A Review of Embedded Systems Education in the Arduino Age: Lessons Learned and Future Directions

https://doi.org/10.3991/ijep.v7i2.6845

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

Abstract—In this paper, the subject of embedded systems education in the Arduino age is examined. Arduino is an open-source microcontroller platform that has been widely popular in the past decade among hobbyists and academics. Arduino is increasingly being adopted in courses that span different disciplines in schools and universities. As a result, numerous papers are being published every year in different engineering education conferences and journals reporting the integration of Arduino in teaching. In this work, the impact of Arduino on embedded systems education is investigated. First, challenges facing embedded systems education are identified from the literature. Second, different Arduino teaching integration methodologies reported in the literature are surveyed and analyzed. Third, the question whether Arduino successfully addresses embedded education challenges or not is discussed taking both surveyed findings and recent market trends into consideration. Finally, a number of open-ended research directions are proposed.

Keywords—Arduino, engineering education, embedded systems, microcontrollers, open educational resources, open source hardware, open source software.

1 Introduction

Embedded systems in the engineering domain refer to systems that do not necessarily have a computational task; yet they are controlled by a computing entity. This computing entity could be a microprocessor, a microcontroller, a Field programmable Gate Array (FPGA), or a Digital Signal Processor (DSP). Nowadays, these are such ubiquitous technologies that are being used in more applications than anyone can imagine. Such applications range from household appliances and office equipment, home automation, consumer electronics, to the automotive industry and beyond.

The joint ACM/IEEE task force developed a 2016 draft model computer engineering curriculum[1] in which the knowledge area of embedded systems could be allocated up to 40 core hours. The embedded systems knowledge area covers many topics containing relevant tools, software techniques, input/output, serial communication, time measurement, and data acquisition. The number of hours allocated to this knowledge

[1] https://www.computer.org/web/peb/curricula

area in 2016 was doubled from 2004 as a reflection of directions in which the discipline has evolved.

The work in [1] identified seven different models for embedded systems education. They identified which models were being used by surveyed universities in North American, Europe, and the Far East. These models ranged from teaching individual embedded systems courses in undergraduate level, graduate level, to developing complete embedded systems programs.

Embedded computing is divided into a number of categories in [2]. One category, which is the focus of this work, is small and single microcontroller applications. The authors stated that this topic serves well as an introduction to other embedded courses. The authors motivated their students with the use of exciting course projects. One major challenge for this type of a course as identified by the authors is that students do not like "heavy" engineering processes and that they tend to skip important software engineering practices to meet deadlines.

In general, teaching embedded systems could be *hardware-oriented*, *software-oriented*, or *hardware-software integration* [3]. The authors presented how an "Introduction to Embedded Systems" course should fit in the engineering curriculum, a placement similar to what was proposed in [4]. Moreover, the authors suggested that in *hardware-oriented* teaching, students should be given the opportunity to select and/or buy their own development kits at the beginning. In addition, the authors identified embedded engineering education challenges to be *student-related* (lack of knowledge, lack of motivation, planning skills …), *lecture-related* (spanning of several fields, dynamic progress of technology, hardware and software compatibility …), and *course-content related* (limited-time, a discipline not well-defined)

A research study was performed in [5] to investigate how to transform academic teaching to better equip students with design skills and fulfill industry needs. An interesting finding was that both academics and professionals agreed that the students' lack of motivation is due to the teaching style. In addition, results clearly showed that students are more interested to learn when presented with hands-on projects and practical applications. It was concluded that educators must focus on practice rather than theory in the classroom.

A challenge facing the industry and academia collaboration was highlighted in [6]. The authors pointed that nowadays the technology cycle is shorter than the engineering education time, which means that what the industry needs now, should have been already provided by education in the past. This is the same *lecture-related* challenge (dynamic progress of technology) highlighted in [3]. The authors pointed out that one solution could be having early engineering education in basic and high schools [7]. Several other challenges identified by the authors included: the incorporation of such concepts as system integration, testing, and versification, the used learning platform, the adopted educational methodology (classes, laboratories, ...) and the followed evaluation and assessment scheme.

In addition, the same authors developed their own embedded engineering learning platform (E2LP) [6, 8]. The platform was an attempt to address the issues previously raised while supporting 5 learning objectives covering embedded microprocessors, DSP, FPGAs, networks, and system integration. The authors use a single hardware platform through all their courses to considerably reduce the learning time at the beginning of each course. Again, this also addresses one of the *lecture-related* challenges (hardware and software compatibility) highlighted in [3].

The idea to develop a single platform for all courses (E2LP) was followed to overcome difficulties with laboratory classes highlighted in [9]. These difficulties included the time needed to learn new tools, the time needed to acquire and/or fabricate new parts, the support needed for design tools, the little reuse of equipment across different courses, the time needed to manage projects. Authors pointed out 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.

Fig. 1 illustrates different embedded education challenges identified in the literature. The figure adopts the same classification in [3] while adding a new class and highlighting how similar challenges were identified in other works.

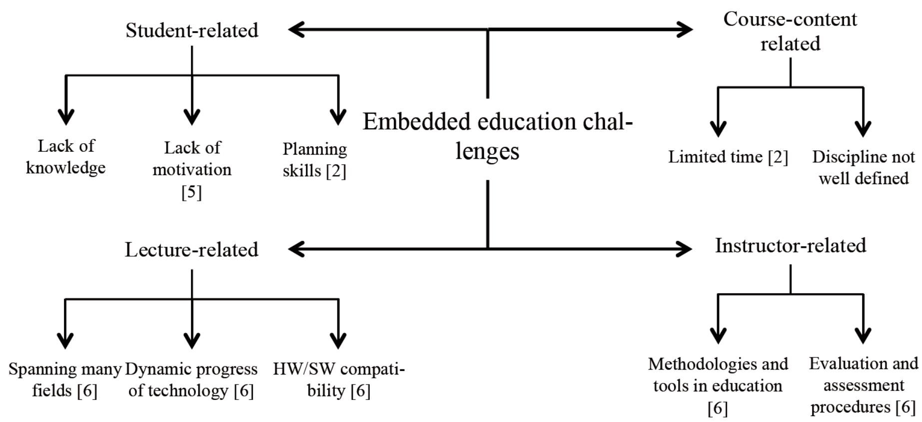

Fig. 1. Embedded education challenges.

The added class in Fig. 1, *instructor-related*, refers to decision that need to be taken by the instructor prior or during the course. This is based on two challenges identified in [6]. One challenge is the methodologies and tools for education, which highlights the need to carefully design the course structure and select the appropriate learning platform. Note that the platform selection could be also tied to the HW/SW compatibility challenge. The second challenge is the evaluation and assessment procedure adopted by the instructor in the course.

This review focuses on the use of Arduino in embedded systems courses. To the best of our knowledge, such an investigation has not been carried before in the literature. In specific, the goal of the study is to address the following questions:

- How was Arduino adopted in embedded education?
- Does the use of Arduino address the current challenges faced by embedded engineering education?
- What future research directions could be investigated?

The rest of the paper is divided as follows: Section 2 gives a brief background about Arduino. Previous works on the use of Arduino in embedded engineering education are analyzed in Section 3. Section 4 summarizes the lessons learned and proposes future research directions. Finally, the paper is concluded in Section 5.

2 The Arduino Platform

The Arduino project was first developed in Italy in 2005. As defined on their website[2] "Arduino is an open-source electronics platform based on easy-to-use hardware and software". The Arduino platforms were built to address several issues in other microcontroller products making it more appealing for hobbyists, students, and teachers. It has the advantages of low cost, cross-platform, simplicity of programming, and open-source extendable software and hardware.

Arduino products are based on the 8-bit ATmega microcontrollers. Boards are equipped with a large number of digital and analog IO pins, serial communication modules, USB connection, and ICSP capability. Arduino boards are easily interfaced with external components for data acquisition and control applications. All Arduino boards could be connected to a number of shields developed for different applications.

As a measure of the popularity of Arduino, Fig. 2 illustrates the results extracted by Google Trends regarding searches for three keywords, namely: Arduino, Raspberry Pi, and the PIC microcontroller. An interesting behavior to note is the general continuous increase of the number of searches for Arduino and Raspberry Pi with Arduino having a higher number. On the other hand, the number of searches for the PIC microcontroller is gradually decreasing over the past five years.

Moreover, as Arduino became more and more popular among academics, the number of publications involving the Arduino platform has considerably increased over the years. Fig. 3 illustrates the number of Arduino-related publications in the following engineering education conferences: American Society for Engineering Education (ASEE), Frontiers in Education (FIE), IEEE Teaching, Assessment, and Learning for Engineering (TALE), and IEEE Global Engineering Education Conference (EDUCON). These publications either report using Arduino in education (in primary schools, high schools and universities), introducing new Arduino educational boards, recommending its use, or just acknowledging the technology.

Some programs offer complete courses for learning Arduino. For example, two courses are dedicated for learning Arduino and its programming environment, and interfacing principles at the University of California, Irvine[3] in an Internet-of-Things (IoT) specialization.

Arduino is gaining interest from the industry as well. Referring to an embedded systems market study carried by UBM[4] in 2014, 19% of the surveyed professionals are considering the use of Arduino in their next embedded project.

[2] https://www.arduino.cc/en/Guide/Introduction
[3] https://www.coursera.org/specializations/iot
[4] http://bd.eduweb.hhs.nl/es/2014-embedded-market-study-then-now-whats-next.pdf

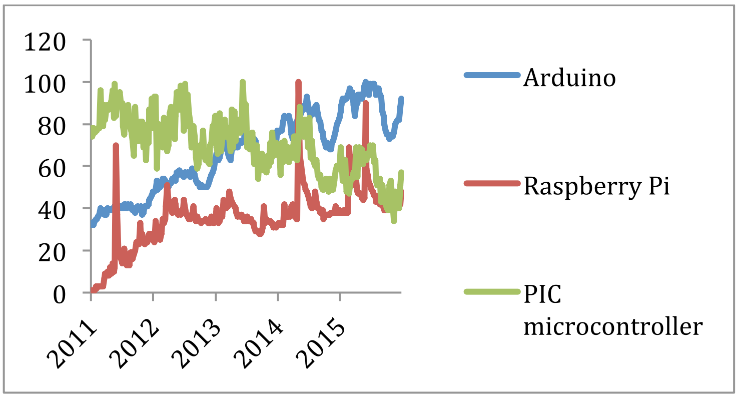

Fig. 2. Google trends extracted history of searches for the past 5 years.

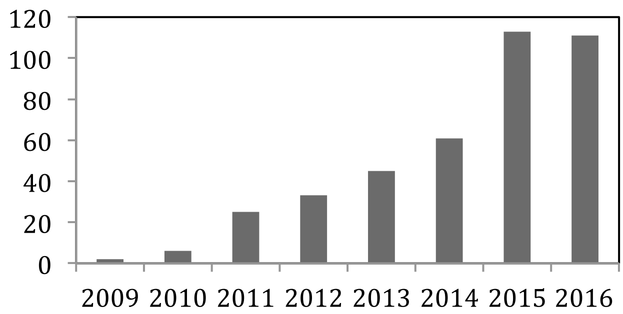

Fig. 3. Number of Arduino-related publications per year.

3 Arduino In Embedded Systems Education

This section analyzes previous works using Arduino in teaching embedded systems. Other works, which might not be cited, apply Arduino in freshmen engineering, adopt it in other disciplines, develop Arduino-based remote laboratories, implement Arduino-based educational kits, or use Arduino in pre-university education.

Table 1 provides a comparison among different Arduino integration methods in embedded systems courses. The comparison is based on three metrics: *Platform adoption* (Single vs. Multiple), *Project type* (Free vs. Restricted), *Programming knowledge* (Low-level vs. High-level vs. Both). These metrics are selected to reflect the level to which Arduino was injected into the course. A "Free" project type refers to students selecting their own project topics and/or technologies while "Restricted" means that these are enforced by the instructor [15]. In addition, Table 1 reports the course level, the assessment scheme followed by different works, and any reported advantages and/or concerns.

Table 1. Arduino integration methodologies comparison

Work	Course Level	Platform	Project	Program	Assessment	Advantages	Concerns
10	Junior	Arduino	Free	C	Reports + Presentations + Video Demos	Student's motivation; Accessibility	Timers inaccessible
11	Third year	Arduino + FPGA + PIC	Free	C	Presentations + Demo	Interesting varied and projects; Ease of interfacing external chips	Low-level concern; Student's contribution; missing topics
12	Third year	Arduino + Raspberry Pi + BeagleBone	Free	C	Reports + Presentations + Demo	Improved system design capabilities	NR
13	Senior	Arduino + FPGA	Free	C	Reports + Presentations + Demos	Ease of use; Less development time; Logistical issues	NR
14	Second year	Arduino	Free	C	Reports + Presentations + Demos	Increase in success rate	Low-level concern; Student's contribution
15	Third year	PIC + Arduino	Free	Assembly + C	Reports + Presentations + Demos	Interesting projects; Improved performance in capstone	Student's contribution
16	Junior	Arduino + Suggested	Free	C + Assembly	Reports + Presentations	Support material; little lab space;	Student's contribution; Hidden code; Clocking applications
17	Master	Arduino	Restricted	C	Reports + Presentations + Video Demos + Self- and Peer-assessment	Increased satisfaction; Increase in success rate	NR

Inspecting the first metric in Table 1 shows that Arduino was the *single* learning platform adopted throughout the course in [10, 14, 16, 17]. In [11], Arduino was compared against FPGAs and the PIC microcontroller used in previous course offerings. In [12], the Arduino, Raspberry Pi, and BeagleBone were used in the embedded systems and capstone courses. In [13], both Arduino and FPGAs are concurrently used in the course delivery. In [15], the main platform adopted was the PIC microcontroller, while Arduino was the popular platform selected by students in implementing their course projects. In [16], students were encouraged to use another microcontroller in the project but Arduino was still allowed. An interesting observation is that over 90% of students, when given the choice, select Arduino to implement their projects [11, 15].

For the course project metric, work in [17] enforced the project topic in home-management Systems. All other works provided the students with the freedom to select their project topic. This usually resulted in increased students' enthusiasm and motivation as well as more creative projects being developed. This could be a direct result of students working on projects they feel passionate about.

Projects based on integrating already available resources were accepted in [11] to improve system integration skills. In [12], same authors reported that the main microprocessors course learning objective is to "create". Hence; the assessment scheme was updated with a deliverable document distinguishing the students' contribution from used resources. Authors reported the improvement of the students' system design capabilities.

In most previous works, Arduino was programmed using its C-like language; thus only providing high-level knowledge. The single exception was for [16], in which Assembly was used during the final stage of the course using AVR studio[5] but not necessarily for Arduino. Assembly usage in [15] was for the PIC microcontroller during the course. However, Arduino was always programmed using C in the project. This still offers both low-level and high-level knowledge as in [16]. Both approaches address "the lack of low-level knowledge" concern raised in [11, 14].

As for the effect on students' performance, the percentage of students successfully passing the course in [14] has increased, after using Arduino, from 61% to 92% in module I and from 66% to 93% in module II. In [15], a study conducted over three consecutive semesters revealed that 94% of students selected Arduino for their course projects. Out of these students, 59% continued to use Arduino in their capstone courses resulting in an improved performance. In [17], the cooperative learning methodology, with the use of Arduino, resulted in improving the academic success of students with 93.5% scoring above 85%.

One cited advantage in [13] is the reduction in project development time allowing for post evaluation, and increasing code complexity. Same advantage was highlighted in [15], as using Arduino allowed students to develop fully functioning systems with the appropriate documentation in less time.

The concern highlighted in [10] about timers was also identified in [16]. The concern was addressed by changing the course delivery method in [16] to use AVR Studio towards the end of the course in order to program Arduino in low-level. An important advantage reported in [16] is the need for little lab space. As students can buy their own Arduino platforms at an affordable price, there becomes no need to purchase and install dedicated laboratory equipment for the course.

Finally, one major advantage identified in most of these works is the ease of learning and using the Arduino platform (Accessibility in [10], Ease of interfacing external chips in [11], Ease of use in [13], and Support material in [16]). With the availability of numerous online forums and groups, tutorials, and previously implemented projects, it becomes easier for students to learn such a platform in less time.

4 Lessons Learned and Future Directions

In this section, and based on the literature and recent market trends, the question whether Arduino addresses embedded education challenges is investigated. Moreover,

[5] http://www.atmel.com/Microsite/atmel-studio/

future research directions are identified to better assess the current state and to improve Arduino integration in teaching.

The Student-related Challenges: A commonly reported advantage, when using Arduino, is the increase in students' motivation and interest. This directly addresses the *student-related* challenges in [3]. This observation is also directly related to the conclusions reached and actions recommended in [5]. Such an advantage does not only help in improving students' performance in classes [14, 15], but it can also attract students to the engineering major. For example, switching the course technology to Arduino in [18] has resulted in minimizing the students' evasion rate from the computer engineering program.

Another *student-related* challenge is the student's planning skills, which was also emphasized in [2]. This is implicitly handled by using Arduino as developing a complete system with this platform can take less development time in general [13, 15]. This provides the students with more time for code maintenance, debugging, and documentation.

The Lecture-related (Spanning many Fields) Challenge: One challenge is the wide range of topics to be covered in embedded systems courses. These topics are divided across 13 units (12 core + 1 supplementary) under the embedded systems knowledge area in the 2016 ACM/IEEE model curriculum. Arduino could be used to cover most of these topics with varying depths. Units are presented in Table 2 with possible degrees of coverage.

Table 2. Arduino coverage of embedded systems units

Number	Knowledge Unit	Coverage
1	History and overview	Full
2	Relevant tools, standards, and/or engineering constraints	Full
3	Characteristics of embedded systems	Full
4	Basic SW techniques for embedded applications	Full
5	Parallel input and output	Full
6	Asynchronous and synchronous communication	Full
7	Periodic interrupts, waveform generation, time measurement	Medium
8	Data acquisition, control, sensors, actuators	Full
9	Implementation strategies for complex embedded systems	Low
10	Techniques for low power generation	Medium
11	Mobile and networked embedded systems	Full
12	Advanced input/output topics	Medium
13	Computing platforms for embedded systems	Low

"Full" coverage is for units that could be covered using Arduino and ready-made shields even if this requires a deeper study of the shield and its software library.

"Medium" coverage in units 7, 10, and 12 is chosen due to the difficulty of accessing a number of microcontroller hardware components unless Assembly is used.

"Low" coverage is selected for unit 9 as it is difficult to use Arduino to teach embedded operating systems [11]. Further studies are required to investigate if such systems currently available for Arduino[6,7] make it possible to cover this topic. “Low” coverage is selected for unit 13 as it is not possible to teach GPUs, FPGAs, or multi-core processors using Arduino.

This challenge could be also addressed on a different level by introducing interdisciplinary subjects and projects. Interfacing Arudino with MATLAB [19-20], LabView [21-23], and its use in the IoT domain [24-26] paves the way for unlimited possibilities of multidisciplinary applications.

The Lecture-related (HW/SW Compatibility) and the Instructor-related (Tools for Education) Challenge: The *first* question faced in a microcontroller course and one challenge highlighted in [6], is “What is the appropriate learning tool to be selected?” The existence of many microcontroller products renders the platform selection to be a critical step. It is safe to assume that this selection should be made to serve the industry needs. An embedded market study carried by Gartner[8] in 2014 revealed that 8-bit microcontrollers have around 40% market share. In that category, Microchip and Atmel were numbers 1 and 4 in the market. Recent Microchip acquisition of Atmel[9] makes the PIC and ATmega (on which the Arduino is built) microcontrollers the most dominant. Hence, if an 8-bit microcontroller is selected for embedded systems courses, and from a purely industrial point of view, PIC and Arduino could prove to be the best options. An interesting observation is that in [15], PIC is used in the course while Arduino is mostly selected by students in the project. However, in [16], Arduino is used in the course while PIC is one optional microcontroller suggested for the project.

This learning platform challenge was the focus of a study comparing six different embedded platforms used in freshmen engineering [27]. This was based on four metrics: *hardware-intensive*, *software-intensive*, *ease-of-implementation*, and *course/application relevance*. Arduino was found to be one of the platforms suited for such courses. In order to answer the “platform selection” question, an interesting future research direction is to conduct a similar study, comparing different microcontroller platforms (PIC, Arduino …) and/or microprocessor boards (Raspberry Pi, BeagleBone, Intel Galileo …) for embedded systems courses. Metrics could be based on the topics covered by the ACM/IEEE model curriculum. A different direction is to design your own board and customize it to the learning objectives of your course. However, it would be difficult to provide these boards for the students outside open lab hours.

The methodology of using a single platform through an educational curriculum [6, 8] could be applied using Arduino. Arduino has been adopted in many courses that span the engineering curriculum including Introduction to Engineering [18, 28-33], Chemistry [34-35], Physics [36-38], Electronics [39-41], Control and Robotics [19-20, 42-43], Fuzzy Logic [44], and DSP [45]. Arduino with its overall simplicity, availability of on-line resources, ease of acquiring parts, and fast prototyping process

---

[6] https://bitbucket.org/ctank/ardos-ide/wiki/Home

[7] https://create.arduino.cc/projecthub/feilipu/using-freertos-multi-tasking-in-arduino-ebc3cc

[8] http://ww1.microchip.com/downloads/en/Market_Communication/MicrochipPresentation_Evolution of 8-bit MCUs_Final.pdf

[9] http://www.microchip.com/announcements/microchip-technology-inc-acquires-atmel

overcomes many of the difficulties stated in [9]. However, this raises the question if adopting a single platform through the entire curriculum would limit the students' knowledge.

Revisiting the E2LP platform [6, 8], a number of its presented learning outcomes could be fulfilled by Arduino with the exception of the FPGA ones. Atmega could be adopted to teach microcontrollers using Assembly if required. Network shields could be used to illustrate different networking applications. The DSP shield in [45] could be used for DSP applications. One major concern would be if using already manufactured shields would provide proper learning tools. An interesting direction would be to propose project ideas for building Arduino shields. This would give the students the opportunity to design hardware circuits and experience low-level programming for developing the accompanied libraries. Such developed Arduino shields could be used in other courses; thus improving the overall learning experience.

The Lecture-related (Technology Cycle or Dynamic Progress of Technology) Challenge: Teaching embedded engineering in schools [7] is identified as one of the solutions to address this problem [3, 6, 8]. Arduino is already contributing in that direction as it has been adopted in pre-university education [46-50]. The work in [51] presented a high-school outreach program developed in New Zealand. Part of the program provided teachers with Arduino kits and a suggested set of experiments to be conducted at school. This helped to increase enrollment figures by 36% in one year.

The Development vs. Recombination Challenge: This challenge identified in [6] highlights the need to find the right balance between the students developing their own code and reusing existing codes. Such a challenge falls in the classification of Fig. 1 under both methodologies for education (Teaching how to recognize reuse opportunities) and evaluation and assessment procedures (How to assess Arduino-based projects?). Same issues were raised [11-12, 14, 16] as authors discussed concerns about the *lack of low-level experience* and the *reuse of existing Arduino code*.

According to the 2015 UBM[10] embedded market study; around 75% of embedded applications are programmed in C/C++ while only around 3% are programmed in Assembly. In addition, when asked about the programing language most likely to be used in the next application, around 83% chose C/C++ while only 2% chose Assembly. Such staggering numbers raises the question of how important the teaching of low-level programming in embedded systems courses still is.

This brings us to the *second* question: If Arduino is used in embedded systems education, how can one balance low-level vs. high-level knowledge? And how much low-level programming is required? Answers to these questions could very much depend on the course structure. Is it a lecture-based course? Is there a laboratory component involved? Is it a project-based course? Or is it a combination of all of these? As mentioned in [6], the use of remote laboratories can also improve engineering education. Some works on Arduino-based remote laboratories could be found in [52-54]. Note that Arduino could be programmed with Assembly using AVR studio [16]. This has the benefit of introducing the students to a new software tool other than the Arduino IDE used for programming the Arduino in its C-like language..

[10] http://webpages.uncc.edu/~jmconrad/ECGR4101-2015-08/Notes/UBM Tech 2015 Presentation of Embedded Markets Study World Day1.pdf

The same 2015 UBM market study showed that 86% of developed applications reused previously written code, some of which was open-source. This shows that reusing Arduino code is a behavior that students will face in industry and it will enhance the students' system integration capabilities as identified in [11]. However, this should not undermine the importance of students providing their own contribution.

This raises the *third* question: If Arduino is used in embedded systems education, how can one guarantee that students will provide enough contribution in their projects? One approach requires the submission of a pre-project documentation citing similar existing projects, if any, and providing details of the students' own contributions [12, 15]. This document is reviewed by the instructor, modified if necessary, and finally approved before the students can start working on their projects. This is an essential step as with the available Arduino resources, there is a great probability that any proposed project will be at least partially available on some forum or website. Never the less, this should not be viewed as a complete disadvantage as it was shown in [55] that online forums are supportive means for engineering students looking to expand their knowledge and make connections to other students outside the classroom.

5 Conclusions

In this paper, the subject of embedded systems education was revisited with the introduction of Arduino. The paper surveyed embedded education challenges recently identified in the literature. Moreover, the paper covered previous works integrating Arduino in embedded systems courses.

It was found that Arduino proved to be a very promising educational platform in embedded engineering. It can be utilized to cover a lot of the core units under the embedded systems knowledge area in the 2016 model curriculum for computer engineering. In addition, it can be used to overcome a number of challenges facing embedded education nowadays.

Although Arduino has a clear promise, it cannot be stated with certainty that it is a suitable platform for embedded education. Hence, a number of research directions were proposed in this work to further examine this subject. One direction is to conduct a research study comparing different microcontroller platforms for higher-level education. Another direction is to develop effective teaching methodologies that guarantee the delivery of the learning outcomes, with the appropriate depth, in the case of using Arduino. More specifically, how to balance low-level vs. high-level knowledge? And how to make sure that students provide considerable contribution in their projects? Furthermore, it remains open to investigate the implementation of Arduino-shields projects and the use of Arduino in covering the topic of embedded operating systems.

6 References

[1] Minaie A. and Sanati-Mehrizy R. (2208). Comparison of Embedded Systems Education in the United States, European, and Far Eastern Countries. In the American Society for Engineering Education Annual Conference, pp. 13.19.1-13.19.6.

[2] 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. In ACM Transactions on Embedded Computing Systems, vol. 4, no. 3, pp. 500-528. https://doi.org/10.1145/1086519.1086522

[3] Ibrahim I., Ali R., Zulkefli M., and Elfadil N. (2015). Embedded Systems Pedagogical Issue: Teaching Approaches, Students Readiness, and Design Challenges. In the American Journal of Embedded Systems and Applications, pp. vol. 3, issue 1, pp. 1-10. https://doi.org/10.11648/j.ajesa.20150301.11

[4] Marwedel P. (2005). Towards laying common grounds for embedded system design education. In the ACM SIGBED Review – Special Issue on the First Workshop ion Embedded Systems Education, vol. 2, issue 4, pp. 25-28.

[5] Chandrasekaran S., Stojcevski A., Littlefair G., and Joordens M. (2013). Project-Oriented Design-Based Learning: Aligning Students' Views With Industry Needs. In the International Journal of Engineering Education, vol. 29, no. 5, pp. 1109-1118.

[6] Kastelan I., Teslic N., and Termerinca M. (2016). Challenges in Embedded Engineering Education. In Advances in Intelligent Systems and Computing, vol. 421, pp. 1-27. https://doi.org/10.1007/978-3-319-27540-6_1

[7] MacBride G., Hayward E. L., Hayward G., Spencer E., Ekevall E., Magill J., Bryce A. C., and Stimpson B. (2010). Engineering the Future: Embedding Engineering Permanently Across the School-University Interface. IEEE Transactions on Education, vol. 53, issue 1, pp. 120-127. https://doi.org/10.1109/TE.2009.2025368

[8] Kastelan I., Lopez B. J. R., Artetxe G. E., Pwinski J., Barak M., and Temerinac M. (2014). E2LP: A unified embedded engineering learning platform. Microprocessors and Microsystems, vol. 38, no. 8, pp. 933-946. https://doi.org/10.1016/j.micpro.2014.09.003

[9] 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. In the IEEE International Conference on Microelectronic Systems Education (MSE), pp. 69-72.

[10] Bird N. (2011). Use of The Arduino Platform for a Junior-level Undergraduate Microprocessors Course. In the American Society for Engineering Education Annual Conference, pp. 22.1600.1- 22.1600.11.

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

[12] Jamieson P. and Herdtner J. (2015). More Missing the Boat - Arduino, Raspberry Pi, and Small Prototyping Boards and Engineering Education Needs Them. In the IEEE Frontiers in Education (FIE), pp. 1442-1447. https://doi.org/10.1109/fie.2015.7344259

[13] Dasig Jr. D. (2014). User Experience of Embedded System Students on Arduino and Field Programmable Gate Array (FPGA). In the Second International Conference in Applied science and Environmental Engineering, pp. 124-128.

[14] Herzog P. and Swart A. (2016). Arduino – Enabling Engineering Students to obtain Academic Success in a Design-based Module. In the IEEE Global Engineering Education Conference (EDUCON), pp. 66-73.

[15] El-Abd M. (2016). How Course Projects can Successfully Prepare Engineering Students for Capstone Design Projects. In the IEEE Global Engineering Education Conference (EDUCON), pp. 746-750. https://doi.org/10.1109/EDUCON.2016.7474635
[16] Patińo O. A., Conterars-Ortiz S., and Martínez-Santos J. C. (2016). Evolution of Microcontroller's Course under the Influence of Arduino. In 14th LACCEI International Multi-Conference for Engineering, Education, and Technology: "Engineering Innovations for Global Sustainability".
[17] Rodriguez-Sanchez M. C., Torrado-Carvajal A., Vaquero J., Borromeo S., and Hernandez-Tamames J. A. An Embedded Systems Course for Engineering Students Using Open-Source Platforms in Wireless Scenarios. IEEE Transactions on Education, doi: 10.1109/TE.2016.2526676 https://doi.org/10.1109/TE.2016.2526676
[18] Santos R. R. (2014). Open Hardware Platforms in a First Course of the Computer Engineering Undergraduate Program. In Frontiers in Education (FIE), pp. 2085-2091.
[19] Al-Busaidi A. (2012). Development of an Educational Environment for Online Control of a Biped Robot using MATLAB and Arduino. In Mechatronics-REM, pp. 337-344. https://doi.org/10.1109/mecatronics.2012.6451030
[20] Cheng H., Hao L., Luo Z., and Wang F. (2016). Establishing the Connection between Control Theory Education and Application: An Arduino Based Rapid Control Prototyping Approach. The Journal of Learning and Teaching, vol. 2, no. 1, pp. 67-72.
[21] Kuwan W-H., Tseng C-H., Chen S., and Wong C-C. (2016). Development of a Computer-Assisted Instrumentation Curriculum for Physics Students: Using LabVIEW and Arduino Platform. The Journal of Science Education and Technology, vol. 25, pp. 427-438. https://doi.org/10.1007/s10956-016-9603-y
[22] Williams W. (2013). LabVIEW and Arduino as a gateway to PLC programming. In the American Society for Engineering Education Annual Conference, pp. 23.846.1-23.846.12.
[23] D. Cãlinoiu, R. Ionel, M. Lascu, and A. Cioablã. (2014). Arduino and LabVIEW in Educational Remote Monitoring Applications. In the IEEE Frontiers in Education (FIE), pp. 245-249.
[24] Georgitzikis V., Akribopoulos O., and Chatzigiannakis I. (2012). Controlling Physical Objects via the Internet using the Arduino Platform over 802.15.4 Networks. IEEE Latin America Transactions, vol. 10, no. 3, pp. 1686-1689. https://doi.org/10.1109/TLA.2012.6222571
[25] Mullet G. (2016). Teaching the Internet of Things (IoT) Using Universally Available Raspberry Pi and Arduino Platforms. In the American Society for Engineering Education Annual Conference. https://doi.org/10.18260/p.26053
[26] Dobrilovic D. and Zeljko S. (2016). Design of Open-Source Platform for Introducing Internet of Things in University Curricula. In the 11th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), pp. 273-276.
[27] Pritchard J. and Mina M. (2013). Modern Embedded Systems as a Platform for Problem Solving in Freshman Engineering: What is the Best Option? In the American Society for Engineering Education Annual Conference, pp. 23.911.1-23.911.12.
[28] Recktenwald G. and Hall D. (2011). Using Arduino as a Platform for Programming, Design, and Measurement in a Freshman Engineering Course. In the American Society for Engineering Education Annual Conference, pp. 22.1609.1-22.1609.23.
[29] Vasquez H. and Fuentes A. (2013). Integration of Sensors and Low-Cost Microcontrollers into the Undergraduate Mechanical Engineering Design Sequence. In the American Society for Engineering Education Annual Conference, pp. 23.789.1-23.789.13.
[30] Tian J. (2014). A Design Approach in an Introduction to Engineering Course. In the American Society for Engineering Education Annual Conference, pp. 24.44.1-24.44.12.

[31] Orr M. K., Swafford C., Hahler S., and Hall D. (2104). Factors that influence confidence: Untangling the influences of gender, achievement, and hands-on activities. In the IEEE Frontiers in Education (FIE), pp. 2580-2584.

[32] Berry C., Chang D., and Miller C. (2016). From LEGO to Arduino: Enhancement of ECE Freshman Design with Practical Applications. In the American Society for Engineering Education. https://doi.org/10.18260/p.26972

[33] Wu N. and Zeng K. (2016). Embedded System Based First-Year Engineering Course with Aid of Online Simulation and Social Media. In the 9th First Year Engineering Experience (FYEE) Conference.

[34] Butterfield A. and Branch J. (2015). Results & Lessons Learned from a Chemical Engineering Freshman Design Laboratory. In the American Society for Engineering Education Annual Conference, pp. 26.1337.126.1337.25.

[35] Kubinova S. and Slger J. (2015). ChemDuino – Low cost chemical measurements for use in lecture and laboratory. In the Journal of Chemical Education, vol. 92, issue 10, pp. 1751-1753. https://doi.org/10.1021/ed5008102

[36] Zachriadou K. and Yiasemides K. (2012). A low-cost computer-controlled Arduino-based educational laboratory system for teaching the fundamentals of photovoltaic cells. In the European Journal of Physics, vol. 33, pp. 1599-1610. https://doi.org/10.1088/0143-0807/33/6/1599

[37] Huang P. (2015). Open-source Hardware – Microcontrollers and Physics Education – Integrating DIY Sensors and Data Acquisition with Arduino. In the American Society for Engineering Education Annual Conference, pp. 26.1205.1-26.1205.13.

[38] Haugen A. and Moore N. A model for including Arduino microcontroller programming in the introductory physics lab. arXiv:1407.7613v1 [physics.ed-ph].

[39] Kendre S. S., Mulmule P. V., and Shirbahadurkar S. D. (2013). Developing Experimental Platforms Using Common Software Tools For Enhancing Technical Skills of Electronics Engineering Students in Microcontrollers. In the IEEE Frontiers in Education (FIE), pp. 1450-1452. https://doi.org/10.1109/fie.2013.6685072

[40] Onime C., Uhomoibhi J., and Zennaro M. (2014). A Low Cost Implementation of an Existing Hands-on Laboratory Experiment in Electronic Engineering. In the International Journal of Engineering Pedagogy (iJEP), vol. 4, issue 4, pp. 4-7. https://doi.org/10.3991/ijep.v4i4.3707

[41] Yoder R. B. (2015). An Arduino-Based Alternative to the Traditional Electronics Laboratory. In the Conference on Laboratory Instruction Beyond the First Year (BFY), pp. 107-110. https://doi.org/10.1119/bfy.2015.pr.027

[42] Sobota J., Pisl R., Balda P., and Schlegel M. (2013). Raspberry Pi and Arduino Boards in Control Education. In the 10th IFAC Symposium Advances in Control Education, pp. 7-12. https://doi.org/10.3182/20130828-3-uk-2039.00003

[43] Candela F. A., Garcia G. J., Puente S., Pomares J., Jara C. A., Perez J., Mira D., and Torres F. (2015). Experiences on using Arduino for laboratory experiments of Automatic Control and Robotics. In The International Federation of automatic Control (IFAC), pp. 105-110. https://doi.org/10.1016/j.ifacol.2015.11.221

[44] Albayrak A., Albayrak M., and Bayir R. (2015). Design of Matlab/Simulink Based Development Board for Fuzzy Logic Education. In the IEEE International Conference on Fuzzy Systems, pp. 1-7.

[45] Esposito W. J., Mujica F. A., Garcia D. G., and Kovacs G. T. A. (2015). The Lab-In-A-Box project: An Arduino compatible signals and electronics teaching system. In the IEEE Signal Processing and Signal Processing Education Workshop (SP/SPE). https://doi.org/10.1109/DSP-SPE.2015.7369570

[46] Hill L. and Ciccarelli S. (2013). Using a Low-Cost Open Source Hardware Development Platform in Teaching Young Students Programming Skills. In the 14th Annual ACM SIGITE Conference on Information Technology Education, pp. 63-68. https://doi.org/10.1145/2512276.2512289
[47] Tenra K. (2014). Practice of the Programming Education using Arduino and the Class Support System. In the 22nd International Conference on Computers in Education (ICCE), pp. 969-974.
[48] Bojic I. and Arratia J. F. (2015). Teaching K-12 Students STEM-C Related Topics through Playing and Conducting Research. In the IEEE Frontiers in Education (FIE), pp. 548-555. https://doi.org/10.1109/fie.2015.7344109
[49] Tsukamoto H., Takemura Y., Nagumo H., Ikeda I., Monden A., and Matsumoto K-I. (2015). Programming Education for Primary Schoolchildren Using a Textual Programming Language. In the IEEE Frontiers in Education (FIE), pp. 1008-1014.
[50] Herger L. M. and Bodarky M. (2015). Engaging Students with Open Source Technologies and Arduino. In the IEEE Integrated STEM Education Conference (ISEC), pp. 1-7. https://doi.org/10.1109/isecon.2015.7119938
[51] Carnegie D. and Watterson C. (2013). Issues of Recruitment and Retention for a New Engineering Provider. In the IEEE Frontiers in Education (FIE), pp. 665-671.
[52] Fotopoulos V., Spiliopoulos A. I., and Fanariotis A. (2013). Preparing a remote conducted course for microcontrollers based on Arduino. In The 7th International Conference in Open and Distance Learning, pp. 133-139.
[53] Thornton M. A., Manikas T. W., and Laplante P. A. (2013). Embedded and Real-time Systems Classes in Traditional and Distance Education Format. In Frontiers in Education (FIE), pp. 1379-1385.
[54] Parkhomenko A., Gladkova O., Ivanov E., Sokolyanskii A., and Kurson S. (2015). Development and Application of Remote Laboratory for Embedded Systems Design. In the International Journal of Online Engineering (iJOE), vol. 11, issue 3, pp. 27-31. https://doi.org/10.3991/ijoe.v11i3.4519
[55] Teo H. J., Johri A., and Brogan D. S. (2013). Towards an Understanding of ECE Students' Use of Online Homework Help Forums. In the IEEE Frontiers in Education (FIE), pp. 400-404. https://doi.org/10.1109/fie.2013.6684854

7 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 specific interests in embedded systems, micro-processors, micro-controllers, computer architecture, robotics and artificial intelligence.

Article submitted 06 March 2017. Published as resubmitted by the author 27 April 2017.