete working project demonstrating a working prototype that involves both hardware and software, all in a single course and in a single semester
- Aiding the students in understanding the hardware-software co-design paradigm and the necessity of insuring compliance
- Offering practical hands-on experience in building embedded systems based on programmable devices such as microcontrollers. Also, providing experience in the use of digital and analogue devices for interfacing along with the required software control
- Helping the students understand the intricacies involved in applied embedded systems especially with topics that involve real time operation and control
- Introducing the students practically to the current tools and methodologies used in embedded system design and development

We outline the current introduction to embedded systems course and its accompanying laboratory offered by the Computer Engineering Department in the Faculty of Engineering and Technology at the University of Jordan.

## **MATERIALS AND METHODS**

**Course structure:** Students taking this course come from different disciplines i.e., Computer, electrical and mechatronic engineering. For all of them, the course is mandatory. However, their backgrounds differed which had to be taken into consideration. Students were either second or third year as the course is offered early on. By the time students take this course, all are expected to have strong background in computer applications, digital design fundamentals, electrical circuits, computer programming languages and electronic devices and their use. However, students from electrical and mechatronic engineering lacked previous exposure to assembly language, while some students from computer engineering lacked exposure to electronics design basics.

These differences had an impact on the structure of the course since it had to take them into consideration. The content had to be based on the common denominator and where it was necessary, to provide introductory background information. Each department will supplement the current course with other advanced material using other course offerings.

The approach taken in designing the course and accompanying lab took into consideration the following factors:

- The current prevalent academic environments where resources are a precious commodity, faculty are overburdened and the number of credit hours required for graduations are shrinking
- Although the course is introductory, the focus was on depth rather than breadth. This included in-depth look at interrupt structures and associated software, real-time issues and requirements, data acquisition systems and their components, advance use of debugging tools and utilities in the lab
- Reinforcing the different topics learned in the core curriculum by allowing the students to apply them and associate them in the course and lab. These include concepts relating to electronic circuits used in interfacing, communications hardware, digital logic design, signal integrity
- The realization that embedded systems designers in the market now utilize advanced platforms with complex capabilities in networking, user interfaces, processing power. This puts a burden on the software component side to be able to handle such tools and systems
- The fact that students are coming from different disciplines with slightly varying backgrounds in the prerequisite materials

The learning objectives of this basic introductory embedded systems design course and lab are summarized in the following areas:

1. Hardware understanding, limitations and implementation including Human Machine Interfaces (HMI), real world control techniques, I/O and data acquisition methodologies and timing
2. Software development for embedded systems taking into account storage limitations, startup sequence, optimized code, and the usage of current tools and debugging techniques.
3. The system design process and methodologies of system implementation testing and commissioning.

Table 1, presents a summary of the subject areas that the course covers.

Table 1: The areas and topics which the course addresses

General area and field	Topic(s)
Embedded architecture, system level	General CPU architectures, buses, general interfacing, the embedded system types (hard, soft and firm)
Embedded architecture, internal block diagrams	Embedded machine organization, built-in peripherals, RTL, HW control abstraction layer, I/O
Embedded communications architectures	Communications protocols, different serial interfaces, microcontroller communications ports, real world limitations, types of signal interference
Real-time embedded systems	Task scheduling, interrupt structures and latency, software requirements, critical code, multi-tasking
Data acquisition and embedded systems	System structure and hardware requirements, A/D and D/A, transducer interfacing, signal conditioning
Software design requirements	Hardware-software interface, co-design, embedded software architectures, assembly language programming
Embedded control and real world interfacing	External hardware interfacing techniques, different HMI, motor control, PWM techniques

Table 2: The course and lab outline

Topic	Lab work
CPU architectures and systems, basic computer organization	Intro to the lab and review of basic analogue and digital logic hardware concepts and constraints
Basic embedded systems architecture with emphasis on microcontrollers	Detailed introduction to the software working environment including step by step tutorial of debugging tools and methodologies
The minimum embedded system including CPU, I/O peripherals, timers/counters and basic interrupts	Counters, timers, simple I/O interfacing such as push button keys and LEDs and the use of simple interrupts
Embedded software development techniques and requirements	Covered in all experiments
<b>Microcontroller assembly language</b>	
Data Acquisition systems, structure, transducers, signal conditioning, analog multiplexing, sample and hold, A/D, D/A and PWM generation	Temperature sensor interfacing and periodic monitoring and logging. Generating PWM using both software and hardware
I/O real world interfacing techniques for digital and analogue signals including DC and stepper motor control. HMI including keypads, LCDs, multiplexed 7-segment displays	Covered in few experiments
<b>Position feedback using shaft encoders</b>	
Serial communications principles and constraints, simple RS232 protocol, multi drop RS485. Basic addressable USART	Data transfer between PC and microcontroller kit using RS232
Advanced Interrupt structures and interrupt latency	During data acquisition and control experiments
<b>Hard and soft real-time systems</b>	
System design basics from concept and requirements to commissioning, including multi-tasking	Term project from idea to concept to implementation

Based on the above objectives, factors, and constraints, the course outline was developed as shown in Table 2 along with the required lab exercises.

One of the difficult tasks faced when deciding on the content of the course was the choice of text book.

Unfortunately, no single book could be found that adequately covers the required material. For example, one of the widely used books is Vahid and Givargis (2002), however, the content is quite general and does not address practical matters nor particular platforms. Peatman (2003) covers the detailed usage of the 18F452 PIC microcontroller but does not cover the necessary general concepts of embedded systems. Simon (1999) focuses too much on the software aspects.

After surveying several text books on the subject, it was decided that since the course is inherently practical in nature, the best thing to do was to use a text book as a main resource plus rely heavily on technical data sheets to enhance the understanding of the material. The chosen text book was Wilmshurst (2001) An

Introduction to the design of small-scale Embedded Systems (Wilmshurst, 2001). Its type of hands-on approach has been proven to significantly enhance the grasping of material by students (Korwin and Ronald, 1990).

**Lab facilities:** Due to the large demand on the course and lab, special lab space and equipment were provided exclusively for its use. Due to financial constraints which are endemic these days in the academic sector and since this is an introductory course and because of the various different backgrounds of the students, ready-made hardware kits were considered for the lab together with simple to grasp and learn software tools. Based on the constraints, it was decided to go with microcontroller-based embedded systems as a basis as it was thought that they were easier to grasp for students.

The next step was to decide on which microcontroller to use. This is a critical step as the choice of a microcontroller will affect how the students

will react towards the course and the lab. There are several criteria which ought to be taken into consideration when making this choice:

- The architecture of the microcontroller: In a practical hands-on embedded systems course, it is important to understand quickly the internal structure as this will allow proper utilization and usage in the lab. Microcontrollers come in mainly two flavors; CISC or RISC. We opted for a RISC-based architecture as this architecture is simple, the instruction set architecture does not contain many addressing modes and therefore easy to understand and the instruction set is limited and straight forward which will allow the students to quickly be able to write control programs and reduce the learning curve
- H/W peripheral richness: Since the lab experiments cover many different topics with different requirements, it is important to find a microcontroller that has on-board as many if not all of the hardware peripherals required such as timer/counters, USART or serial interfaces, A/D converter and PWM generation. This will reduce cost and will allow the students to concentrate on the system design
- Cost and popularity: Although cost is not the prime factor in selecting the microcontroller and it should not become an obstacle for students or faculty, but nevertheless, choosing one at a reasonable cost is logical. Regarding popularity, it is very helpful if the selected controller has a wide base of usage and popular following on the Internet and is supported by many tutorial books and application notes
- Development and debugging tools: These are extremely essential for a successful utilization of the hardware. These tools should be very stable, easy to use and provide all the essential utilities (preferably integrated) to perform development and implementation. These include compilers, cross assemblers, full feature debuggers, in-circuit emulators and simulators
- Performance: The chosen microcontroller should be able to run at a suitable level of execution speed to support the different requirement of the experiments. This means bus speeds, instruction cycle time and interrupt latency are factors that need to be taken into consideration

Based on the above discussion, we decided to use Microchip's 16 series family of PIC microcontrollers as it is very suitable for an introductory course for it

contains many different parts with varying complexity yet a simple instruction set.

**Lab equipment:** In order to focus the experiments on the applications and to reduce cost, we opted to buy readymade kits that contained most of the components necessary to perform the required experiments but was also expandable. After reviewing several kits available in the market, we selected a kit called the Millennium board. We also selected Microchip's PIC Start Plus (2009) as a means for PIC programming. Later, Microchip's PICDEM™ Mechatronic Demonstration Board (2009) and the ICD2 debugger/programmer (2009) were also acquired to complement the previous set of lab kits.

Figure 1 shows the various different hardware kits utilized in the lab. These are used together with PCs that contain the MPLAB IDE (2009) development environment and the necessary software compilers and debuggers. Table 3, lists the main features and capabilities of the kits used.

**Lab experiments:** To complement and practically clarify the ideas presented in the course, a set of experiments have been devised that closely follow the plan laid down for the course. Each student has access to an embedded system design kit plus a PC workstation. Each lab session is three hours long.

Table 3: Specifications of the HW kit used in the lab

---

**The PIC millennium board features**

Sockets for 8, 18, 28 and 40 pin PIC  
Integrated 5V power, can use wall mounted adapter  
4 digit 7-Seg common cathode LED display with multiplexing circuitry  
20x2 line LCD display (based on HD44780 controller)  
4x4 alphanumeric keypad  
On board oscillator sockets and 2 piece oscillators provided (4 and 8 MHz)  
6 LEDs, 4 switches, 4 pushbuttons, 1 potentiometer, Buzzer  
TTL/RS232 level converter  
9 pin-SUBD  
On board breadboard

**Microchip's PICDEM™ mechatronic demonstration board features**

PIC16F917 MCU with an up to 8 MHz internal oscillator and on-board LCD module  
Displays: 8 LEDs and LCD (pre-connected to the LCD module)  
Stepper motor and brushed DC motors  
Analog temperature sensor (Microchip TC1047A)  
Light sensor  
32.768 kHz crystal for time generation  
Human inputs: 4 tactile switches, 2 potentiometers  
Over-current protection circuit  
ICSP™ programming capability and MPLAB® ICD 2 interface connector  
RS-232 interface  
5V regulator  
Optical speed sensor for brushed DC motor  
20-pin socket (compatible with all 8-, 14- and 20-pin Flash devices)

---

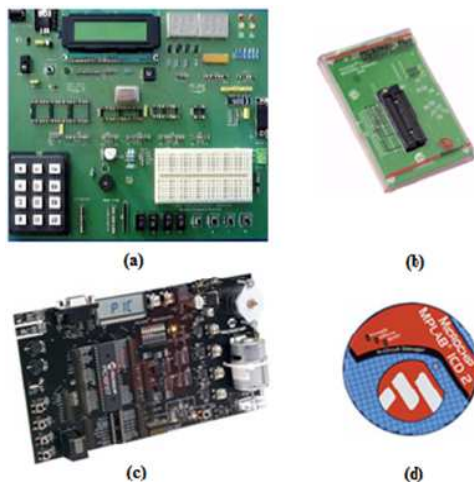


Fig. 1: The hardware used in the Lab. (a) PIC Millennium Board; (b) PIC Start Plus programmer; (c) PICDEM™ Mechatronic demonstration board and (d) ICD2 in-system programmer

Due to the varying backgrounds of the students, a specific style for experiments has been adopted.

Early lab experiments are divided into two sections; the first provides a complete working program fully commented that can be run directly on the target processor to accomplish a specific task. This is followed by questions that probe the student's understanding of the inner workings of the implemented task. The second section will either require the students to use the program of the first part as a template and add extra processes and features, or it will require them to rewrite the code to accomplish the same task using a different available methodology. We have found that this style greatly helps in reducing the learning curve. Below is a description of the main features of the lab sessions:

Weeks 1-4 introduction to Lab. Topics in the Development Environment and Assembly Programming Since the lab is a co-requisite to the theory class, these sessions start by deeply covering assembly programming and IDE simulation skills. This will allow lab sessions to follow orderly the theory class.

Early lab sessions introduce the following:

- Rules and regulations governing the lab as well as safety precautions
- Lab working environment including the hardware kit, the programmer and the MPLAB Integrated Development Environment (IDE).

- Target microcontroller addressing modes, internal registers and memory map
- The instruction set and assembler directives through example programs. These examples are based on mathematical grounds: Basic arithmetic operations, arrays and BCD with emphasis on conditional, comparison and repetition techniques as well as the use of look-up tables
- Programming techniques and methodologies with emphasis on modularity concepts, writing an optimum code and the concept of interrupts
- The whole process of installing and configuring the software, entering a provided code, compiling, debugging methodologies programming a microcontroller using the programmer and finally getting the kit to work with the programmed part to achieve the objective

At the end of the lab, students are presented with lab sheets which are full of related questions which the student will be able to solve after running and simulating the example codes.

Weeks 5-8 introduction to Embedded Systems Hardware/Software Design and Basic HMI.

These lab sessions concentrate on empowering students with logical and analytical skills to solve real life system design problems. This part is concerned with introducing the following:

- Off-the-shelf components and ICs covering regulators, resonators, encoders, decoders, flip flops, 7-segments, LEDs, switches, push buttons, LCDs and keypads. This will ensure that all students from different backgrounds are brought to the same level of knowledge-base required
- Illustrating a simplified example drawn from an industrial automation background such as a filling machine
- Guiding students through the steps of system requirement analysis and definition, program flow analysis and flowchart design, system and subsystem design
- Coding the design and building the hardware
- Familiarizing the students with the concept of unit and system testing through simulation

Weeks 9-12 introduction to Embedded Systems Peripherals

Through these set of lab sessions, embedded systems hardware as well as the microcontrollers built-in peripherals are emphasized. These experiments are concerned with the following:

- Introducing the built-in 8/16 bit timers which are employed in various examples such as digital clocks, time lapse measurements and timed interrupts
- Intersystem communication in modern embedded systems, the USART peripheral as an example covering Master/Slave communication and communicating with PC through hyper terminal
- Introducing the A/D module, software and hardware Pulse Width Modulation (PWM) techniques in an example which drives a DC motor based on the temperature acquired and digitized through the ADC in an open loop control scenario. Optical speed sensor circuit are presented and used to measure the speed of rotation

**Lab project:** To make it possible for the students to grasp and practically implement the ideas learned during the course and lab sessions, the students are required to design, implement and then demonstrate a working embedded system project. Two approaches to the format of the project were tested:

The first approach had the format for the project quite loose; the student was allowed to choose his/her own project with a minimum of constraints. The motivation for adopting such an approach is as follows:

- Encourage and allow the students to use their own ingenuity, creativity and ideas
- Allow the students to be exposed to quite a variety of applications and implementations as the students demonstrate and present their work in front of others
- Let the students take direct full control of their projects from idea to concept to design to final implementation therefore mimicking real-life situations
- Allow the students the freedom to use their own tools, choose their own work methodology and their own system design criteria
- Allow students to work in groups

On the other hand, this approach presented some problems.

Due to the openness and few restrictions imposed, it was quite difficult to provide the required infrastructure to support the high diversity of requirements to support the different projects. However, most students were so enthusiastic that they supplied their own supplies at their own expense. We recognized that this problem caused some concern to some students, but we solved it by making things clear from the beginning regarding

our abilities and left the decision to proceed with the project to the students. Since the project is solely the responsibility of the student, time was also of concern due to the large number of students and our limited space and equipment in the lab. This was solved with the help of the students as we scheduled lab openings throughout the day all week long.

This approach was tested for two years with mixed results. For many students this approach has proved very successful as they managed to produce some very interesting and sometimes even ingenious projects using low-cost hardware platforms. For others, the approach proved difficult. From our monitoring and requested anonymous written feedback, we concluded that the majority of the problems were to do with the issue of project choice. This was due to the requirement that each student or group had to choose a different project. This, overtime, led to the fact that many of the well known ideas got selected and the students quickly ran out of unique simple ideas.

The later approach adopted was to provide all students with a unified basic project idea which students will develop and submit as a minimum. No further restrictions were imposed. This allowed the project to be of a highly competitive nature. Students submitting projects fulfilling the above criteria with fully working extra features and adhering to best system design practices were awarded with bonus marks. Projects with similar features were weighed against each other in terms of code simplicity, program space, efficiency and hardware cost. Projects following optimum system design and implementation methodologies were also rewarded.

However, two main problems were observed in this approach, first of which is the similar feature set which the students come up with. This was a clear sign of the limited creativity that this approach forces. Sadly, we have also observed some occurrences of plagiarism. We sought to address and limit this problem by introducing a project timeline where students should submit their project in phases. This will allow us to closely monitor student's progress from the early stages and help those with difficulties rather than letting them revert to plagiarism.

**Final project sample:** The lab projects exemplify good embedded system design practices and demonstrate the effectiveness of the lab. It also shows that the objectives and skills which the lab addresses were successfully met. Students involved in these projects demonstrated through their fully working prototypes thorough understanding and knowledge of assembly programming, system design models and practices, I/O interfacing and electronics.

As an example of the second approach utilized, one of these projects called Amazons Board Game is described here and its main features are highlighted:

The project is an embedded systems implementation of the Amazon board game. The game is played on a 10 by 10 chess board. Each player has four queens which can be moved about the board. Each move consists of moving a queen along a straight line to a new square and then firing an arrow along a straight line and destroying the square where it lands. Queens and arrows may not move over queens or destroyed squares.

The design was based on the 16F877A microcontroller and it utilized around 300 LEDs and 20 push buttons. Clearly the I/O interfacing proved to be quite a challenge in the design process, so students explored the potential of using various decoder and encoder ICs to implement their design. This has also raised the level of complexity of software design and stressed the importance of the software/hardware co-design process. Students from multi-disciplines were encouraged to form groups and share their knowledge.

Figure 2, shows a sample of a project designed and implemented by a group of students.

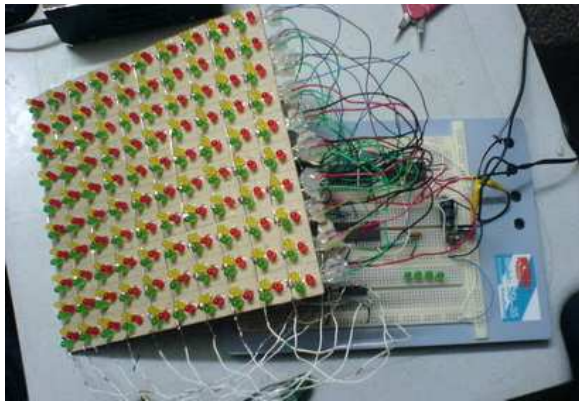


Fig. 2: Lab project example: Amazons board game

## RESULTS

**Overall assessment:** Anonymous written feedback collected from 330 students over a period spanning one and a half academic years from February 2008 to June 2009 has shown that around 70-80% of the students taking the lab are greatly satisfied with the embedded systems design experience they have obtained. Table 4 lists the results of the anonymous feedback. It has also been shown that the Demonstrate, Work-to-Know and Design approach adopted in laying out the structure of the experiments significantly reduces the learning curve and always keeps students intrigued and excited. We have concluded through student feedback that the lab successfully met the objectives it was designed for. However, continuous refinement of lab experiments is an on-going process. The breadth and depth of knowledge presented, though overwhelming at first, is easily overcome by careful and continuous guidance in each stage and by presenting a challenging environment for students to work in.

Furthermore, as students are now equipped with in-depth knowledge of embedded systems software and hardware co-design concepts and driven by the knowledge gained from the lab project, many choose to base their final year capstone graduation projects on embedded systems. Over the years, senior students have developed complex control, monitoring and communication systems with a microcontroller in the heart of the design. Many of these projects found their way to the top 10% of projects submitted to the departments of computer, electrical and mechatronic engineering since the introduction of the course and lab into the curriculum. Some projects participated in the National Technology Parade (2008)-An annual event organized by leading Jordanian universities, IEEE Jordan and the United Nations Development Fund for Women. These projects were awarded national prizes of the first degree. Moreover, many other embedded-based projects won the Jordanian Engineers Association (JEA) and other company supported awards (2010).

Table 4: Anonymous written feedback results (Feb. 2008- June 2009)

	Excel. (%)	Good (%)	Satisf. (%)	Poor (%)
The lab is well-equipped in terms of educational resources and equipment	45	29	21	5
The lab makes use of modern e-learning techniques	41	28	7	24
The H/W kits and IDEs are easy to use	14	70	14	2
The experiments are well-structured and easy to understand	22	56	13	9
Examples are sufficient to grasp the experiment ideas	22	63	5	10
Difficulty level of the exercises presented in the lab sheet varies from easy to challenging	19	35	26	20
Experiments follow the theory classes	11	38	22	29
Satisfaction with project idea	31	42	20	17
Is time (7 weeks) allotted to complete the project enough?	17	23	43	17
Is the project cost suitable to your budget?	6	16	52	26
Rate the cooperation and guidance of instructors, lab engineers and TAs	61	27	8	4



## DISCUSSION

As the survey results indicate and also based on the exponential growth of the utilization of embedded processing systems in final year capstone projects, it is fair to say that the embedded systems course and its associated lab have fulfilled their objectives. Although, the course is seen as tough in the beginning by many students, but the Demonstrate, Work-to-Know and Design methodology has proven to ease the learning curve and provide an enjoyable and rewarding learning environment.

## CONCLUSION

In this paper, the importance of introducing embedded systems in current engineering curricula is highlighted. The process and experience of designing and developing an embedded systems course that serves different disciplines with different backgrounds and requirements at the University of Jordan together with its associated laboratory is reported. Emphasis is given throughout to the design methodology used for the laboratory facilities and the learning outcomes are explored. Successful design projects undertaken by students and their results are reported.

## REFERENCES

- Department of Computer Engineering, University of Jordan Awards website, 2010. <http://www.ju.edu.jo/faculties/FacultyofEngineering/computer/Pages/Awards.aspx>
- IEEE Computer Society/ACM Task Force on Computing Curriculum, 2004. Computing curricula-computer engineering. Final Report, <http://www.eng.auburn.edu/ece/CCCE/>
- Jackson D.J. and P. Caspi, 2005. Embedded systems education: future directions, initiatives and cooperation, *ACM SIGBED Rev.*, 2: 1-4. DOI: 10.1145/1121812.1121814
- Korwin, A.R. and R.E. Jones, 1990. Do hands-on, technology-based activities enhance learning by reinforcing cognitive knowledge and retention? *J. Technol. Educ.*, 1: 1-12. <http://scholar.lib.vt.edu/ejournals/JTE/v1n2/pdf/jones.pdf>
- MPLAB ICD 2, 2009. Microchip Technology Inc. [http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=1406&dDocName=en010046&part=DV164005](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en010046&part=DV164005)
- MPLAB IDE, 2009. Microchip Technology Inc, [http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=1406&dDocName=en019469&part=SW007002](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469&part=SW007002)
- National Technology Parade website, 2008. <http://fetweb.ju.edu.jo/tp/>
- Peatman, J.B., 2003. *Embedded Design with the PIC18F452 Microcontroller*, Prentice Hall, ISBN: 10: 0130462136.
- PICDEM™ Mechatronic Demonstration Board, 2009. Microchip Technology Inc. [http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=1406&dDocName=en023837](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en023837)
- PICSTART Plus, 2009. Microchip Technology Inc, [http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=1406&dDocName=en010020](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en010020)
- Simon D.E., 1999. *An Embedded Software Primer*, Addison-Wesley, Reading, Massachusetts. ISBN10:0-201-61569-X
- Vahid F. and T.G. Givargis, 2002. *Embedded System Design: A Unified Hardware/Software Introduction*. John Wiley and Sons, New York, ISBN: 978-0-471-38678-0.
- Wilmshurst, T., 2001. *An Introduction to the Design of Small-Scale Embedded Systems*, Palgrave Macmillan, England. ISBN: 10: 0333929942.