Evolution of Microcontroller's Course under the Influence of Arduino

Oscar Acevedo Patiño, PhD, Sonia Contreras-Ortiz, PhD, and Juan Carlos Martínez-Santos, PhD
Universidad Tecnológica de Bolívar, Colombia
oacevedo@unitecnologica.edu.co, scontreras@unitecnologica.edu.co, jcmartinezs@unitecnologica.edu.co

Abstract– This paper presents the evolution of an undergraduate microcontroller's course that uses Arduino as its main platform. In earlier versions of the course, it focused on teaching Arduino's framework and its interconnection to compatible hardware. This allowed rapid familiarization with the hardware and software, but students were losing the capability to design their own prototypes. So, we tried a different approach that still uses Arduino, but includes a deeper study of technical details to have more control on the device. This paper presents the first comprehensive analysis of these methodologies. The proposed course starts studying Arduino's platform as a tool to get familiar with the hardware and software interface. On the second part, students gain a deeper understanding of the Arduino framework, and are able to configure its main features. On the last part, students put the Arduino framework aside and focus on the microcontroller to learn the principal features of its architecture. We observed that the evolution of the course has helped to increase the student's skills and motivation.

Keywords-- Embedded systems, microcontrollers, Arduino, project-based learning.

Digital Object Identifier
(DOI):http://dx.doi.org/10.18687/LACCEI2016.1.1.183
ISBN: 978-0-9822896-9-3
ISSN: 2414-6390

Evolution of Microcontroller's Course under the Influence of Arduino

Oscar Acevedo Patiño, PhD, Sonia Contreras-Ortiz, PhD, and Juan Carlos Martínez-Santos, PhD
Universidad Tecnológica de Bolívar, Colombia
oacevedo@unitecnologica.edu.co, scontreras@unitecnologica.edu.co, jcmartinezs@unitecnologica.edu.co

Abstract*– This paper presents the evolution of an undergraduate microcontroller's course that uses Arduino as its main platform. In earlier versions of the course, it focused on teaching Arduino's framework and its interconnection to compatible hardware. This allowed rapid familiarization with the hardware and software, but students were losing the capability to design their own prototypes. So, we tried a different approach that still uses Arduino, but includes a deeper study of technical details to have more control on the device. This paper presents the first comprehensive analysis of these methodologies. The proposed course starts studying Arduino's platform as a tool to get familiar with the hardware and software interface. On the second part, students gain a deeper understanding of the Arduino framework, and are able to configure its main features. On the last part, students put the Arduino framework aside and focus on the microcontroller to learn the principal features of its architecture. We observed that the evolution of the course has helped to increase the student's skills and motivation.

Keywords*-- Embedded systems, microcontrollers, Arduino, project-based learning.

I. Introduction

It is common to have at least one course related to microcontrollers in the ECE curriculum. Examples of these courses include embedded systems, microcomputer interfaces, and robotics [1]. In some courses, microcontrollers are the preferred platform to develop systems to test principles taught in class. Other courses focus on the study of the device and the way to interface it with the outside world. This is the case of ECE1463, the Microcontrollers course taught at Universidad Tecnológica de Bolívar. In this course, the microcontroller is the main component to develop system prototypes to address engineering design problems.

The course started around twenty years ago by using Microchip PIC microcontrollers. At that time, there were no C/C++ compilers available, so the programming was done in assembly language. The PIC microcontroller has an RISC architecture that features only 20 instructions, so it was not hard to get familiar with it. However, register management to use embedded peripheral devices was a more difficult task. In addition, there were no code libraries for using external devices such as liquid crystal display (LCD), so they had to be coded by the user. Later came the C/C++ compilers and libraries for multiple functions that made prototype development easier. The arrival of new microcontroller brands such as Freescale and Texas Instrument strengthens the course as the topics became more general and less dependent on the microcontroller brand used.

Now, we have frameworks developed and supported by the e-community. For example, Arduino is an open-source hardware/software platform for the development of electronic systems based on Atmel microcontrollers. It started in 2005, and in 2012 was adopted as the platform for the Microcontrollers course at Universidad Tecnológica de Bolívar. It has several advantages over other microcontroller platforms. The hardware is low price, and the software IDE can be downloaded for free. The programmer is built-in the board, so no extra hardware is required to program the device. Additional hardware is available on the form of expansion boards (shields) to connect to peripheral devices such as Ethernet adapters, wireless communication, RFID, Bluetooth, GPS, and many other devices with their respective drives. Arduino's software includes a variety of libraries for multiple purposes. Finally, there is a large amount of information available to the designer to build different applications.

After several course releases, it was noted that the applications developed by the students reached a high quality level. Most of them included sensors of different types, wireless transmission to other devices, and Internet connection. However, it was also noted that the students had difficulties when they needed to go outside the Arduino's world. It seems they struggle to develop applications (hardware and software) that are not present on the vast information network for Arduino. The feeling was that the students became too dependent on what was available from Arduino and its community.

This paper presents the experiences gathered from teaching the ECE1463 Microcontrollers course at the Universidad Tecnológica de Bolívar, which uses the Arduino board as its development platform. The advantages and the problems observed during the last three years are presented. It is also discussed how to take advantage of the Arduino's platform and take it to the next level. Students should recognize the microcontroller as the device behind Arduino and learn how to use it with more specialized tools in order to develop optimized engineering projects.

The remainder of this paper is organized as follows. Section II summarizes the related work. Section III describes the challenges of the course. Section IV introduces our pedagogical approach of the course. Section V describes the microcontroller course at the Universidad Tecnológica de Bolívar. Section VI shows the student evaluation of the

Digital Object Identifier (DOI): http://dx.doi.org/10.18687/LACCEI2016.1.1.183
ISBN: 978-0-9822896-9-3
ISSN: 2414-6390

course. Finally, Section VII and Section VIII present discussions and conclude the paper.

I. Related Work

An initial review of the related work shows that several embedded systems courses use a variety of hardware such as microprocessors, microcontrollers and FPGAs in their laboratory and project assignments [2]. We selected some previous works that use Arduino as the main platform for the course.

In Jamieson [3], the author presents a positive experience with the use of Arduino as its main platform, where, in addition to the commonly expected skill, other abilities like code understanding and system integration are also developed. However, the author poses questions about how much contribution for the project comes from the student and whether the level of their work was deep enough to be considered a successful learning.

The Arduino board is used as the platform in [1]. According to the authors, this is a course where students are able to build hardware/software systems that incorporate design patterns, multi-threading, embedded programming, and wireless communication in an effective manner within an one-semester course.

Sometimes, the usage of the Arduino's on-line community has replaced demonstration laboratories as mentioned by Chancharoen [4]. It is claimed by the authors that students accelerate the study by learning from the on-line community resources in parallel to the lectures.

Using Arduino in early stages helps students to stay motivated. In Spain, a cooperative project developed by a network of schools and companies of different regions has been using Arduino for easy development of new applications and hardware modules in automation courses [5]. Even though the results are interesting, there is the requirement that any additional device must be adapted to working within the network, and this can be difficult with devices that are not fully compatible with Arduino.

In addition to Arduino-like shields, other platforms have been developed based on the same framework. This is the case for the lab kit from Columbia University [6]. These kits allow students to test and control several hardware devices such as displays, leds, switches and some sensors. This approach enables rapid developing, but at the same time limits the student's options. According to [6], this is because all the practices are based on the Arduino-like kit.

From previous works, it can be seen that the course goals are what really defines whether Arduino is the right tool for a course. However, even in more advanced courses where Arduino may not be considered the most suitable tool, it still can be used with very good results, as we show next.

III. PROBLEM IDENTIFICATION

Arduino is a very popular platform for creating projects. The reasons for this lie on the hardware that can be attached to the main board, the code libraries, and the information available as books, tutorials, videos, do-it-yourself projects, and the e-community. In order to be a platform accessible for everybody, (i.e. non-engineers), Arduino's developers decided to hide most microcontroller technical details behind library functions and expansion boards. These functions and boards take care of all implementation and configuration details required to run a sketch (application program) in the microcontroller and translate these tasks to simple program commands.

Although many design problems are solved successfully, it was also noted that when a design problem that required something different to what was available: a modification in a hidden parameter, or the use of peripherals without a shield, it became a challenge for the student, with a moderate chance for a successful end. A talk with students facing these problems revealed a dependency on the information from the Arduino's community. The students did not go outside this world to find out what was required to solve their problem. The conclusion from the talk was that some of the nice features provided by the Arduino's community became a hindrance for an engineering student when a specialized or optimized solution was required.

All the features, provided by Arduino and its community, are good for someone starting on electronics and embedded systems. However, this may not be the case for engineering students, because they need not only to solve a design problem, they need to solve it in a more efficient way. This means, to develop a functionally correct system under certain constrains like low power consumption, reduced execution time, and minimum hardware. These performance goals are not easy to handle within the Arduino's environment, because the implementation details are hidden to the user, and they are not as ease to access and modify for specific situations as the regular libraries. The problems when using the Arduino's platform can be divided into four categories: code-related problems, hardware-related problem, system-level related problems, and pedagogical issues. In the following paragraphs, these issues are addressed in detail.

A. Code Related Problems

There are two functions to write a program: *setup( )* and *loop( )*. The former is used for hardware setup and variable initialization. The most common task for hardware setup is to define whether a port will be used as input or output. The latter function is the equivalent to the *main( )* function in any C/C++ program. Of course, it is possible to add more functions (in the same file) to the program. The Arduino's environment, is based on Java, which has a *main( )* function where it starts execution. This is an indication that the user actually has no access to the real *main( )* function of the program. His code is embedded on a larger program, and it is hard to tell what other functions and libraries are added to the user program because its code is hidden to the user. This works against code optimization and the microcontroller's

memory usage, which is a critical constrain in microcontroller-based systems.

Writing complex programs is another difficult task in Arduino's programming environment. Good programming practices state that a large code should be divided into different files, so it can be handled easily. Header files should be created to describe classes, data structures, constants and any relevant information. It is not evident how to do this on Arduino's IDE. It seems the user needs to write the whole code in a single file, which makes the program long and clutter. This is not a good programming practice for either a single programmer or a programming team.

Code optimization is also a difficult task on Arduino's environment. The ATmega328, the microcontroller inside the Arduino's board, has a limited memory (32K) to store the program. A mid-range application program could easily require this memory amount. It means the programmer must take advantage of each memory byte present. In other words, the application code should be optimized, so it can fit on the available memory. There is no tool to perform this optimization on the Arduino's environment. In addition, the hidden code added by the Arduino's environment, makes it hard to optimize and measure the real amount of memory required by the application.

Code debugging is another difficult task on Arduino's environment. There is no tool for this purpose. This situation, added to a large single file, could make the task of finding an error a real challenge.

B. *Hardware Related Problems*

The problem with hardware rises from the need to adapt the systems behavior to specific design problems. A typical situation is the clock frequency, which may be set to a different value to the Arduino's current clock frequency in order to reduce power consumption. Any variation on clock frequency compromises the *delay( )* function that is the baseline time in the stack libraries and also for several external devices that depends on the fixed clock frequency to work as expected.

A fixed clock frequency also affects the TIMER modules, which are used to generate a pulse-width modulation (PWM) signal. The PWM is commonly used for emulating a variable DC output voltage. In Arduino, this feature is utilized on *analogWrite( )* function that is also used to control DC motors. However, the generated signal has a fixed frequency that may not be suitable for a particular motor, which may require a different signal frequency. Then, the user needs to create its own baseline time. This is not a problem if the user is familiarized with the TIMER's registers.

The analog-to-digital converter module is also affected by the fixed time base, because the sample rate is synchronized with the clock frequency. It is not an easy task, in Arduino, to reconfigure this parameter in order to get a specific sample rate frequency.

In general, any activity related to timing is affected by the selected clock frequency. The user needs to either reconfigure the functions or create its own procedures. At the same time, the user must validate that the changes do not affect the functionality of any external hardware connected to the board.

C. *System-level Related Problems*

When a project reaches the final stage of development, an engineer has to decide what hardware and software will be used. Any extra hardware or software may be considered a waste of resources. Therefore, it is likely that only the microcontroller, the sensors, and actuators will be used. There is no need for the Arduino's board or extra shields. This means that it would be necessary to separate from the Arduino's environment, work directly with the hardware pieces, and maybe use other specialized software tools. The transition from Arduino to other tools could be a difficult task for a student, because there is a chance that most of Arduino' functions won't be available. Then, students will have to learn what a microcontroller is and how to use it. In addition, they may need to create their own version of Arduino's functions used in the code. This may become a stressing situation for developers, and delay their work.

D. *Pedagogical Issues*

It is said that the use of the Arduino environment increased the success rate for course projects. A question that may be asked when a student submits a project done using the Arduino's environment is, how much of this project is really the student's work? The large amount of already-done examples and projects increases the chances that students find a complete solution for their project. Then, what were the student's contributions? What did they learn?

It is also possible that students find all the required parts for their project ready, so their job now is to integrate these parts to generate a solution for their design problem. Although some people may say that this is code reuse, which is a valid programming practice, the problem is that other learning aspects might be weakened, like innovation or critical thinking.

Is the use of the Arduino board appropriate for a particular course? The course objectives are the guideline for selecting the most suitable tool that can fulfill them. Embedded systems and similar courses, which are junior or senior-level courses, require not only to build functional systems, but to gain a deeper knowledge to find optimal solutions.

IV. PROPOSED APPROACH

This section shows the proposed approach to address the issues presented in previous section. First, we present what is expected from a Microcontroller's course. Second, we show the course methodology taken in order to meet the expectations for the course and to overcome the problems posed on the previous section. Finally, we show how our approach is aligned with ABET and IET requirements.

A. *Microcontroller's Course Goals*

The main purpose of the Microcontrollers course at Universidad Tecnológica de Bolívar is to expose students to concepts related with embedded systems such as design, hardware-software co-design, prototyping, optimization, validation, and testing. Due to time limitations, more specialized topics like real-time operating systems are left for other courses.

The course has a project-based learning approach. Students are exposed to problems with increasing difficulty through the course. From each problem, it is expected that the student develops skills to identify and learn how to use new hardware devices. Students have to create and debug code for their hardware interfaces and to create a whole system that solves the problem at hand. In addition, it is expected that students build their own component library, so they can port it to any microcontroller brand with reduced difficulty.

B. Course Methodology

The proposed coursework is divided into three main modules. The first module introduces the student to the general idea about microcontrollers and what can be done with them. Here, the Arduino board is used as the main platform. It is combined with different hardware boards (shields) in order to develop several applications. The focus is to use the boards as black boxes and available libraries to perform the actions required by the applications.

The second module goes beyond the Arduino board and focuses on the microcontroller. In this module, students learn about the microcontroller and what is inside the boards connected to it. Additionally, students learn how to create their own functions to control the microcontroller embedded hardware, and the boards attached to it. The Arduino environment is still used, but now the programming focuses on handling registers, hardware configuration, and protocols. The usage of Arduino's libraries is reduced.

The last module separates the students from the Arduino's environment. The AVR studio [7] environment is used. However, other microcontroller brands, such as MicroChip (PIC) and Texas Instrument (Tiva C) are encouraged. The student will use the available tools from the selected brand for creating and debugging their projects. The Arduino board can be used, but now only as a programmer, for those who decided to keep working with the ATmega microcontroller. Students learn how to create and optimize their own functions and applications on their selected environment. They also go through a full design cycle, from problem formulation to the final solution, including prototyping.

The course goes from a general view (provided by Arduino) to a deeper and specialized view. Students go from applications that use devices and code in a black-box fashion, to optimized applications that use any hardware and software adapted to the application's needs. Finally, students go from a basic environment to a more specialized environment, where they are able to use several tools for program debugging and code optimization.

C. ABET and IET

The proposed coursework must comply with requirements for electrical and computer engineering. These requirements are synthesized on ABET and IET student outcomes.

1) ABET Outcomes: The following paragraphs present the student outcomes directly addressed by the proposed course.

(a) An ability to apply knowledge of mathematics, science and engineering. Laboratory sessions and class activities require direct application of mathematical, science, and engineering knowledge to generate a satisfactory solution.

(b) An ability to design and conduct experiments, as well as to analyze and interpret data. Laboratory sessions require students to design experiments to characterize sensors and other devices. Activities related to statistical data analysis, such as curve fitting, are essential part of these experiments.

(c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. All course activities include design constrains, such as clock frequency, number of input-outputs allowed, and hardware availability. Software is also constrained by the limited memory on the micro- controller. Advanced constrains include: power consumption, execution time, and external issues, such as environmental temperature and humidity.

(d) An ability to function on multidisciplinary teams. Several course activities promote multidisciplinary team work. The course is taken by students from different engineering majors such as Electrical, Electronics, and Mechatronics. In addition, the laboratory sessions and course project focus on different fields, such as audio signal, physical and chemical variables, robotic applications, and multimedia interfaces, where the students involve themselves on these topics to understand and create adequate solutions. All these activities require an active effort to reach an acceptable result.

(e) An ability to identify, formulate, and solve engineering problems. This is achieved by assigning a not very detailed problem statement to the students. Each team needs to analyze the problem to formulate a more detailed and constrained environment, and then generate a solution that includes hardware and software components. The hardware includes microcontrollers, external sensors, and actuators. Students are encouraged to design their software using flowcharts or pseudo-code. It is also required to describe a benchmark setup for functional tests and performance metrics.

(k) An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice. Students are exposed to modern hardware and software tools to create, optimize, implement, and test their designs. They also read up-to-date information about microcontrollers, hardware, and applications.

2) IET Learning Outcomes: The proposed course also satisfies the IET learning outcomes for electrical and computer engineering.

Underpinning science and mathematics learning outcome. This learning outcome is not directly satisfied by the proposed course because it is not a math course. However, it is expected that students taking this course use math and scientific principles as the foundation to generate creative solutions to a given problem.

Engineering analysis learning outcome. This course is a direct application of engineering principles to solve a design problem. The course project, as the main design problem, requires the students to devise a solution founded in science principles, to analyze and evaluate the performance of its solution at hardware and software level.

Design. This learning outcome is addressed along the whole course. The student is confronted with practical and real design problems. The student must understand each problem and design a solution that complies with several constrains.

Engineering practice. The use of microcontrollers allows for the development of whole systems for real-time applications. The course project, as the main application designed by the students, consists of a prototype, including the PCB and package. The idea is to expose the students to the whole fabrication process and make them aware of difficulties at this stage of the product fabrication. The report for this project follows IEEE publication guidelines. This is with the purpose that students familiarize with technical writing formats and style.

V. Course Design

The lecture-laboratory course presented in this paper, ECE1463 - Microcontroller, is a three-credit course required for all students pursuing four-and-a-half-year degrees in electronics (EE) and mechatronics (ME) engineering at Universidad Tecnológica de Bolívar, Cartagena, Colombia. Each term consists of 16 weeks of instruction. A three-credit course in this system typically consists of three 50-min lectures each week and expects six hours per week for self-study, reading and homework assignments.

The students usually take this course in their junior year after Digital Systems and Electronics I courses. The microcontroller's course aims to develop skills in the design and construction of applications based on embedded systems. Student should make use of several aspects that they have learned on previous courses such as electronic design concepts, programming, digital systems, and signal processing. Additionally, the course stimulates team work in the development of laboratories and a final project.

The course's learning outcomes are shown in Table I. Implementation of each learning outcome is discussed next. The first learning outcome is about using the Arduino IDE. The approach adopted to present the hardware and software components of the Arduino framework is similar to the one used by Margolis [8]. Students get familiar installing and creating sketches. Since this is an EE course, it is important that students incorporate sensors like as temperature, pressure, touch, accelerometers, RFID, motion, as well as wireless networking technology (Wi-Fi and ZigBee), to build their final course projects. In general, students start with an idea based on successful projects as shown in [9]. We call it seed project. Students are able to test working hardware and software. This motivates student to propose new ideas for projects based on their own experiences.

TABLE I
LEARNING OUTCOME FOR IETR1463 MICROCONTROLLER COURSE

	Learning Outcome
1	Know the features of available tools for designing embedded systems
2	Know the internal architecture of a microcontroller
3	Understand the procedure to program and develop applications based on microcontrollers
4	Know the available information sources for developing embedded systems

The second learning outcome is about knowing the internal architecture of a microcontroller. In the course, we study the details of the ATmega328 microcontroller following the methodology presented in [10]. However, students are encouraged to explore other architectures like Microchip or Texas Instrument. At this point, students have to provide a preliminary version of their final projects using their platform. They are allowed to take advantage of the large number of cheap sensor boards available.

The third learning outcome is about developing their personal programs (functions and libraries) and hardware boards (sensors and actuators) to meet the design requirements of their own projects. The main focus on this part of the course is the understanding of TIMER module and the sleep modes. The first one is later used to adjust the base time of other modules like the analog-to-digital converter (ADC) and the programmable serial USART. The sleep modes are needed to reduce the system's power consumption. Final course projects are battery powered, so energy efficiency is encouraged.

During the whole course, students are in contact with the available information sources. This is the fourth learning outcome. The first-hand information is obtained directly from the microcontroller makers like as Atmel [11], Texas Instruments [12], and Microchip [13]. In addition, students are encouraged to get involved with e-communities like Arduino [14], Instructables [15], Energia [16], and Processing [17].

The laboratory schedule for the course is shown in Table II. As the Table shows, six weeks are completely dedicated to the course project, two weeks each module. First module (weeks 5 and 6) is for presenting the seed project and the final project proposal. Students select a working project, which they modify, improve, and show an upgraded version of it. Second module (weeks 10 and 11) is the opportunity to show a progress report about the project and readjust the final requirements if needed. Last module (weeks 15 and 16) is for presenting the results of their projects. Students are encouraged to highlight their difficulties as well as how they overcame them.

Final course projects are intended to use physical sensors, actuators, and networking. Projects require persistent storage and appropriate use of internal resources. Prototyping implies physical building that requires skills like PCB design, soldering, and testing. In addition, the completed system (software, hardware, and networking) had to be documented. Finally, in order to encourage peer-to-peer learning, the students work in teams of two to four members.

TABLE II
LABORATORY SCHEDULE FOR IETR1463 MICROCONTROLLER COURSE

Week	Laboratory Activity
1	Getting Started
2	Serial Communication Subsystem
3	ADC and DAC
4	Interrupt Subsystem
5	Demo *Seed Project*
6	Final Project Proposal
7	Timing Subsystem
8	Atmel AVR Operating Parameter and Interfacing
9	Visual Output
10	Mid Term Project Presentation
11	Project Requirement
12	Physical Output
13	Using Display
14	Audio Output
15	Final Report Presentation
16	Final Report

The course is divided into three learning modules. For each module, students are graded on weekly lab assignments, weekly discussion participation, and weekly quiz. At the end of the module, the students are given a comprehensive final exam. Table III shows the grading scheme for the course. All assessment instruments were individually administered except for the final class project that was assigned a single grade for a team of 2-4 students.

TABLE III
GRADING SCHEME FOR IETR1463 MICROCONTROLLER COURSE

Assessment	1-module	2-module	3-module
Exam	50%	35%	15%
Lab and quizzes	35%	30%	35%
Project	15%	35%	50%

Table IV shows sample projects for the last years the course has been taught. Before 2012, projects were focused on control, and later in 2013, we started with a project-based learning approach when an underwater remotely operated vehicle (UROV) was the topic for two consecutive terms. Since 2014, all the projects involve building physical artifacts that include sensors and microcontrollers to communicate with a computer using wireless and wired networking protocols. In many instances, students also used Processing [17], an electronic sketchbook for developing visual interfaces. Some course projects evolved as capstone projects, and have been presented in international conferences [18], [19].

VI. EVALUATION

Each semester the course was evaluated based on five aspects: mission, pedagogical model, disciplinary aspects, evaluation, and professor. Mission aspect is focused on the professional skills the student must gain in the course. Pedagogical model refers to how the activities developed during the course help them to gain the skill they need. Disciplinary aspects intend to measure the interaction between the contents of the course and their applicability on other areas or subjects. Evaluation is the aspect where students agree on the evaluation process. The final aspect, professor, is about how the professor helps students during the course in the learning process. Students are asked to evaluate how close the course meets the requirements on each aspect. Figure 1 shows the average percentage given by students on each aspect in the last five terms. As Figure 1 shows, the general perception is that the course is interesting, and it has a major impact due to the involvement of several disciplines to solve practical problems. This is reflected on an average satisfaction level over 80%. However, it is noticed a declining trend in the satisfaction level, which reach its lower value on the 2014 term, around the 75%. At this point, it was introduced the approach presented on this paper, and it changed the tendency, reaching the highest satisfaction value, over 85%, in just one term, although the projects became more challenging and required more student involvement with new hardware devices and tools.

TABLE IV
A SAMPLE OF PREVIOUS PROJECT

Year	Project Name
2015	miUniversidad pre-paid e-Card
	Tweeter Alarm
2014	Multi-parameter platform for controlling greenhouses to ensure optimal plant growth
	Self parking system
	A power meter solution [18]
	Proximity glasses for visual impaired
	Medicament dispenser system
	Irrigation system
	System for measurement of moisture in-room
	An accelerometer-based system [19]
2013	Underwater remotely operated vehicle - UROV
2012	Control of filling a tank
	Line follower robot
	Temperature-controlled fan

VII. DISCUSSION

Using open-source tool-chains worked well because the student had ample support materials available on the Internet. Another advantage is that very little laboratory space is required because students use external "shields" that they buy. A full lab with multi-meters, oscilloscopes, soldering stations, tools, wires, and other materials is provided to the students as needed. The students were also encouraged to use the facilities in the manufacturing laboratory.

Most course projects successfully combined hardware and software co-design methodology in a logical manner

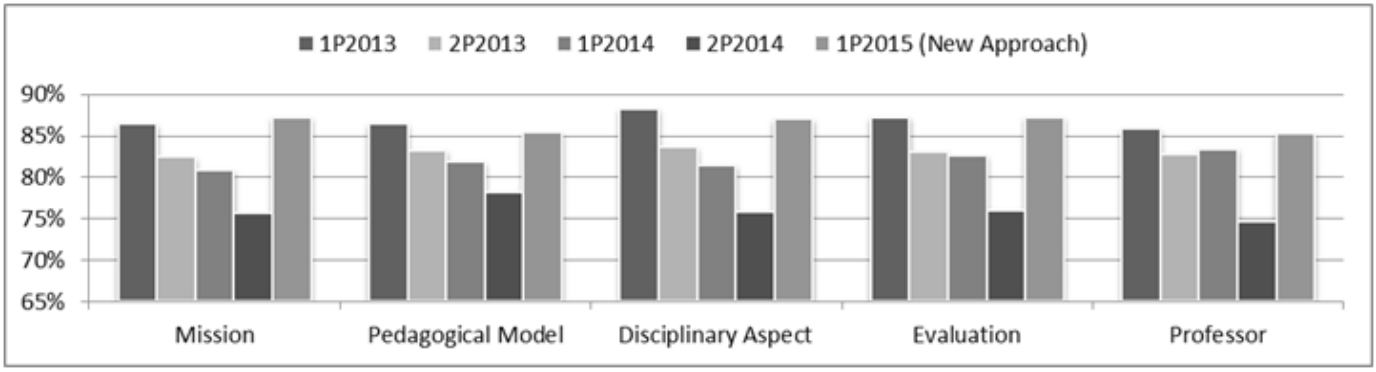

Fig. 1 Student evaluation for IETR1463 Microcontroller Course.

producing good designs. This is evidenced from the rubrics used to grade the final course projects.

Fig. 1 Student evaluation of Microcontroller Course.

One interesting side-effect of the student taking the course is that many students have started using the open-source hardware and sensors introduced in the course in their capstone projects.

VIII. Conclusion

This paper presented a project-oriented microcontroller course for electrical and computer engineering students. The course takes advantage of the Arduino's platform to have a rapid introduction to the microcontrollers and then performs a detailed study of the device for the development of applications to solve design problems under constraints. Overall, the course methodology has been well received by the students, and the results show that the course has helped them to improve their design skills. By going from a general knowledge of the Arduino platform to a deeper knowledge of the microcontroller architecture and use, the student gets a better understanding of the device and is not constrained to the hardware and software tools that are available in Arduino's community. Additionally, it is easier for the instructor to assess the student's contribution to the project development.

Acknowledgment

We would like to thank the anonymous reviewers for their comments and feedback on the ideas in this paper.

References

[1] I. Zualkernan, "A course for teaching integrated system design to computer engineering students," in *Global Engineering Education Conference (EDUCON), 2014 IEEE*, April 2014, pp. 470–474.

[2] K. G. Ricks, W. A. Stapleton, and D. J. Jackson, "An embedded systems course and course sequence," in *Proceedings of the 2005 Workshop on Computer Architecture Education: Held in Conjunction with the 32Nd International Symposium on Computer Architecture*, ser. WCAE '05. New York, NY, USA: ACM, 2005. [Online]. Available: http://doi.acm.org/10.1145/1275604.1275617

[3] P. Jamieson, "Arduino for teaching embedded systems. are computer scientists and engineering educators missing the boat?" in *Proc. FECS*, 2010, 2010, pp. 289–294.

[4] R. Chancharoen, A. Sripakagorn, and K. Maneeratana, "An Arduino kit for learning mechatronics and its scalability in semester projects," in *Teaching, Assessment and Learning (TALE), 2014 International Conference on*, Dec 2014, pp. 505–510.

[5] M. Del Carmen Curras-Francos, J. Diz-Bugarin, J. Garcia-Vila, and A. Orte-Caballero, "Cooperative development of an arduino-compatible building automation system for the practical teaching of electronics," *Tecnologias del Aprendizaje, IEEE Revista Iberoamericana de*, vol. 9, no. 3, pp. 91–97, Aug 2014.

[6] J. Sarik and I. Kymissis, "Lab kits using the arduino prototyping platform," in *Frontiers in Education Conference (FIE), 2010 IEEE*, Oct 2010, pp. T3C–1–T3C–5.

[7] A. Corporation. (2015) Atmel studio 6. [Online]. Available: http://www.atmel.com/microsite/atmel studio6/

[8] M. Margolis, Arduino cookbook. " O'Reilly Media, Inc.", 2011.

[9] J. Boxall, *Arduino Workshop: A Hands-On Introduction with 65 Projects*. San Francisco, CA, USA: No Starch Press, 2013.

[10] S. F. Barrett, *Arduino Microcontroller Processing for Everyone!* Morgan and Claypool Publishers, 2010.

[11] A. Corporation. (2015, Jun.) Atmel Corporation - Microcontrollers, 32 bits, and touch solutions. [Online]. Available: http://www.atmel.com/

[12] T. Instruments. (2015, Jun.) Analog, Embedded Processing, Semiconductor Company, Texas Instruments - TI.com. [Online]. Available: http://www.ti.com/

[13] Microchip. (2015, Jun.) Microchip Technology Inc. [Online]. Available: http://www.microchip.com/

[14] Arduino. (2015, Jun.) Arduino - Home. [Online]. Available: https://www.arduino.cc/

[15] Instructables. (2015, Jun.) Instructable - DYI How to Make Instructions. [Online]. Available: http://www.instructables.com/

[16] Energia. (2015, Jun.) Energia. [Online]. Available: http://energia.nu/

[17] Processing. (2015, Jun.) Processing. [Online]. Available: https://processing.org/

[18] M. Maryori Sabalza, J. Borre, and J. Martinez Santos, "Design and construction of a power meter to optimize usage of the electric power," in *Engineering Mechatronics and Automation (CIIMA), 2014 III International Congress of,* Oct 2014, pp. 1–5.

[19] R. Diaz Parada and J. Martinez Santos, "Study of the lower limp's angle during weightlifting exercises using an accelerometer-based system," in *Engineering Mechatronics and Automation (CIIMA), 2014 III International Congress of,* Oct 2014, pp. 1–4.