Effective Active Learning Tools for an Embedded Systems Course

Girma S. Tewolde, Ph.D.
Electrical and Computer Engineering Department
College of Engineering, Kettering University
Flint, MI 48504

Abstract— **This paper presents tools the author uses to enhance student motivation in a microcontroller-based embedded systems course. The course is offered as part of a requirement in a computer engineering degree program. In the traditional lecture-based teaching and learning process, information typically flows in one direction with very little active involvement by the students. There are a number of techniques available in the literature that present the use of active learning techniques in the classroom. To take this active learning approach to the next level, this paper presents three important tools that are utilized in a systematic way to further enhance the students' role in their learning. The first tool uses laboratory assignments for practical hands-on activities that help reinforce the theoretical concepts covered in the classroom. The second tool uses "peer-teaching" technique in which students take active role in learning a topic and present it to the class, followed by questions, discussions, and quiz. The third tool is "self-proposed" final course project in which students are expected to apply their creativity in coming up with a proposal for a project of their own choice (with the instructor approval), implement the project and present it to the class.**

Keywords— active-learning; peer-teaching; embedded system

I. Introduction

Embedded systems courses form the core of the curriculum of most computer engineering, electrical engineering, and computer science degree programs. One of the primary learning objectives of this course in our institute is to enable students to use a microcontroller as part of a system to build intelligent sensing and control applications. There is a fast growing demand in the industry for students who have training in this area. The job outlook for software developers, including embedded systems software, is expected to grow at a much faster than average rate of 17% from 2014 to 2024 [1,2]. There is an increasing interest in turning everything people interact with on a daily basis into 'smart things' and automating everyday chores in households and service sectors [3,4,5]. Examples of such 'smart things' that people are already getting exposed to include smart thermostat, smart appliances, robotic vacuums, smart cars with advanced driver assist features and future potential for automated driving.

In most of the smart systems the central element of the system that handles the decision making task is a microprocessor or microcontroller. Students need a fundamental knowledge on the principle of operation and application development for this critical component. It is also critical to develop skills in hardware and software interfacing of the microcontroller to external devices such as sensors, actuators, communication devices, and user interface elements.

Recent advances in cloud-based services and the fierce competition among the companies operating in this domain is driving cost of developing products that take advantage of connectivity through the cloud simpler than ever before. New development hardware modules come with networking facilities such as WiFi and Bluetooth low energy (BLE) that reduce the complexity of the system for cloud-based solutions [6,7]. Taking advantage of these hardware and software advancements, new applications in the domain of internet-of-things (IoT) have recently been attracting huge interest for commercial, industrial, automotive, household, agricultural, and other application domains.

To satisfy the needs of the embedded systems industry, students need to be prepared with strong theoretical foundations as well as practical skills on the use of modern tools for the development of embedded systems. Studies show that traditional teaching methods that primarily rely only on non-interactive and passive approaches such as lectures, homework assignments, and tests do not produce effective results [8]. Active engagement of the students using well designed classroom and outside activities is found to contribute to enhanced motivation and better learning experience [9].

This paper presents three active learning tools that the author used in his embedded systems course: a) practical lab activities, 2) peer-teaching assignments, 3) self-selected final project. The rest of the paper is organized as follows. In section II the embedded systems course is briefly presented. Section III presents the lab component of the course. Section IV presents the peer-teaching component of the course. Section V presents the final project component. Section VI discusses about the assessment followed by conclusion in Section VII.

II. Course Information

The embedded systems course presented here focuses on contemporary 16 and 32 bit general purpose microcontroller architectures. It uses Microchip's PIC24 [10] and PIC32 [11] MCUs, although the concepts taught also apply to most other general purpose MCUs with appropriate modifications in the hardware specifics.

The course is presented in a combination of lectures and lab assignments. There are five contact hours per week which are

978-1-5090-5920-1/17/$31.00 ©2017 IEEE

made up of three 1-hour lectures and one 2-hour lab. The lectures present the theoretical material about embedded system design; go over details of the target microcontroller architecture, the interfacing modules, and the programming techniques. A list of the course topics presented is given below:

a. Introduction to embedded systems
b. PIC24 architecture and programming
c. PIC32 architecture and programming
d. Input/output interface modules
e. Interrupts
f. Timer modules
g. Timer functions – input capture and output compare
h. Real-time clock and calendar
i. Serial communication protocols: UART, SPI, I2C
j. Parallel master port
k. Introduction to Controller Area Network (CAN)
l. Embedded wireless connectivity
m. Analog to digital converter

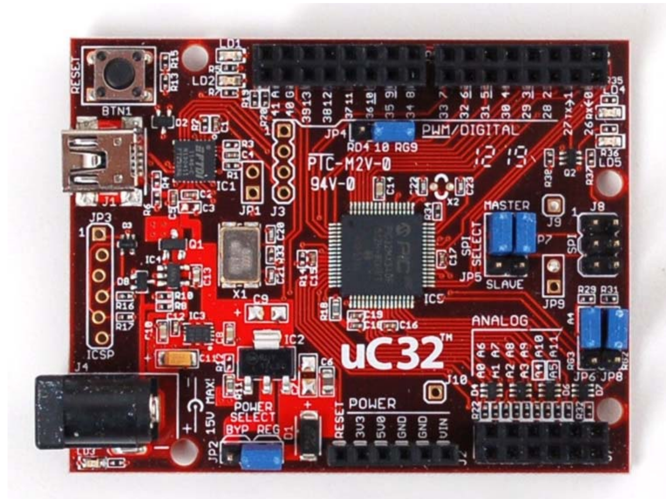
In addition to the lecture materials on the essential MCU architecture, programming techniques, and various built-in peripheral interface modules, the course offers several practical laboratory activities that help enhance the understanding of the concepts and demonstrate real-world applications. The lab exercises include activities on low level programming to give insights on how the processor operates, manipulates data between different types of storages, and see how the compiler manages the hardware resources, etc. But for the most part the laboratory activities focus on peripheral interfacing techniques to let students explore ways the MCU communicates to common input/output devices in embedded environments.

III. Lab Hardware Kit and Assignments

A. Lab hardware kit

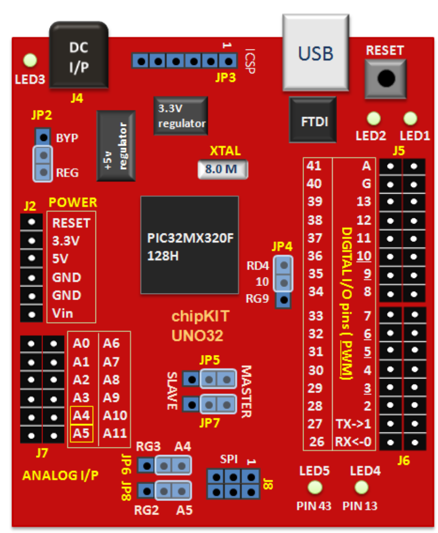
The course instructor explored several options to identify appropriate lab kit to support the course. It was important to choose low-cost and easily accessible hardware and software kits. The current lab set-up is based on the chipKIT uc32 low-cost kit [12]. This kit has the same footprint as the popular Arduino Uno board, which most students are familiar with. However, it is more powerful than Arduino Uno, with 32-bit processor running at 80 MHz clock and a much larger program and data memory. Microchip offers free version of its industry standard MPLAB-X [14] integrated development environment and compilers. The built-in simulation tool in MPLAB-X simplifies the software development process. Hardware programming and debugging on a target device can be done using a low-cost chipKIT PGM programmer/debugger device [15].

Since the chipKIT uc32 board comes with only limited on-board peripheral hardware, we developed an input/output (I/O) shield to support a few practical lab assignments. This shield includes 8 LEDs, two pushbutton switches, 4 seven-segment displays that share common I/O ports for data with separate control signals, and a Bluetooth module. Figure 1 shows the chipKIT uc32 board, the PGM – programmer/debugger module, and the I/O pin out of the microcontroller board. Figure 2 shows the in-house developed I/O shield.



(a) ChipKIT uc32 [12]

(b) chipKIT PGM Programmer/Debugger [15]



(c) Pinout of chipKIT UNO32 and uc32 [13]

Figure 1. ChipKIT uc32

B. Lab assignments

The lab assignments focus on the fundamental microcontroller hardware concepts and basic interfacing techniques. The list of the lab assignments is given below:

1. Introduction to the MPLAB-X and program development process (using simulator)
2. Assembly and C programming of PIC24 (using simulator)
3. Basic I/O interfacing with switches and LEDs using the chipKIT uc32 low-cost hardware kit
4. Interfacing Keypad, and sending data over UART
5. Implementing timer interrupts for interfacing displays and handling tasks in a kitchen-timer
6. Implementing Bluetooth interface to provide remote access for a timer application
7. Implementing a reaction-game using change notification feature on input pins of the microcontroller

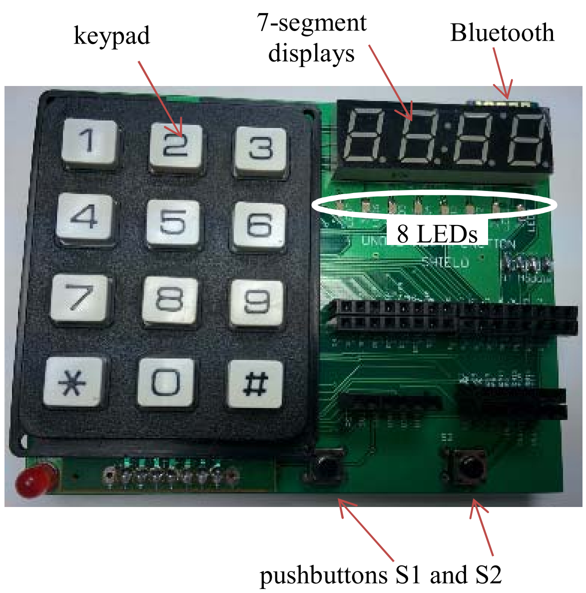


Figure 2. I/O shield for chipKIT uc32/UNO32

IV. Peer Teaching

One of the objectives of the embedded systems course is to develop the students' research, self-learning, critical thinking and communication skills. In addition to targeting these skills, the peer-teaching component of the course is aimed at motivating the students to take active role in the teaching and learning activities. It is meant to take students out of their passive comfort zone and challenging them to take the lead in learning an assigned course topic and teaching it to the class. The inspiration for peer-teaching came from previous research in the literature [16], which describes the effectiveness of the method in actively engaging the participants and enhancing learning in both the peer-teachers and learners.

After the first four weeks of the course students will have built the necessary background to make informed decisions about the topics they would like to investigate further. Most of the peer-teaching topics identified are microcontroller peripheral modules, such as the various serial communication interface protocols, analog interfacing, and wireless communication. The instructor provides guidance to the students as they conduct the research and prepare their teaching materials. Students are required to submit electronic copy of their presentation three days ahead to the professor for review and feedback. The presenters also prepare short quiz, which could be modified by the professor, and given to the class at the end of the presentations. The presentations typically run for about 30 minutes, followed by 15 minutes of questions and discussions, with the last 15 minutes left for quiz.

V. Final Projects

After completion of the lab assignments described in Section III, the students start working on a final term project. The main objective of the project is to give the students opportunity to apply what they have learned in the course to real projects of their own choice in an innovative way. The main requirements for the project are that it has to have a microcontroller, input/output devices, an intuitive user interface, and most importantly the project needs to have a real value. The students do not necessarily have to use the same processor as the one covered in the course.

There is a strong emphasis to inspire the students to entrepreneurial mindset [17] by motivating them to think creatively to identify problems and build solutions that could create values. During the first week of the project, the students, working in groups of two, need to do preliminary research about the problem they want to focus on, develop a proposal and make a short presentation of their ideas to the class. The instructor and other students in the class provide constructive feedback to the students for improving the project ideas.

The main reason for allowing the students to self-select their project ideas is so that they own their project, i.e. they don't feel like they are being forced to work on someone else's ideas. Thus, they get excited about the project, they put a great deal of effort to see the project succeed, they know what they are expecting as a result by the completion of the project, and they want to prove that they are capable of achieving what they set their minds on. This approach has been successful as has been demonstrated with results from several terms of the course. Below is a short list of the student projects that were created in the past few semesters:

a) Brain trainer v2.0
b) Playing agent for Connect 4
c) PIC32 for mobile roboat control
d) Cloud based low-cost home automation and monitoring
e) Fingerprint door lock
f) Smart Nerf gun
g) Drink mixer machine
h) Vehicular communication using BLE
i) Audio spectrum visualyzer display
j) Automatic farming system
k) Newtorked motor controllers for robotic applications
l) Spinning LED clock
m) Robot arm vs. Human tic-tac-toe game
n) Dashboard assistant using on-board diagnostics
o) LED matrix backpack

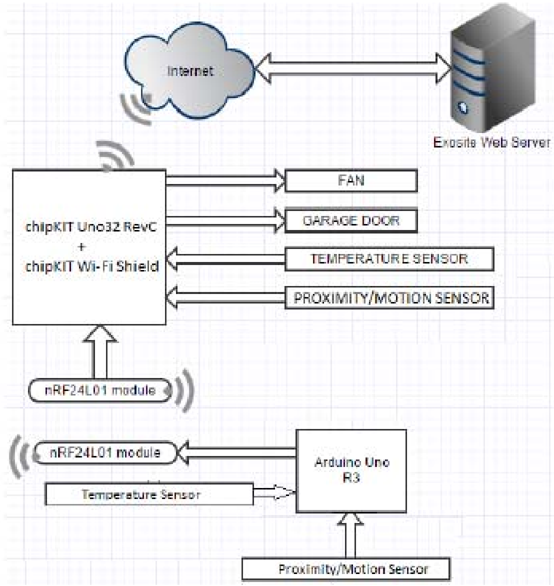


Figure 3: Block diagram of the home automation project

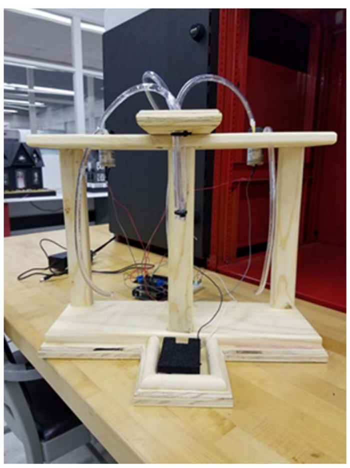


Figure 4: Drink mixing machine project

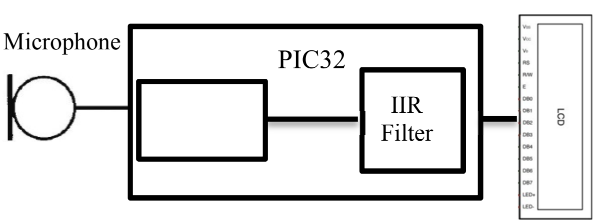


Figure 5: Audio spectrum visualizer project

VI. Assessment

There are two assessment tools that are used for the course. The university has a standard online course assessment that all students complete at the end of each school term. Since the university wide common course assessment survey does not target the active learning tools we want to evaluate, a separate survey had to be developed primarily for that purpose. We use SII (Strength-Improvement-Insight) survey to collect feedback about the students' experience with the active learning tools. This survey specifically targets the peer-teaching, labs and final projects in the course. The questions are posed in open ended form so students can express their opinion about their experiences with those active learning activities, and give feedback to help improve the course in the future. For example, in the first offering of the peer-teaching activity an important improvement suggested by a number of students was to add a quiz at the end of each presentation to make sure the class is paying good attention to the peer presentations. A few quotes of students' comments are given below:

A. On labs

Strengths: "*Very hands 0n. Create our own learning. Fun. Feeling of accomplishment.*"

Improvement: "*Keep it basic. May be introduce more peripheral devices for fun.*"

Insight: "*Be prepared to lab.*" "*I learned a lot and put a lot of work into it. Overall it was a great experience.*"

B. On peer-teaching:

Strength: "*Explaining a concept always helps me retain information, so I have a better understanding of my topic following the peer-teaching period.*"

Improvement: "*I did get frustrated at one point because the topic scope went beyond the course and I wasn't sure what was most important to understand.*"

Insight: "*Best way to learn something is to try to teach it.*"

C. On final projects:

Strength: "*Allows students to have more freedom and be innovative.*" "*It allows creative design to allow students to pursue a project they are more likely to have a personal vested interest in.*"

Improvement: "*Help students reign in the project to realistic proportions for the time they have to work on it. Do more planning/milestones to keep students on track.*" "*More required meetings. Time management is the hardest part to it.*"

Insight: "*The project allows students who have passion to really continue on beyond a normal classroom limitation and really explore the potential of that area of study ... in this case microcomputers.*"

VII. Conclusions

The paper presented effective active learning tools introduced in a microcontroller based embedded systems course at our institution. Hands-on lab assignments, peer-teaching, and self-selected final course projects were used as the primary tools for the active learning experience. The assessment results and the feedback received from the students at the completion of the course demonstrate that the tools employed in the course achieved the intended goals by providing opportunities for active engagement in classroom and lab, and motivating the students to be innovative in their design projects.

References

[1] Bureau of Labor Statistics. https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm

[2] Salary and Career Info for Embedded Systems Engineers. http://study.com/articles/Salary_and_Career_Info_for_Embedded_Systems_Engineers.html

[3] Grimm, C., Neumann, P., and Mahlknecht, S. (Eds.), Embedded Systems for Smart Appliances and Energy Management, Springer, 2013.

[4] White Paper, Intel Embedded Processors, "Rise of the Embedded Internet," 2009, Intel Corporation. http://download.intel.com/newsroom/kits/embedded/pdfs/ECG_WhitePaper.pdf

[5] Raghavan, S. and Tewolde, G. S., "Cloud Based low-cost Home Monitoring and Automation System," Proceedings of the 2015 ASEE North Central Section Conference.

[6] Intel Edison Kit for Arduino, Intel. http://www.intel.com/buy/us/en/product/emergingtechnologies/intel-edison-kit-462187

[7] chipKIT Wi-Fire: WiFi Enabled PIC32MZ Microcontroller Board. http://store.digilentinc.com/chipkit-wi-fire-wifi-enabled-mz-microcontroller-board/

[8] Harris, P. and Johnson, R., "Non-traditional Teaching and Learning Strategies," teaching leaning committee, Montana State University. http://www.montana.edu/facultyexcellence/Papers/activelearn2.pdf

[9] Wood, K., & Jensen, D., & Dutson, A., & Green, M. (2003, June), "Active Learning Approaches In Engineering Design Courses," Proceedings of the 2003 ASEE Annual Conference & Exposition.

[10] 16-bit PIC24 MCUs and dsPIC DSCs, Microchip Inc. http://www.microchip.com/design-centers/16-bit

[11] Microchip's 32-bit MCUs, Microchip Inc. http://www.microchip.com/design-centers/32-bit

[12] chipKIT uc32: Basic Microcontroller Board with Uno R3 Headers, Digilent Inc. http://store.digilentinc.com/chipkit-uc32-basic-microcontroller-board-with-uno-r3-headers/

[13] Exploring the chipKIT Uno32 Board. http://embedded-lab.com/blog/exploring-the-chipkit-uno32-board/

[14] MPLAB-X Integrated Development Environment (IDE), Microchip Inc. http://www.microchip.com/mplab/mplab-x-ide

[15] chipKIT PGM Programmer/Debugger, Digilent Inc. https://reference.digilentinc.com/chipkit_pgm:chipkit_pgm

[16] Whitman, Neal A. *Peer Teaching: To Teach is to Learn Twice*. ASHE-ERIC Higher Education Report No. 4. Washington, D.C.: Association for the Study of Higher Education, 1988

[17] Kriewall, T. and Mekemson, K., "Instilling the Entrepreneurial Mindset into Engineering Undergraduates," JEEN: The Journal of Engineering Entrepreneurship, Vol. 1, No. 1, July 2010.