# Integrating Hardware Prototyping Platforms into the Classroom

Eyhab Al-Masri
School of Engineering and Technology
University of Washington Tacoma
Tacoma, Washington, USA
ealmasri@uw.edu

***Abstract*—Computer education is in the midst of a major transformation as a result of the rapid emergence of new technologies, intersection of computing with other disciplines and growing interest among students in computing courses. Such transformation prompted educators to redefine the way computing education is delivered and experiment a wide range of learning approaches. One of these approaches is the integration of hardware prototyping platforms such as Arduino, Raspberry Pi and Intel Internet of Things as part of project-based learning. These platforms have become in recent years instrumental tools for improving students' learning process and promoting proactive thinking. The aim of this study is to investigate the usefulness of integrating low cost open-source hardware platforms into engineering and computer science courses. In particular, we explore the use of an experiential learning approach using these kits for improving the students' learning experience in computer science and engineering courses. Throughout this paper, we discuss the outcome of our investigation and provide insights on how to make classrooms more experiential using the hardware prototyping paradigm.**

***Keywords— Curriculum, Computing, Education, Courses, Classrooms, Project-Based Learning, Raspberry Pi, Arduino.***

## I. Introduction

Hardware prototyping platforms such as Arduino and Raspberry Pi are becoming increasingly popular across both the research and teaching communities. Over the past few years, several institutions began working on developing educational modules to enhance the engineering and computer science curricula through the integration of prototyping development boards into lectures and laboratory sessions. On the other hand, many startup projects on crowdfunding platforms such as Kickstarter.com make use of these hardware prototyping platforms. In another effort, companies such as Think Engineer specialize in helping startup companies and researchers in developing fully functional, high-tech prototypes that are based on Raspberry Pi and Arduino. Furthermore, the popularity of such open source platforms has also prompted microcontroller manufacturers such as Intel, NXP Semiconductors, STMicroelectronics, among many others to create development kits or boards that are compatible (e.g. header compatibility) such that these boards or shields can be easily integrated into the Arduino or Raspberry Pi platforms.

At the education level, educators have been becoming increasingly motivated to adopt such platforms into classrooms. For example, educators at the University of Granada offer introductory programming courses using Arduino [1]. At Case Western Reserve University, researchers have been investigating the effectiveness of a system engineering approach to incorporate technology into the enhanced classroom-based curriculum [2]. Educators at the Miami University are investigating the challenges in adopting hardware prototyping platforms from an educational standpoint into the engineering curriculum [3]. Researchers at the Columbia University are developing lab kits using the Arduino prototyping platform for delivering consistent, high quality laboratory experience for on-campus and remote students [4]. In [6], the Raspberry Pi platform has been adopted in a novel project-based teaching and learning approach devised in an Internet of Things course for undergraduate students. The paper presents evidence of increased engaging environment for learning and experimentation among students.

More recently, MIT's Internet of Things (IoT) bootcamp provided students with the ability to understand IoT topics by supplying them with Arduino, Raspberry Pi and sensor kits [7]. In [8], instructors integrate the Raspberry Pi platform into a mobile applications and IoT course. In [9], an engineering course at the University of Maryland integrates Arduino and Raspberry Pi as part of student-driven projects in an attempt to strengthen students' understanding of mechatronics concepts and extending them to the Internet of Things domain. In [16], the authors designed and built internet-accessible laboratories for remote access that uses hardware prototyping modules. A computer science course in robotics focuses on the use of Raspberry Pi as an essential component of the course design and technology learning objectives [10]. At Florida Atlantic University (FAU), a course on embedded electronics uses the Raspberry Pi and Arduino-based autonomous robots to teach students the basic building blocks of the Internet of Things [11]. At the University of Buffalo, a course on real-time and embedded operating systems integrates these hardware prototyping platforms [12].

To investigate the effectiveness of integrating these hardware prototyping platforms into classrooms, we designed and developed multiple learning modules that describe how to use them for the first time (e.g. out-of-the-box experience). These modules were used in a number of project-based courses in computer science. Some of the developed learning modules were integrated into class sessions as demonstrations. Additional tutorial or laboratory sessions were also designed. These sessions aim to provide students with hands-on computing experience using the hardware prototyping platforms. Participating students had access to use the Arduino-based or a Raspberry Pi starter kits (e.g. CanaKit) that include

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

components such as prototyping board, sensors (e.g. temperature, flame, tilt), dot matrix display, resistors, jumper wires, photoresister, potentiometer, multiple LEDs, and breadboard. The cost of these kits range between $45 to $90 and students had access to these kits at the start of the semester.

The selection of the Arduino and Raspberry Pi as prototyping platforms for integration into the surveyed courses was based on several factors including: (a) ease of use, (b) availability of open source projects, (c) user community, and (d) cost. To engage students, the lecture demonstrations included examples such as detection of sensor data, displaying patterns using LEDs, generating melodies, using servo motors to control movements, among others.

## II. Case Study 1: Internet of Things

In this graduate Internet of Things (IoT) course (taught by one of the authors) and offered at a North American public university [12], students were introduced the fundamental IoT functional blocks and architecture, protocols and communication models. Students learn how to build IoT-based solutions and apply the (technical) knowledge acquired through a course project. Coursework included four reading assignments, eight in-class activities and two 2-hour exams. Students were asked to work in groups of 2-3 students each and participate in building a real-world IoT application. Each group was provided two kits: (a) Canakit Raspberry Pi 3 Ultimate Starter Kit [13] and (b) GrovePi+ for Raspberry Pi [14].

One of the main goals of this case study is to investigate the effectiveness of applying an experiential learning approach that utilizes hardware prototyping platforms in this IoT course. To this extent, one of the exams was developed in two parts: (a) theoretical component and (b) practical component. Both components were graded equally (25 points). The 1-hour theoretical component aimed at testing students on the basic understanding of IoT concepts such as physical and logical design, building blocks of the IoT architecture, protocols and communication models, and IoT enabling technologies. The 1-hour practical component involved students' engagement in a collaborative computer programming task. In particular, this exam component aimed at testing students on the ability to utilize IoT development tools to build an IoT application within the given time frame. To achieve this, students were asked to develop an IoT application that collects sensor data and sends the data to ThingSpeak, an open source IoT platform store and retrieve data from IoT devices in real-time [14]. We will focus on one of the exam results to demonstrate the usefulness of our experiential learning approach.

Results from this study, have shown that students attained a higher exam grade in the practical component compared to that of the theoretical one. In particular, students achieved an average of 95% in the practical component compared to 87% in the theoretical component. The grading for the practical component did not only focus on a working prototype but also additional components including the utilization of the IoT functional building blocks, IoT design methodology and standard coding practices (i.e. code documentation, program structure, appropriate data structures and exception handling). Figures 1 and 2 present a screenshots of sample working prototypes and the corresponding real-time representation on the ThingSpeak IoT platform.

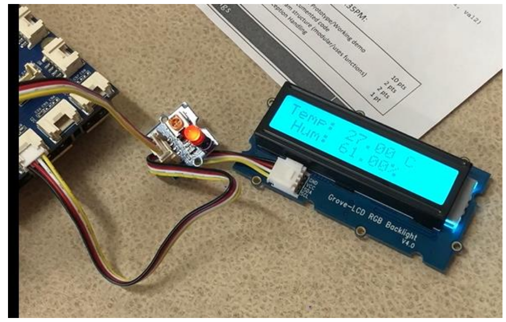


Figure 1: Working Prototype

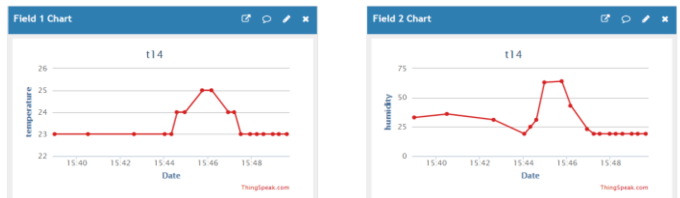


Figure 2: ThingSpeak Representation

## III. Case Study 2: Introduction to Programming

In this undergraduate introductory programming course taught by one of the authors, students were encouraged to participate in acquiring the (technical) knowledge through an optional course project. Coursework included four reading assignments, eight in-class activities and two 2-hour exams. Students were asked to work in groups of 2-3 students. Participating groups were provided additional hands-on tutorials on the use of the Arduino and Raspberry Pi starter kits.

The chief goal of this case study is to assess the effectiveness of integrating open-source hardware platforms into introductory (first year) programming undergraduate courses. In particular, we were interested in determining the impact in terms of students' motivation in the coursework and computer science programming through this integration. To this extent, we compare and contrast the coursework performance of the participating group (students that participated in an optional project) and the non-participating group (students that did not participate in an optional project). Although the number of participating and non-participating students is not equal, the outcomes from this comparative study demonstrate that the participating groups attained higher overall coursework grades compared to the non-participating group by more than 14% as shown in Figure 3.

Results presented in Figure 3 demonstrate that the participating groups' motivation is significantly higher compared to the non-participating group. For example, the participating group outperforms the non-participating group across all of the coursework components including assignments, quizzes, and exams. In addition, the impact of this integration is evident by the difference in the overall performance of the final exam which is more than twice of that of the midterm exam.

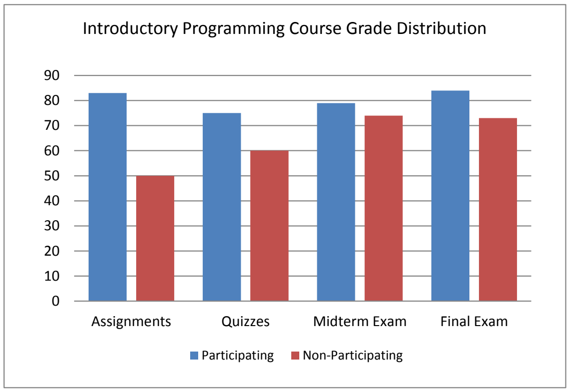


Figure 3: Programming Course Grade Comparison

It was noted that many of the errors discovered in the execution of students' code for the non-participating group comprise of common programming mistakes including: (1) inconsistent coding style (i.e. proper documentation, indentation, meaningful names), (2) lack of modularity (i.e. writing long functions, hard to identify functional components of the program structure, many paths through the code), excessive number of control structures in a single function, (3) excessive use of global variable declarations, and (4) mishandling exceptions (i.e. lack of handling errors). This is likely due to the fact that many of these topics are thoroughly learned and acquired to a great extent with the use of hardware prototyping platforms. For example, in order to constantly read temperature data via the Raspberry Pi, it is essential to declare a looping mechanism (i.e. a while loop) to continuously capture sensor data. A major part of writing a program to achieve this task is to create a function that reads sensors data and properly handling exceptions in case there is a keyboard interrupt and/or input-output error. As shown in Figure 3, the participating group's learning grows toward the completion of the coursework and demonstrates that the students' learning curve is constantly increasing throughout the semester.

## IV. Case Study 3: Software Engineering

In this undergraduate software engineering course, students were provided learning material that focused on the fundamental software engineering concepts mainly requirements determination and specifications. This course (taught by one of the authors) included a 3-hour weekly lecture, a biweekly tutorial, a course project, a 1-hour midterm exam and a 3-hour final exam. In order to facilitate the integration of hardware prototyping platforms in this software engineering course, students were asked, as part of their required projects, to develop a system prototype by the end of the term to users and project sponsors. This way, students are able to build a tangible product that enables them to understand the importance and practicality iterative nature of the software development life cycle (SDLC) in delivering a software product within a short period of time (i.e. three months).

One of the main goals of prototyping using hardware platforms in this software engineering course is to facilitate requirements analysis strategies and requirements gathering techniques such that it would assist students working in groups to define and quickly refine real requirements of a system prototype. To this extent, groups were asked to develop applications of a Radio Frequency Identification (RFID) management system using the open-source hardware platforms (e.g. access control, asset tracking, event management, parking control, airline and airport, car dealership, property management, inventory and retail, sensor control, among many others).

The aim of this case study is to provide students with an opportunity to relate requirements determination to real-world applications. In particular, we were interested in making the software engineering course more experiential, enlighten students' thoughts to deepen their learning by relating fundamental software engineering concepts to real-world applications and enable students to develop self-directed learning skills. The impact of integrating hardware prototyping platforms into the software engineering course is demonstrated by the results shown in Figure 4. In order to assess the effectiveness of our approach in this course, we examined the students' performance during the semester in which hardware prototyping was integrated (participating group – blue lines) and compare the results to the same course taught a year earlier (by the same instructor) that did not involve any hardware prototyping (non-participating group – red lines).

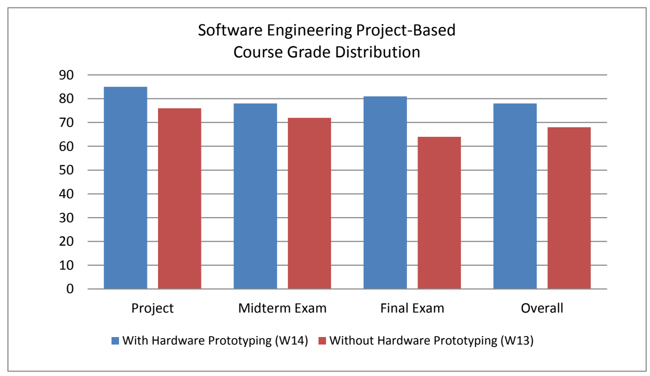


Figure 4: Software Engineering Course Grade Comparison

As shown in Figure 4, the participating group outperforms the non-participating group across all coursework components and attained significantly higher grades by more than 9%. This is a reflection of the students' motivation in the coursework. That is, the participating group's motivation is significantly higher compared to the non-participating group. This was also evident by the quality of the projects and presentations performed by the students at the end of term. In addition, the integration of the hardware prototyping platforms into this software engineering course has deepened the students' intellectual development in which they have engaged in critical thinking at various levels (i.e. technical and non-technical) with the goal of developing critical thinking and problem solving skills while applying knowledge gained throughout the course across contexts.

## V. Discussion and Lessons Learned

The use of hardware prototyping platforms has been integrated into multiple courses at various levels, graduate and undergraduate. For the undergraduate courses, we have integrated these platforms in a first year course (i.e. an

**Table 1. Student Feedback from Courses that Used Hardware Prototyping Platforms**

| Survey Question | Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree |
|---|---|---|---|---|---|
| The use of hardware prototyping platforms has increased my interest in engineering or computer science | 84% | 8% | 8% | 0% | 0% |
| Because of my experience with the use of the hardware prototyping platforms in this class, it is more likely that I will take courses in engineering or computer science | 67% | 17% | 16% | 0% | 0% |
| The use of hardware prototyping platforms helped create a challenging and exciting atmosphere in and outside classroom | 68% | 25% | 7% | 0% | 0% |
| Working with hardware prototyping platforms has strengthened my intellectual curiosity and critical-thinking skills | 67% | 33% | 0% | 0% | 0% |
| The course has inspired my desire to learn and grow my knowledge in engineering or computer science | 75% | 25% | 0% | 0% | 0% |

introductory programming course), a senior level course (i.e. software engineering) and a graduate course (i.e. Internet of Things). Across all of these courses, results have demonstrated usefulness of integrating hardware prototyping platforms such as Arduino and Raspberry Pi into the classroom environment.

To further demonstrate the effectiveness of the students' learning experience through the integration of these open-source hardware platforms, students in the programming and software engineering courses were invited to participate in an end-of-term survey. Fifty-three students participated in this survey. The survey was presented to students after the completion of the course requirements and results of this survey are shown in Table 1.

Results from the survey show that 84% of students strongly agree that the use of hardware prototyping platforms has increased their interest in engineering or computer science. In addition, 93% of the students that participated in this post-semester survey either strongly agree or agree that the use of hardware prototyping platforms helped create a challenging and exciting atmosphere in and outside classroom. The information is significant in the sense that it reflects the students' futuristic goals and the likelihood of using hardware platforms into future course projects.

## VI. Conclusion and Future Work

As we rely more on computing devices to perform our daily activities, it becomes crucial to integrate new technologies into curriculum design for enhancing the students' learning process and making classrooms more experiential. Hardware prototyping platforms such as Arduino and Raspberry Pi offer students the means for embracing the intellectual challenge by making creative ideas accessible to all learners.

In this study, we presented intriguing results that demonstrate the effectiveness of integrating hardware prototyping technologies into software engineering and computer science courses. Using hardware prototyping tools in classrooms shows high student engagement rates, increases the students' motivation in course material, creates an exciting atmosphere in the classroom and enhances the students' overall learning experience. More importantly, open-source hardware platforms can effectively make classrooms more experiential. As a future work, we would like to examine the effectiveness of this integration across a broader number of courses and collaborating with other institutions. We also would like to expand this research study to develop more concrete measures of the effectiveness of integrating open-source hardware platforms and making classrooms more experiential.

## References


[1] Rubio, M. and Zaliz R. "Enhancing an introductory programming course with physical computing modules." IEEE Frontiers in Education Conference, 2014, pp. 1-8.

[2] Yang, P. "A system engineering approach: Integrating technology into the classroom-based curriculum." International Conference on Systems and Informatics, 2012, pp. 1000-1004.

[3] Jamieson, P. "More missing the Boat — Arduino, Raspberry PI, and small prototyping boards and engineering education needs them." IEEE Frontiers in Education Conference, 2015, pp. 1-6.

[4] Sarik, J., Kymissis, I. "Lab kits using the Arduino prototyping platform." IEEE Frontiers in Education Conference, 2010: 1-5.

[5] Zhong, X., & Liang, Y. Raspberry Pi: An Effective Vehicle in Teaching the Internet of Things in Computer Science and Engineering. Electronics, 2016, 5(3), p. 56.

[6] MIT Internet of Things Bootcamp, https://bootcamp.mit.edu/iot/ (Last Accessed July 8, 2018)

[7] CSCI 318: Mobile Apps and the Internet of Things, http://home.wlu.edu/~levys/courses/csci318f2016/syllabus.html, (Last Accessed April 25, 2018)

[8] ENME 489B Mechatronics and the Internet Of Things, http://meugrad.umd.edu/wp-content/uploads/2015/09/ENME-489B.pdf (Last Accessed July 8, 2018)

[9] CSCI 373 Robotics Project, http://www.cs.unca.edu/~bruce/Fall14/373/, (Last Accessed July 8, 2018)

[10] CSE 321 Realtime and Embedded Operating Systems, https://www.cse.buffalo.edu/~bina/cse321/fall2017/SyllabusAug28.pdf, (Last Accessed July 8, 2018)

[11] COT 4930/5930 – Embedded Robotics (aka Internet of Things or IOT), http://robotics.fau.edu/2014/11/12/spring-2015-course-embedded-robotics-aka-internet-of-things , (Last Accessed July 8, 2018)

[12] University of Washington Tacoma, Internet of Things Course, https://www.tacoma.uw.edu, Last Accessed July 8, 2018.

[13] Canakit Raspberry Pi Ultimate Kit, https://www.canakit.com/raspberry-pi-3-ultimate-kit.html, (Last Accessed July 8, 2018)

[14] GrovePi+ Kit for Raspberry Pi, https://www.seeedstudio.com/GrovePi%2B-Starter-Kit-for-Raspberry-Pi-p-2240.html, (Last Accessed July 8, 2018)

[15] ThingSpeak, https://en.wikipedia.org/wiki/ThingSpeak, (Last Accessed July 8, 2018)

[16] E. Al-Masri, "Lab-as-a-Service (LaaS): A Middleware Approach for Internet-Accessible Laboratories", IEEE 48th Annual Frontiers in Education (FIE) Conference, 2018.