Preparation of Engineering Students for Capstone Design Experience through a Microprocessors Course

https://doi.org/10.3991/ijep.v7i4.6787

Mohammed El-Abd
American University of Kuwait
melabd@auk.edu.kw

Abstract—This paper presents the outcomes of a developed methodology to handle the project component in a higher-level undergraduate course. The approach relies on providing the students the freedom to choose their own project area as well as the utilized technology. At the same time, the students have to follow certain regulation to allow for the creation of a semi-capstone experience. We illustrate how this approach has a positive effect, not only on the project outcomes at the course level, but also on the students' performances in subsequent capstone courses. Data collected, over five consecutive course offerings, shows that this approach is an effective method to prepare engineering students for their senior design capstone courses.

Keywords—capstone project; course project; engineering education; student preparation; design experience; microprocessors course; microcontrollers course.

1 Introduction

In order to successfully prepare engineering students for real-life practice, ABET criterion 5 [1] explicitly states that the students must go through a "*curriculum culminating in a major design experience*". This major design experience is realized by most engineering programs world-wide through one or two senior design capstone courses that usually run for an entire year. In these courses, engineering students should apply all theoretical knowledge and practical skills gained through the entire engineering curriculum to tackle a real-life engineering challenge. Multiple previous studies [2, 3, 4] have asserted the extreme importance of such course.

Moreover, senior design capstone courses play a major role in accreditation processes as they can be used to directly measure all ABET learning outcomes [1]. However, the main issue to be addressed is that this is the student's first experience in meeting a challenge of this magnitude. To guarantee a successful capstone experience, students have to be scientifically and mentally prepared. Hence; an important research question, which we address in this work, is how to effectively prepare undergraduate students for a successful capstone experience?

One can argue that the scientific part of the preparation could be fully covered through the curriculum. We believe that one promising approach, which we present and test in this work, for mentally preparing students for their capstone experience is to provide them with a similar experience on a smaller scale.

The mental preparation of students involves developing a complete understanding of the entire design process as well as the evaluation criteria. In other words, students should know what to expect before embarking on such a challenge. The full design process includes many steps: identifying a practical engineering problem, modeling the problem identified, surveying relevant literature, producing design specifications, implementing the problem solution, analyzing the performance, and presenting and defending the developed solution.

The project component in undergraduate courses presents a natural venue to facilitate this preparation. Course projects can be used to:

- Teach students many of the previously mentioned steps on a smaller scale,
- Correct wrong perceptions students might have about capstone projects, and
- Address any problems that are expected to occur.

In order to have a similar experience on a "smaller scale", the students should be guided through the design cycle in order to solve a simple problem in less amount of time compared to capstone projects.

A clear example of points 2 and 3 above is that students might believe that a capstone supervisor should provide the solution for any faced problems. It was reported in [4] that faculty members understand the role of a capstone supervisor to be more of a "guide" through the entire process rather than an "instructor". The study showed that the least important responsibility of a supervisor is to tell the students what to do or how to solve a problem. Another example is to deal with the "free-rider" situation and to fairly measure the student's progress [5, 6].

The current study presents a methodology to handle the course project component in order to successfully prepare students for their capstone experience. The approach is applied over five consecutive course offerings and results are analyzed based on the selected technology, and the performance of students.

The rest of the paper is divided as follows: Section 2 introduces the methodology. Results are presented in Section 3. Finally, the paper is concluded in Section 4.

2 Methodology

Based on our classification in [7], there are four main categories, shown in Figure 1, on how to handle the project component. Approaches are classified based on two dimensions: the *problem* to solve, and the *technology* to adopt.

The *Restricted* approach offers the entire class a single application and/or problem, which all students must attempt to solve using a specific technology. From the instructor's point of view, this approach provides easier marking on the since all student groups would go through the exact same steps and could be easily compared against one another. Furthermore, it is ensured that all students are faced with the exact same set of challenges. We believe this method is more suitable for lower-level courses (e.g., electric circuits, instrumentation, and digital logic) where students' design abilities are still limited and the focus is on providing students with specific scientific knowledge.

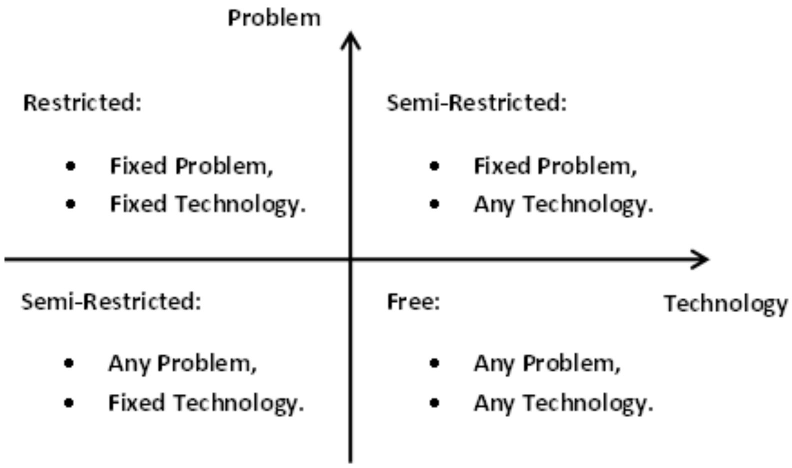

Fig. 1. Project handling classification.

The *Semi-restricted* approach offers the students the freedom of choosing the application and/or problem they desire to work on. However, all students must use a specific technology. Such an approach can work for both low-level and high-level courses. For example, in a computer architecture course, this could involve implementing different parts of the machine using VHDL. Another example, in a microprocessors course, is to implement different simple embedded system applications using different microcontrollers. The use of a fixed technology across all groups ensures that they will experience similar challenges related to, for example, the limitations and/or available features of the VHDL or the selected microcontroller. On the other hand, it provides the students with an opportunity to follow different problems according to their interests. Such an approach can cover more steps of the engineering design process.

The other *Semi-restricted* approach, in which the technology used is left to the students' discretion, offers the opportunity of evaluating the student's ability to properly select a technology. At the end of the course, students could be presented with data comparing different adopted technologies for the same problem. However, once the selection is done, the instructor might guide the students to use a more suited technology in order to realize the best projects possible. Hence, it will eventually become similar to the previous method.

The *Free* approach is based on offering the students the complete freedom of selecting the application and/or problem they want to work on as well as the technology they want to adopt. On the instructor's side, this requires the development of some sort of a rubric to be able to evaluate the wide range of developed projects and acquired technological skills. Across the different projects, students will be facing different challenges depending on their chosen application and selected technology. In addition, the course instructor can evaluate the students' abilities to compare different technologies/designs and to choose the most appropriate for their selected application. We believe that such an approach is more suitable for higher-level courses (e.g., microprocessors, embedded systems, and DSP). We also believe that such an approach will provide the closest experience similar to a senior design capstone course. This is because not only students will be involved in implementing a stand-alone application and/or system as in the capstone; they can also be graded based on a similar rubric. This is different from internship courses where the final grade in many situations is just a pass or fail based on a number of submitted reports.

In the *Semi-restricted* or *Free* methods, the students work on a problem that they deem interesting. This increase the students' own satisfaction and improve their overall learning experience. Another side benefit is that some students can opt to further extend their course project in their capstone.

We adopted the *Free* approach and applied it in a higher-level "Microprocessors and Interfacing" course. In our recently ABET-accredited engineering program, this course is a three credit hours third-level core course that serves as a pre-requisite for the capstone project. Moreover, the course is accompanied with a one credit hour laboratory component. The learning outcomes of both courses include understanding the internal architecture of the PIC microcontroller, writing assembly programs utilizing different addressing modes and performing input/output operations, mastering different interfacing techniques, using available microcontroller features (e.g. interrupts, and timers), and covering multiple subjects including data conversion and communication. The decision to teach the PIC microcontroller is based on the fact that it has a big share in the embedded systems market [8].

The rationale behind selecting such a course for our study is a three-fold:

- It is an upper-level course allowing us to adopt the *Free* approach,
- It is a direct pre-requisite to the senior design capstone course. Thus, will provide the students with a very recent experience before facing a project of a larger scale, and
- It presents the students with knowledge about a ubiquitous technology that is seen everywhere and that is most probably used by students in their capstone projects.

To ensure the quality of the delivered course projects and to fully simulate a capstone experience, several steps are required:

- Students are requested to secure the instructor's approval of their idea. The instructor examines the proposed project ideas to ensure an appropriate level of difficulty and to manage students' expectations,
- Students follow specific guidelines for writing the final report and presenting projects in-class,
- Projects are evaluated following the same rubric used for capstone courses [9].

3 Results

The findings of our proposed methodology, based on five consecutive semesters, are presented in this section.

3.1 Creative Course Projects

As pointed out in the previous section, adopting the *Free* approach presents the students with the opportunity of implementing project ideas they are most interested in. Another benefit is that adopting this method ensures the realization of a wide range of projects spanning different interesting areas.

Previous work in [10] reported that using Arduino [11] resulted in the students implementing very creative projects. The same conclusion is emphasized by our study as the students implemented many complete stand-alone embedded systems covering a wide range of applications.

Many students implemented robotics applications including the maze-solving robot and the pixel-wise drawing robot shown in Figure 2.

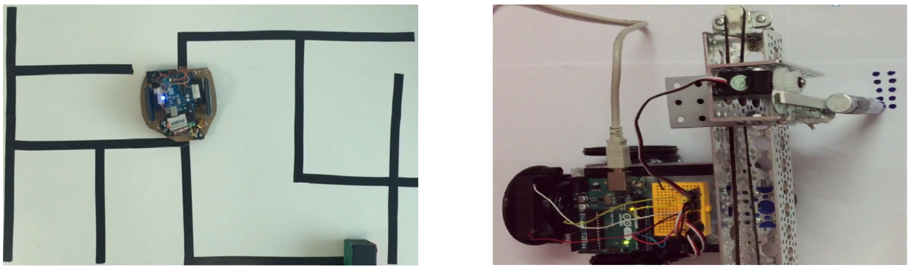

Fig. 2. A maze solving robot and a pixel-wise drawing robot.

Other students implemented gaming applications including the Fluffies game and the complete Arduino-based arcade game shown in Figure 3, and the Tic-Tac-Toe and the ping-pong games shown in Figure 4. Other stand-alone applications included the portable mp3 player shown in Figure 5.

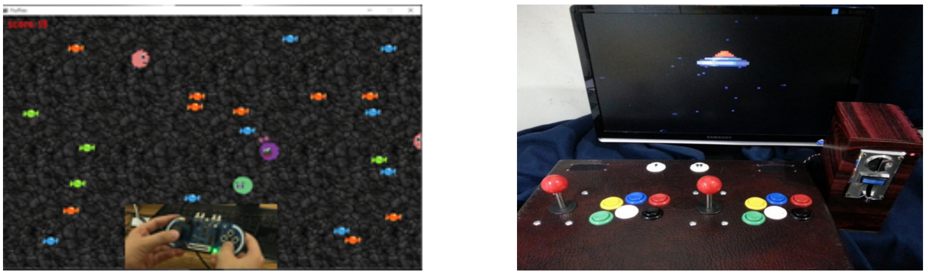

Fig. 3. A game of fluffies and Arcduino (An Arduino-based arcade game).

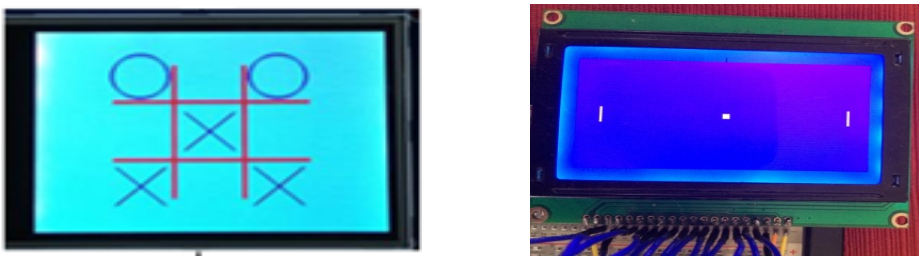

Fig. 4. A Tic-Tac-Toe game touch screen interface and a simple ping-pong game.

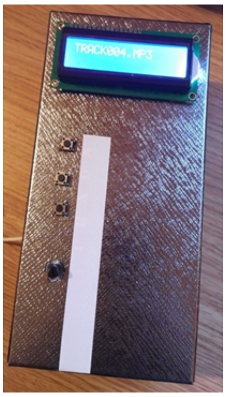

Fig. 5. Arduino -based portable mp3 player.

Moreover, the use of Arduino enables the students of exploring other directions and different programming approaches. For example, Fig. 10 shows a traffic-light control model programmed using Simulink and implemented using Arduino.

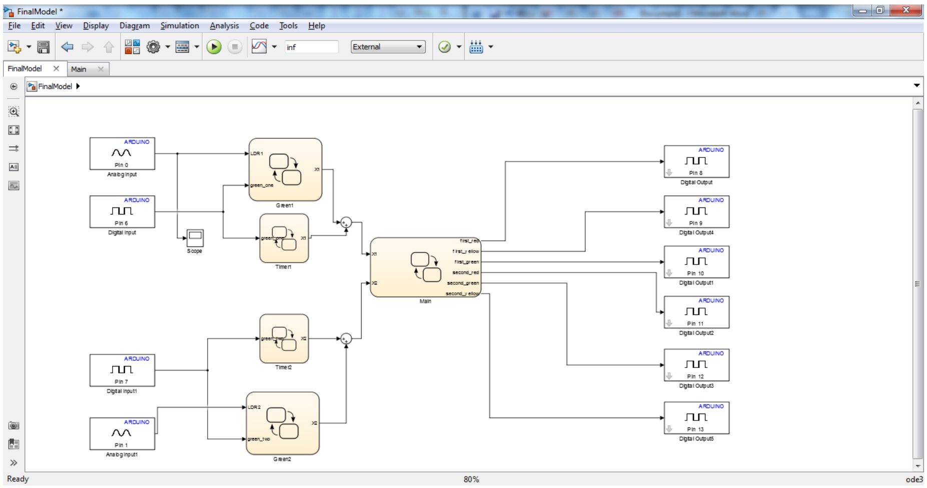

Fig. 6. Arduino-based traffic lights implementation using Simulink.

When enrolled in embedded systems courses, work in [12] used Gamification to motivate computer science students. An interesting observation in the current work is that 14 engineering students chose to explore the area of game implementation in their projects. This constitutes around 10% of the entire student population covered in this study.

3.2 Effect on Capstone Projects

The number of students enrolled in the "Microprocessors and Interfacing" course, when offered over five consecutive semesters, is presented in Tables 1 and 2. Moreover, the technology used by students in their course and capstone projects is also reported. Resulted presented here are an extension on our previous work in [7].

Table 1. Breakdown of used technology in the course project.

Semester	Number of Students	Adopted Technology	
		Arduino	***PIC***
Fall 2013	51	51	0
Fall 2014	33	29	4
Spring 2015	16	14	2
Fall 2015	34	34	0
Spring 2016	25	25	0

Table 2. Breakdown of used technology in the capstone project.

Semester	Number of Students	Adopted Technology		Percentage
		Arduino	***Other***	
Fall 2013	51	32	12 (7)	73%
Fall 2014	29	18	11	59%
Spring 2015	14	10	4	71%
Fall 2015	34	30	4	88%
Spring 2016	25	16	6 (3)	73%

We would like to note here that the students are requested to implement their projects using a different PIC microcontroller than the one studied in the course or a different microcontroller altogether. This decision is taken to improve their life-long learning abilities and to make sure they are exposed to more software and hardware tools beyond what's being covered in class.

Results presented in Table 1 clearly illustrate that 96% of students opted to use the Arduino boards in their course projects. Arduino is the popular choice for its low cost, adoption of the open-source philosophy, availability of online resources, and ease of expandability. Similar behaviors were also identified in [10].

Results presented in Table 2 illustrate that the percentage of students that continue to use the same technology in their capstone projects is at least around 60%. The percentages provided are calculated after excluding purely software capstone projects (presented in parenthesis). When implementing hardware-based capstone projects, other popular technologies include data-acquisition systems like Phidgets [13] and LabVIEW [14].

In conclusion, the overall percentage of students pursuing the same technology over the five consecutive semesters is 69.28%. Although similar trends were also highlighted in [15], to the best of our knowledge, the current work represents the first conducted study to assess the extent and the effect of such a behavior over a number of semesters.

To assess the effect of using the same technology on the students' performance, Table 3 presents the average grades for students using the same technology in their course and capstone projects. *Number of students* is provided for students who completed the two capstone courses. Other students are still in their first or second capstone course and hence their product has not been yet finalized.

In the first two semesters, results show that average grades slightly increase reflecting an improved performance. In later semesters, performance of students in capstone projects has considerably improved showing an increase between 3% and 6% in their final grades. The significance of these results stems from the fact that evaluating capstone projects is more scrutinizing than evaluating course projects and that the final product is of much larger scale.

Table 3. Average grades for the course and capstone projects.

Semester	Number of Students	Average Grades	
		Course	*Capstone*
Fall 2013	32	88.09%	88.66%
Fall 2014	18	87.87%	87.89%
Spring 2015	10	81.6%	86.5%
Fall 2015	27	86.37%	89.63%
Spring 2016	9	82.77%	89.44%

Table 4 presents how the performance of individual students using the same technology has changed from the course project to the capstone project. *Better* means that the student has achieved a higher letter grade, *Same* means the same letter grade was maintained, while *Worse* means that the student has achieved a lower letter grade in the capstone compared to the course project component. Again, *Number of students* is provided for students who completed the two capstone courses.

Results show that around 54.17% of the students have demonstrated better performances in their capstone projects. Note that for the ten students having the same performance from the Fall 2013 class, six of them have actually remained in the A-/A region, which is an excellent performance already. In addition, two of the seven students who had a worse performance have only slightly dropped from A to A-. Moreover, out of the four students having a similar performance in Fall 2014, two of them remained in A-/A region. In addition, one of the six students with a worse performance has slightly dropped from A to A-. For Fall 2015, two students of the four having a similar performance remained in the A region, while three of the eight students having a worse performance have slightly dropped for A- to B+. Overall, 52 students out of 96 (54.17%) have an improved letter grade in their capstone projects. On the other hand, only 16 students (16.8%) have a considerably worse (at least a two grade letters drop) performance.

Table 4. Performance comparison for course and capstone projects.

Semester	Number of Students	Performance		
		Better	*Same*	*Worse*
Fall 2013	32	15	10	7
Fall 2014	18	8	4	6
Spring 2015	10	7	1	2
Fall 2015	27	15	4	8
Spring 2016	9	7	2	0

This overall improved performance of students in their capstone course is partially due to the reduced learning time needed for new technology. It was stated in [16] that using different hardware platforms and laboratory tools across different courses can introduce around 30% overhead in both time and effort in order to learn the new tools. Hence, as students are already using a technology they are familiar with, they can advance in their capstone course at a faster pace.

Although the adopted methodology shows real promise, there are a number of students that still show a deteriorated performance. There are a number of factors that could have contributed to this observation. These include the lack of time management skills and the tendency to avoid important software engineering practices [17], the lack of team work skills, the lack of study skills [18, 19], and the difficulty of the project topic with respect to the student's academic level.

4 Conclusions

In this paper, we posed the question of how we can properly prepare students for their capstone experience. We believe that one effective approach to facilitate this preparation is by the appropriate handling of the project component in a carefully selected course (or set of courses).

The work presented different methods for managing course project component. The study adopted the *Free* approach in which the students are free to select their project topic as well as the adopted technology. We applied the selected approach to a "Microprocessor and Interfacing" course. This course is a higher-level core course that is directly followed by the capstone course, and covers popular technology. The followed methodology resulted in a wide range of very interesting and creative projects.

The study was conducted over five consecutive semesters. Results showed that for a total of 153 students, around 69% of them continued to adopt the same technology used in their course projects to implement their capstone projects.

Moreover, results showed that 54% of the students adopting the same technology have demonstrated a better performance in their final capstone projects in comparison to their course projects.

In future work, we intend to extend our results to cover more microcontroller and/or microprocessor platform (Raspberry Pi, Beagle Bone, … etc.). We also plan to study the effect of different identified factors on the overall performance. Furthermore, we will initiate a questionnaire to collect graduating students' experiences regarding the presented methodology.

5 References

[1] http://www.abet.org/wp-content/uploads/2015/05/E001-15-16-EAC-Criteria-03-10-15.pdf

[2] Todd, R. H., Magleby, S. P., Sorensen, C. D., Swan, B. R. and Anthony,D. K. (1995). A survey of capstone engineering courses in North America. Journal of Engineering Education, 84: 165–174 https://doi.org/10.1002/j.2168-9830.1995.tb00163.x

[3] McKenzie, L.J., Trevisan, M. S., Davis, D. C., and Beyerlein, S. W. (2004). Capstone design courses and assessment: a national study. Proceedings of ASEE Annual Conference & Exposition, 9.286.1-9.286.18.

[4] Pembridge, J. and Paretti, M. (2010). The current state of capstone design pedagogy. Proceedings of ASEE Annual Conference & Exposition, 15.1217.1-15.1217.13.

[5] Kyte A. (2013). A 'Fresh Eyes' Look at Improving the Effectiveness of Engineering Group Design Projects. Engineering Education, 8(1): 81-97. https://doi.org/10.11120/ened.2013.00009

[6] Kadry S., El Hami A. (2016). A Novel Design of Management Senior Project for Engineering Students. International Journal of Information and Education Technology, 6(6): 504-507. https://doi.org/10.7763/IJIET.2016.V6.741

[7] El-Abd M. (2016). How course projects can successfully prepare engineering students for capstone design projects. Proceeding of the IEEE Global Engineering Education Conference, EDUCON, 746-750. https://doi.org/10.1109/EDUCON.2016.7474635

[8] http://www.microchip.com/pagehandler/en-us/press-release/microchip-reclaims-top-8-bit-m.html

[9] Yousafzai, J., Damaj, I., and El-Abd, M. (2015). A unified approach for assessing capstone design projects and student outcomes in computer engineering programs. Proceeding of the IEEE Global Engineering Education Conference, EDUCON, 333-339. https://doi.org/10.1109/EDUCON.2015.7095993

[10] Jamieson P. (2011). Arduino for Teaching Embedded Systems. Are Computer Scientists and Engineering Educators Missing the Boat? Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS).

[11] https://www.arduino.cc/

[12] Ristov S., Ackvoska N., and Kirandziska V. (2015). Positive Experience of the Project Gamification in the Microprocessors and Microcontrollers. Proceedings of the IEEE Global Engineering Education Conference, EDUCON, 511-517. https://doi.org/10.1109/EDUCON.2015.7096018

[13] http://www.phidgets.com/

[14] http://www.ni.com/labview/

[15] Patińo O. A., Conterars-Ortiz S., and Martínez-Santos J. C. (2016). Evolution of Microcontroller's Course under the Influence of Arduino. Proceedings of the 14th LACCEI In-

ternational Multi-Conference for Engineering, Education, and Technology: Engineering Innovations for Global Sustainability, 1-7.

[16] Ravel M., Chang M., McDermott M., Morrow M., Teslic N., Katona M., Bapat J. (2009). A Cross-curriculum Open Design Platform Approach to electronic and Computing Systems Education. Proceedings of the IEEE International Conference on Microelectronic Systems Education (MSE), 69-72.

[17] Koopman P., Choset H., Gandhi R., Krogh B., Marculescu D., Narasimhan P., Paul J. M., Rajkumar R., Siewiorek D., Smailagic A., Steenkiste P., Thomas D. E., and Wang C. (2005). Undergraduate Embedded System Education at Carnegie Mellon. ACM Transactions on Embedded Computing Systems, (4)3: 500-528. https://doi.org/10.1145/1086519.1086522

[18] Zimmerman B. J. (2010). Becoming a Self-regulated Learner: An Overview. Theory into Practice, (41)2: 64-70.

[19] Utesch M. C. (2016). A Successful Approach to Study Skills: Go4C´s Projects Strengthen Teamwork. International Journal of Engineering Pedagogy (iJEP), (6)1: 35-43. https://doi.org/10.3991/ijep.v6i1.5359

6 Author

Mohammed El-Abd is an Associate Professor of Computer Engineering in the ECE department at the American University of Kuwait (AUK). He is a senior member of IEEE. He has over 15 years of teaching experience with special interests in embedded systems, micro-processors, micro-controllers, computer architecture, robotics and artificial intelligence.

This article is a revised version of a paper presented at the EDUCON2017 conference held in Athens, Greece, 25-28 April 2017. Article submitted 20 February 2017. Published as resubmitted by the author 04 April 2017.