Computational Thinking and Solving Problems – an experience with Arduino in a Electronic Engineering Career

Milagros Zegarra[1], Elizabeth Vidal[1]

[1] Universidad Nacional de San Agustín de Arequipa

mzegarra@unsa.edu.pe

evidald@unsa.edu.pe

Abstract. Computational thinking is a fundamental skill since it helps to improve analytical ability. Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the fundamentals of computer science. Literature has shown different ways to bring to classroom computational thinking. Most of the experiences are related to programming classes with different kind of software according to the age. This paper present and exploratory study that describes our experience related to the first programming course at the University level for teaching computational thinking for solving problems. The use of Arduino to teach computational thinking has been incorporated into the Electronic Engineering degree at the Universidad Nacional de San Agustín de Arequipa since 2017. Students program components oriented to solve problems related to their profession. The first results have shown the effectiveness in the use of Arduino to develop computational thinking. This first experience gave us the foundation to expand our research to analyze quantitative data regarding to student outcomes about solving problems. We believe that our experience can be replicated not only in other Electronic Engineering careers, but in any engineering that have programming courses. We also believe that our experience could be replicated at a high school level.

Keywords: Arduino, Computational Thinking, Solving Problems, STEM

1 Introduction

The main definition of computational thinking is state that "computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" [1]. Another definition is presented by García-Peñalvo [2] that refers to the application of high level of abstraction and an algorithmic approach to solve any kind of problems. This second definition come closer to the diverse kind of implementations related to computational thinking [3], to develop students' problem solving skills throughout programming approaches [14].

But when we bring computational thinking to University level, programming courses are already designed. Several experiences have been found in the literature that make use of Arduino to teach programming. Rubio et al [4] describes a modular course that combine Arduino with Lego. The authors highlight the high degree of motivation that the students showed. Duch and Jaworski [5] presented the experience of Lodz University of Technology (Poland) which, using Arduino, aims to make programming learning more attractive by expanding the interaction with devices such as joysticks, light emitting elements, keyboards and radio communications. In more specialized courses there are four laboratory experiments for automatic control and robotics courses at the University of Alicante. The results showed that the proposed experiments were attractive to students [6]. Also in the work of Brock, Bruce and Reiser [7] it is emphasized that the use of Arduino is an inexpensive way to teach embedded system design and introductory programming courses. The work highlights the use of simple electronic interfaces used in the real world using LEDs and sensors. Titon and Ramirez [13] presented the experience of teaching programming concepts using educational robotics supported by the Arduino. All the described experiences mentioned before are related to programming courses.

This papers show our experience in the use or Arduino under a first programming course format for teaching computational thinking and solving problem at the University Level. We describe the main content of the course. We present the sequence of assignments and highlight the prototypes presented by students focused on solving real problems. The rest of the paper is organized as follow: section 2 describes the methodology, section 3 shows our initial results about using Arduino and some discussion. Finally we expose our firsts conclusions.

2 Methodology

Exploratory studies serve to familiarize researchers with relatively unknown phenomena, obtain information on the possibility of carrying out a more complete investigation in a particular context, investigate new problems, identify promising concepts or variables, establish priorities for future research [11]. Within this framework we can say that our work is exploratory and is a first approach to identify variables to study in the future about the impact of keep on working with Arduino to teach computational thinking to develop students' problem solving skills throughout programming approaches.

2.1 Background

Arduino [8] is an open source microcontroller board with very easy to use hardware and software. Arduino handles a series of inputs which can be connected to various sensors and thereby control motors, lights, buzzers, etc. The main features are: economical, multiplatform, simple programming and Open Source. The microcontroller board is programmed using Arduino Programming Language [9].

2.2 Course Description

The Programming for Electronics 2 course is taught in the second semester of the Escuela Profesional de Ingeniería Electrónica [10]. It consists of 17 weeks, 4 credits with 5 theoretical hours and 2 laboratory hours. Since 2017 the use of Arduino has been incorporated. Based on the recommendations of the IEEE / ACM for careers in Electronic Engineering [11], the topics taught in this course are listed in Table 1.

Table 1. Topics of the course.

Topic	Week
Algorithm concepts, basic principles of their development	1,2
Structures Repetition Defined (Repeat) using all methods	3
Indefinite Repeat Structures (While) using all methods	4
Array Structures	5
Introduction to Arduino as a development tool	6
Project implementation starting from the medium difficulty level	7-9
Bluetooth use	10
Android applications	11
Implementation of medium-high level projects	12 - 17

The structure of the course has been based on the practical component through projects which are detailed in Table 2. Students receive the characteristics of each project which are implemented in groups of two students.

Table 2. Projects developed in the semester using Arduino

Week	Project Description
2,3,4	Solar Panel: depending on the light intensity, the generated voltage measurement is carried out.
5,7	Temperature Sensor: build a temperature measurement system, using the Arduino Uno, as well as temperature sensors such as the LM35 or DHT11, or similar ones that allow us to measure the temperature or of a particular environment
9	Motion Sensor and Magnetic Sensor: in this project we use sensors that allow us to create safety projects.
11	LED Matrix or LCD Display: in this project we use either the matrix or the display to be able to send previously programmed messages.
12	Proximity sensor: It gave us the possibility of defining actions to be performed depending on the proximity or remoteness of the objects in relation to the proximity sensor.

Figure 1, Figure 2 and Figure 3 shows some of the implementation made in 2018.

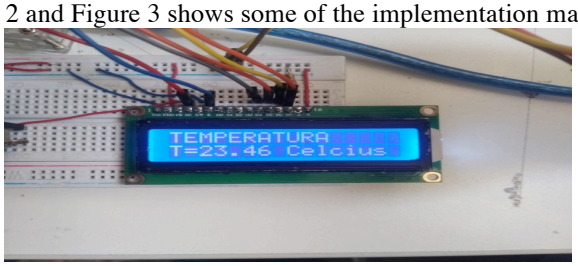

Fig. 1. Temperature Sensor project. In this project students build a temperature measurement system, using the Arduino Uno that allow them to measure the temperature or of a particular environment

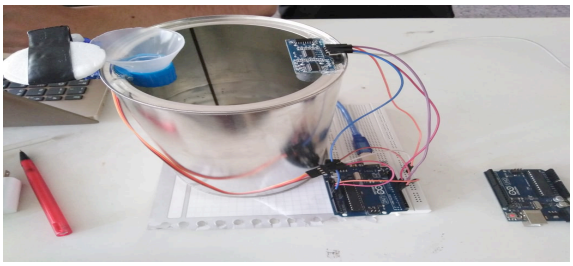

Fig. 2. Proximity Sensor project. In this project the students used the proximity sensor to detect the water level of a container. The signal has been captured by the sensor and according to the programming already configured in the Arduino, depending on the distance read a light signal was shown to be able to give a first alert and an audible signal to give a final alert indicating that the maximum allowed limit was reached.

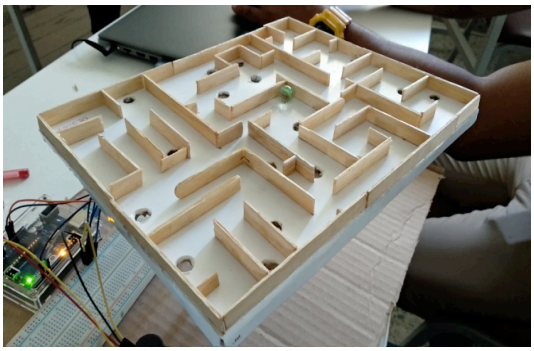

Fig. 3. Labyrinth project. The goal is to have a ball running from the beginning to the end without falling into the holes of the path. Students are able to use a joystick or a keyboard to perform the movements of a pair of servos or stepper motors which had to move in the two axes.

Students have developed different kind of projects with Arduino, but the focus was the high level of abstraction and an algorithmic approach to solve the different kind of projects.

3. Results and Discussion

Since this was an exploratory study, to assess the impact of the use of Arduino regarding the perception of students a survey was applied using the Likert scale. Of the 75 students who took the course, 57 answered the Survey. Our initial findings show us that almost fifty students have a a favorable opinion about using Arduino (Fig. 4).

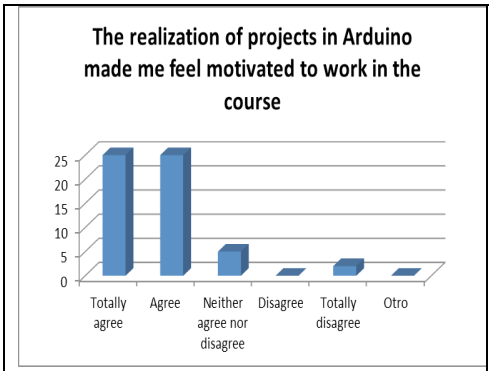

Fig. 4. Results of the perception of using Arduino related to motivation

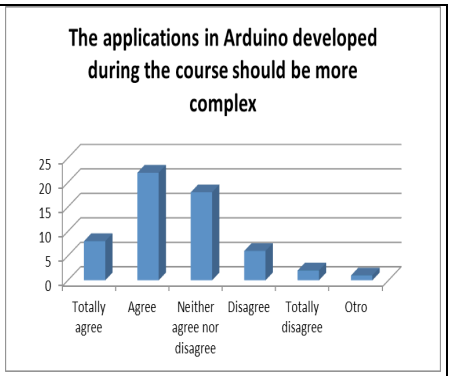

Fig. 5. Students' perception regarding the difficulty of the projects developed. Slightly more than half of the students believe that they could develop more complex projects.

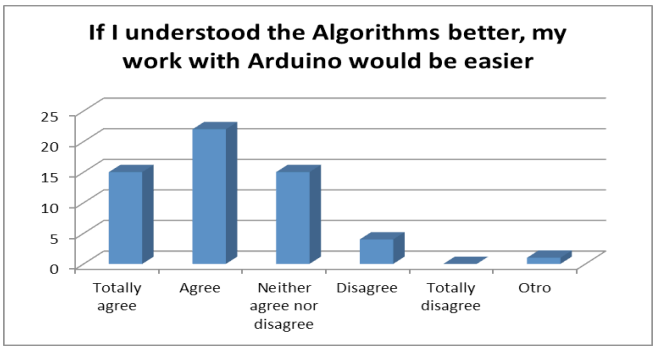

Fig. 6. Students' perception regarding to the importance of algorithms for problem solving

Also we found that slightly more than half of the students surveyed believe that they could develop more complex projects (Fig. 5). As we can see in Fig. 6, students are aware that if they handled algorithm better, it would be easier for them to solve problems with Arduino. We can also highlight the importance of algorithms so that students can translate into a programming language such as in this case the Arduino programming.

4 Conclusions

In this work we have describe the experience of the use of Arduino through the realization of projects for teaching computational thinking focus on high level abstraction and solving problem thorough algorithms. The initial results demonstrate that Arduino plays an important role in terms of the motivation and active learning of the students. The design of the course supported by Arduino with a projects

oriented approach focus on the development of real applications is suitable for develop high level abstraction skills . Designing a course with a high practical content like the one shown in this experience requires a different work than traditional pedagogical approaches. The design of the activities focus on the student requires the professor additional work in the selection of appropriate materials and especially choosing motivating and challenging projects. As a future work, a quantitative analysis is sought based on the results of achievements through the students' grades in years prior to the use of Arduino.

References

1. Wing, J. M. (2006). Computational thinking. *Commun. ACM* **49**, 33–35.
2. García-Peñalvo, F.J., 2016. What Computational Thinking Is. *Journal of Information Technology Research 9*, 3, v-viii.
3. García-Peñalvo, F.J., Rees, A.M., Hughes, J., Jormanainen, I., Toivonen, T., and Vermeersch, J., 2016. A survey of resources for introducing coding into schools. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'16) (Salamanca, Spain, November 2-4, 2016)*, F.J. García-Peñalvo Ed. ACM, New York, NY, USA
4 Rubio, M., Mañoso, C., Zalis, R. and De Madir, A. (2014). Uso de las plataformas LEGO y Arduino en la enseñanza de la programación. Jornadas de Enseñanza Universitaria de la Informática (20es: 2014: Oviedo)
5 Duch, P. and Jaworski, T. (2018). "Enriching Computer Science Programming Classes with Arduino Game Development." 2018 11th International Conference on Human System Interaction (HSI). IEEE.
6 Candelas, F. A., Garcia, G. J., Puente, S., Pomares, J., Jara, C. A, Pérez, J. and Torres, F (2015) . Experiences on using Arduino for laboratory experiments of Automatic Control and Robotics. IFAC-PapersOnLine, 48(29), 105-110.
7 Brock, J. D., Bruce, R. F. and Reiser S. L. (2009). Using Arduino for Introductory Programming. J. of Computing Sciences in Colleges, p. 129-130. Vol 25 Issue 2
8 Aruduino. (2019). Arduino. http://www.arduino.cc/.
9 Massimo, B. and Shiloh, M. (2014). Getting started with Arduino: the open source electronics prototyping platform. Maker Media, Inc.
10 Joint Task Force on Computer Engineering Curricula, IEEE Computer Society, Association for Computing Machinery, "Computer Engineering 2016: Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering," Tech. Rep., 2004. [Online]. Disponible en https://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf,
11 Escuela Profesional de Ingeniería Electrónica. Disponible en .http:/fips.unsa.edu.pe/ingenieriaelectronica/. Ultimo acceso Enero 2019.
12 Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2010). Metodología de la investigación.
13 Titon, W., and Ramirez, García, A. (2018). Teaching Programming Concepts Using Educational Robotics, Supported by the Arduino Platform: An Application in the Industrial Computer Learning Course. In Procc *2018 XIII Latin American Conference on Learning Technologies (LACLO)* (pp. 100-106). IEEE.
14 García-Peñalvo, F. J., and Cruz-Benito, J. (2016). Computational thinking in pre-university education. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 13-17). ACM.