



A Study on Student Attitudes in Learning Programming using Physical Computing

Zuraini H. Zaini¹, Noreen I. Arshad¹, Balbir S. M. Singh¹, Nurshazlyn M. Aszemi¹,
Subuh Anggoro² and Santhy Hawanti²

¹ Universiti Teknologi PETRONAS, MALAYSIA

² Universitas Muhammadiyah Purwokerto, INDONESIA

Received 10 November 2018 • Revised 5 February 2019 • Accepted 4 March 2019

ABSTRACT

Learning to program can be difficult for the students. Students must master language syntax, programming theory and problem-solving techniques. Efforts have been made to assist students in understanding how to program. This study is intended to examine whether Arduino, as a teaching and learning tool, helps in generating students' interests towards programming. Arduino is one of the physical computing tools which has open-source electronics platform based on user-friendly hardware and software for creating different projects and applications. Arduino is easy to be used by beginners, yet flexible enough for advanced users to learn physical computing and programming. This study adopted a quantitative research method to measure the student's attitude in learning programming using physical computing. The sample of this study is 56 students from foundation programme and undergraduate programme. In order to gauge students' perception, students' attitude survey was adapted. The collected data were analyzed using descriptive analysis. Based on the analysis, the study found that, the overall mean score was 4.253. The result indicated that student has positive attitude in learning programming using physical computing.

Keywords: computational thinking, learning, physical computing, programming.

INTRODUCTION

Programming has been recognized as one of the critical competencies that require students to use computational tools to address real-world problems in the 21st century [1,2,3]. Learning programming is not only a prerequisite for becoming a computer scientist, but it is also necessary for the practice of solving problems and designing systems [4]. Programming requires programmers to plan solutions to problems, precisely transform the plans into syntactically correct instructions for execution and assess the consequent results of executing those [5]. However, research revealed that at the conclusion of introductory programming courses, most students have difficulties in decomposing problems, developing plans and implementing their plans with programming languages to solve programming problems [6], some of them lack adequate understanding of fundamental programming constructs, and most of them lack strategies for transforming programming problems into workable plans and

© 2019 by the authors; licensee PGSD UMP. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>).
✉ zuraini.zaini@utp.edu.my (*Correspondence) noreenizza@utp.edu.my
✉ balbir@utp.edu.my ✉ nurshazlyn_17007352@utp.edu.my
✉ santhyhawanti.ump@gmail.com ✉ subuhanggoro@ump.ac.id

algorithms [7]. This may be because the formal instruction in programming mostly focuses on students' mastery of a general-purpose programming language and adopts a programming tool that is intentionally designed for professional programmers [8]. The employment of the general-purpose programming language and the professional programming tool often drives the teachers and students to invest their efforts more on mastering programming language features than on developing design strategies for solving programming.

Several studies suggest that new teaching methodologies might help the student to overcome their initial difficulties and improve their results. Among the studies aiming to devise potential means for enhancing programming, an alternative approach to engaging students in solving computational problems [9] has been recognized as an effective way of cultivating students' programming constructs and skills. This method often provides students with computational problems, which are specially designed to foster specific programming concepts or skills. In a scenario requiring students to solve a computational problem by exercising various programming knowledge and strategies, the students are expected to learn by formulating computer programs and systematically evaluating the consequent results [10]. Many studies have also proposed alternative approaches to the students' difficulties in programming using physical computing in learning to program such as robotic and microcontroller. This approach takes the computational concepts "out of the screen" and into the real world so that the student can interact with them [10]. Therefore, the aim of the study is to measure the student's attitudes towards learning programming using physical computing.

There are several in the literature on Arduino-based educational kits for academic research and educational robotics projects. Therefore, in the following paragraphs, we discuss a few examples of what is currently available to provide a general overview of the topic of research.

Arduino teaches STEM education using Arduino in C Programming [11]. To satisfy this need, a series of hands-on laboratory assignments that contain various electronic components and sensors are created and supervised by two educational trainers. This project provides an interdisciplinary, hands-on approach to teach C programming. Together, the trainers and lab assignments also introduce key concepts in math and science while allowing students hands-on experience with various electronic components and sensors. This will allow students to mimic real-world applications of using the C programming language while exposing them to technology not currently introduced in many high school classrooms. The developed project is targeted at high school students performing at or above the junior level and uses the Arduino Mega open-source Microprocessor and software as the primary control unit.

Meanwhile Hoffer uses the Arduino platform to enhance students' learning experiences [12]. They present preliminary experiences using the Arduino microprocessor platform in the undergraduate computing curricula, at both the upper and lower levels. The goal is to enhance students' learning by engaging them in a contextualized project-based learning experience and introducing them to

fundamental computing and engineering concepts in the context of a highly visual and easy to use environment.

Mellodge and Russell presents a low-cost educational robotics kit based on the Arduino Uno platform [13]. The prototype is intended to be applied in secondary (high) schools by means of educational workshops on robotics. The project is supported by step-by-step documentation (e.g. booklet) that addresses fundamental physics, mathematics, logic programming and robotics concepts. It also offers all the steps for evolving in the construction of a robot, and it employs a block-structured environment (such as Minibloq) to allow easier programming.

Junior et al. utilises Arduino as a Learning Tool for a Mechanical Engineering Measurements Lab [14]. She has found that students are not sufficiently engaged in laboratory courses. To solve this problem, she incorporates the Arduino platform as an additional teaching tool in the lab. Students will gain exposure to a wider variety of sensors and will understand the basics of data acquisition using Matlab. Innovation is being tested that semester for the first time in a mechanical engineering undergraduate laboratory. The low cost of the Arduino platform allows each student to borrow a board and a set of sensors for the entire semester. The feedback from the students is to have more open-ended and practical lab activities.

Arduino allows the students to experiment with new educational processes, experiences and ways of learning. Through problem-based learning (PBL), students develop competencies necessary to be effective 21st-century workforce, including problem-solving, critical thinking, collaboration, communication, and creativity or innovation [15]. 21st-century skills have been the focus of education reform in developing countries. The Partnership for 21st Century Skills, an international organisation (<http://www.p21.org/>) that advocates for 21st century readiness for every student states that "In an economy driven by innovation and knowledge, society is facing complex business, political, scientific, technological, health and environmental challenges, collaborative relationship and social networking are crucial to competitiveness."

MATERIAL AND METHODS

Methods

Centre for Foundation (CFS), Universiti Teknologi PETRONAS believes that it is crucial for students to experience physical computing as one of the teaching and learning tools in understanding the programming subject. Therefore, a series of Arduino course have conducted to exposed student on programming using physical computing technology.

The goal of this course is to introduce the field of computing using simple projects, electronics and programming. Arduino has been introduced to the students as a physical computing learning tool. According to Arduino official website [16], Arduino is an open-source electronics platform based on easy-to-use hardware and software. It is intended for anyone doing interactive projects.

Arduino may have a direct impact on students in the sense that it brings to classroom live experiences by working on hands-on projects to build a prototype that can solve a real-life problem. It can also trigger students' creativity, critical thinking, collaboration and communication skills that are being emphasised in 21st-century skills. The research method used was survey research methods

Instrument

The Students Attitude towards Computer Science survey developed by [19] was adapted to measure the student's perception behaviour after learning programming using Arduino. The survey consisted of four constructs which are: Interest, Usefulness, Confidence and Professionalism – the description of the construct as in Table II. The survey was designed using Likert Scale from 1 to 5 (1: Strongly Disagree, 2: Disagree, 3: Neutral, 4: Agree, or 5: Strongly Agree).

Procedures

The participants were the students from foundation and undergraduate programme. Participation in this training was voluntary. The duration of this course is about 8 hours. A total of 56 students participated in Arduino training. Arduino Uno microcontroller board was selected as the physical computing learning tool because it is an open-source hardware board that is becoming increasingly common within the teaching community [17]. Arduino can interact with its surroundings through different types of sensors, switches, lights and motors. Each student was given a set of Arduino Starter Kit to carry out their Arduino assignment tasks.

The students downloaded Arduino IDE software from the Arduino website at no cost to write programmes using C or C++ programming language. In order to familiarize the students with Arduino environment, several reference webpages as well as sample projects were supplied. The students were being introduced to the basic programming concepts and syntax, for example, sequence, selection, looping, function and variables. Students' learning process during the Arduino lessons was evaluated through the students' responses in an online survey in regard to their learning experiences and learning of 21st-century skills

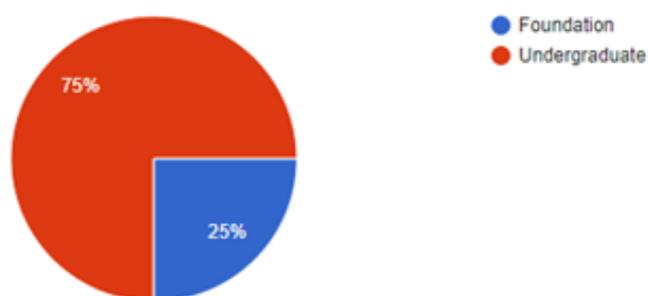


Figure 1. The percentage of the sample by the level of programme of study

Data Analysis

Table 1 shows the learning types and their descriptions that are carried out by the students [18]. To enhance their learning experiences on Arduino, the students are given an assignment, that is learning by creating, which requires them to apply design thinking processes to identify a real-life problem and build a prototype to solve it. The students work as a team to empathize and define the problem, brainstorm and come up with creative solutions, fabricate a prototype and test the prototype system.

Table 1. Learning types

Learning Types	Description
Learning by Imitating	The students imitate examples to learn about Arduino and how to use them.
Learning by Modifying	The students modify the attributes and functions of the systems and programmes based on basic examples.
Learning by Creating	The students apply the examples they have imitated and modified so far to design and develop new prototypes and programmes.

Grouping questions and items can be seen from the separation value . then the quality of the question instruments in terms of the overall respondents and items is getting better because it can identify groups of respondents and items in various ways. The survey consisted of four constructs which are: Interest, Usefulness, Confidence and Professionalism – the description of the construct as in Table II. The survey was designed using Likert Scale

Table 2. Description of Construct

Construct	Description
Interest	Students' interest in programming is generated through physical computing
Usefulness	Students believe the usefulness of learning programming
Confidence	Students confidence in their own ability to learn programming using physical computing
Professionalism	Students belief about professionals in programming area

The level of positive attitude was determined by using the following scale scores in **Table 3.**

Table 3. Level of Positive Attitude

Attitude Level	Scores
High Positive	4.00 - 5.00
Low Positive	3.01 - 3.99
Neutral	3.00
Low Negative	2.01 - 2.99
High Negative	1.00 - 2.00

RESULTS AND DISCUSSION

The data obtained from the questionnaire were analysed with descriptive statistics by using frequency in percentage form. The sample size, n , equal to 56. Among the 56 participants, 14 were foundation students (which was 25%) and 42 were undergraduate students (which was 75%). The grand means for each of the constructs are summarized in [Table 4](#).

Table 4. Mean Score of Survey Result

Attitude Level	Scores
Interest	4.152
Usefulness	4.198
Confident	4.275
Professionalism	4.386
Overall	4.253

Based on [Table 4](#) shown the mean overall positive attitude was 4.253, which was quite high. The undergraduate students seemed to be eager to be taught in a less formal way and appeared to be more enthusiastic about physical computing.

Concerning the interest construct, the undergraduate students mean score higher (4.20) than foundation students (4.00). the older students showed more interested in learning programming using physical computing. [Table 5](#) shows the means score between foundation and undergraduate students.

Table 5. Mean Score of Survey Result by Programme of Study

Attitude Level	Foundation	Under-graduate
Interest	4.00	4.20
Usefulness	4.30	4.17
Confident	4.19	4.30
Professionalism	4.29	4.42
Overall	4.195	4.273

As regards the usefulness construct, the younger students seemed to be much more aware of the extent to which physical computing in programming is useful to their own lives. However, both foundation and undergraduate students recognized that knowledge in programming and physical computing would allow them to secure an excellent job as indicated in the highest score is Professionalism construct (4.386). In the relation of a particular construct, the undergraduate students tended to be more confident than

foundation students. This could be biased as the undergraduate students already have the necessary knowledge of programming.

CONCLUSION

In this paper a result of student's perception towards learning programming using physical computing was presented. The survey's results indicated a high level of positive attitude based on Likert scale for all the programming related construct after the introductory of Arduino course in term of their own ability to learn programming skills, generate interest in programming, awareness of the usefulness of programming associating it to the higher chances of getting a good job. The participant's general feedback was optimistic, with unanimous agreement on the convenience of continuing these practices and approaching in the future

CONFLICT OF INTEREST

The authors declare no conflict of interest

AUTHOR CONTRIBUTIONS

Zuraini and Dr. Noreen design the study, implement the analysis, follow up the experiment, wrote the whole article and revised it.

Dr. Balbir co-supervised the research, revised and commented on the paper.

Subuh and Santhy co-research with Zuraini in conducting the courses with the students, distributing the survey and producing the results and discussion until the research finished.

Nurshazlyn, contributed in finalizing and editing the manuscript.

All authors had approved the final version.

REFERENCES

- [1] Einhorn, S. (2011). *Microworlds, computational thinking, and 21st-century learning*. Logo Computer System Inc, White Paper. Retrieved from <http://www.microworlds.com/>.
- [2] Grover, S., & Pea, R. (2013). Computational thinking in K-12: a review of the state of the field. *Educational Researcher*, 42(1), 38e43
- [3] Yen, C.-Z., Wu, P.-H., & Lin, C.-F. (2012). Analysis of expert's and novice's thinking process. *Engaging Learners through Emerging Technologies, Communication in Computer and Information Science*, 302, 122e134.
- [4] Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: what is next for K-12? *Computers in Human Behavior*, 41, 51e61
- [5] Ismal, M. N., Ngah, N. A., & Umar, I. N. (2010). Instructional strategy in the teaching of computer programming: a need assessment analyses. *The Turkish Online Journal of Educational Technology*, 9(2), 125e131.
- [6] Hazzan, O., Lapidot, T., & Ragnis, N. (2011). *Guide to teaching computer science: An activity-based approach* (2nd ed.). London: Springer.

- [7] Li, F. W. B., & Watson, C. (2011). Game-based concept visualization for learning programming. In Paper presented at the third international ACM workshop on multimedia technologies for distance learning, Scottsdale, AZ, USA.
- [8] Xinogalos, S. (2012). An evaluation of knowledge transfer from microworld programming to conventional programming. *Journal of Educational Computing Research*, 47(3), 251e277
- [9] Edmonds, J. (2008). *How to think about algorithms*. New York: Cambridge University Press.
- [10] G. T. Richard, "Employing Physical Computing in Education: How Teachers and Students Utilized Physical Computing to Develop Embodied and Tangible Learning Objects," *The International Journal of Technology, Knowledge and Society*, 2010
- [11] Arduino. (2018). What is Arduino? Retrieved from: <https://www.arduino.cc/>
- [12] Hoffer, B. M. (2012). Satisfying STEM Education Using the Arduino Microprocessor in C Programming. *Electronic Theses and Dissertations*. Paper 1472.
- [13] Mellodge, P. & Russell, I. (2013). Proceedings of the 18th ACM Conference on Innovation and Technology on Computer Science Education. Page 338
- [14] Junior, L. A., Neto, O. T., Hernandez, M. F., Martins, P. S., Roger, L.L., Guerra, F. A. (2013). A Low-Cost and Simple Arduino-Based Educational Robotics Kit. *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Robotics and Control (JSRC)*, December Edition, 2013 Volume 3, Issue 12
- [15] Basalo, I. (2014). Arduino as a Learning Tool for a Mechanical Engineering Measurements Lab.
- [16] Eguchi, A. (2015). RoboCupJunior for promoting STEM education, 21st century skills, and technological advancement through robotics competition. *Robotics and Autonomous Systems*. Elsevier Journal.
- [17] Jang, Y., Lee, W. & Kim, J. (2015). Assessing the Usefulness of Object-based Programming Education using Arduino. *Indian Journal of Science and Technology*, Vol 8 (S1), 89-96.
- [18] Grasel, J., Vonnegut, W. & Dodds, Z. (2010). Bitwise Biology: Cross-disciplinary Physical Computing atop the Arduino. *AAAI Spring Symposium Series*.
- [19] Hoegh, A., & Moskal, B.M. (2009). Examining science and engineering students' attitudes toward computer science. In *Proceedings of the 39th IEEE international conference on Frontiers in education conference* (pp. 1306-1311). San Antonio, Texas, USA: IEEE Press

<http://jurnalnasional.ump/index.php/dinamika>