







Innovative Robot for Educational Robotics and STEM

Avraam Chatzopoulos¹ , Michail Papoutsidakis¹ ,
Michail Kalogiannakis² , and Sarantos Psycharis³ 

¹ University of West Attica, Thivon 250, 12241 Aigaleo, Attiki, Greece
{xatzopoulos, mipapou}@uniwa.gr

² University of Crete, Gallos University Campus,
74100 Rethymnon, Crete, Greece
mkalogian@edc.uoc.gr

³ ASPETE, 14121 Heraklion, Athens, Attiki, Greece
spsycharis@aspete.gr

Abstract. This paper aims to present the design of a low-cost, open-source, robotic platform for use in Educational Robotics and STEM as a holistic approach to the curriculum. In alignment with the research presented in [1] the robotic platform's innovation is based on two axes: (a) its specifications came from the 1st cycle of participatory action research; (b) it is equipped with a visual programming language integrated into the robot's "brain" itself so that it can be programmed by any device (smartphone, tablet, PC) with Wi-Fi connectivity, without the need for any software or app to be downloaded and installed in the device. The spark for this research arose from an educational robotics survey's data evaluation handled at the municipality of Agia Varvara in Athens-Greece which, while showing a strong students interest in educational robotics, however few of them got involved because of the robotic platform's high cost. So, this research's motivation was to go on designing and developing a robotic platform suitable for the whole educational community that the specifications based on its members' needs and extracted by quantitative and qualitative data collection and analysis tools.

Keywords: Robot · STEM · Educational Robotics · VPL · Action research

1 Introduction

STEM term was firstly introduced by the NSF (National Science Foundation) in the 1990s as SMET [2], and it was used to refers to teaching and learning in the fields of Science, Technology, Engineering, and Mathematics (STEM derivatives), or it is used as a generic label for any action, policy, program or practice that involves one or more of its disciplines [3]. In the literature [1, 4–6] there is a big variety of the STEM education term's definition. There are two different approaches to integrate STEM into education [7]:

- The *content integration* that focuses on merging content fields into a single teaching activity to highlight "big ideas" from multiple content areas.
- The *contextual integration* that focuses on the content of a single scientific field, while frameworks from other disciplines are used to make the subject more relevant.

One such STEM integration is Educational Robotics (ER), a broad term that refers to a collection of activities, educational programs, technology or Robotic Platforms (RPs), educational resources, and pedagogical theories of learning within and outside schools [1, 8]. Robots and specifically ER have been gaining popularity in recent years and Greece, the main mobility currently seen in STEM education in schools concerns ER applications [1]. There has been a great interest among researchers [9], and educators [10], since ER [11] is a powerful learning and supportive tool for the development of cognitive-social skills. ER is introduced in many learning environments as an innovative teaching and learning tool [12] that supports students: (i) developing high-level skills, (ii) creating multiple representations of understanding the object, (iii) constructive communication and collaboration between them, (iv) developing and improving their learning by solving complex authentic problems, (v) implementing abstract design ideas to reflect and immediately notice the results of this effort, (vi) facilitating student learning through research and experimentation by contributing to the development of knowledge in the STEM areas. Besides, STEM and ER activities promote problem-based learning and critical thinking, as they focus on research and analysis of a complex real-world problem and it is also important the play aspect involved in making them particularly attractive to students especially in primary education [13]. Furthermore, gender stereotypes are often the basis of gender roles, the behaviors that society teaches that are ‘correct’ for boys and girls. Several studies have suggested that ER can help to eliminate gender stereotypes about the role of girls with and technology [14–16].

2 Problem Statement

The purpose of the present research is, to design and develop a low-cost, open-source RP for primary school students, to be used as an educational tool for applying ER activities and STEM education. The RP implementation takes into account the students’ needs extracted from the 1st cycle of participatory action research. The research’s objectives are: (i) The 1st version of RP’s design and specifications, are based on observations and survey results, in the framework of action research beginning with an ER event -as described briefly below- and, (ii) RP’s development has to be low-cost, based on open-source software and hardware, and can be programmed using a visual programming language (e.g. block-based programming). In November 2019, in the context of an ER event handled at the municipality of Agia Varvara (Athens, Greece), a survey was conducted to record students and parents on ER and STEM education [17]. Among others were questions about students’ experience in educational robots and the willingness to engage in their development. The results were particularly interesting. The majority (49%) of the participants didn’t know what STEM education is, they are not sure (50%) about what is ER is, but the vast majority of them (86%) know RPs, particularly the Lego RP (68%) [18]. While the vast majority of them (83%) had not participated in an ER seminar in the past, they would like to get involved and develop (88%) their robot and the majority of them (55%) would have unlimited time to complete its construction. Moreover, particularly encouraging is the fact that the vast majority (67%) of their parents want to get involved and support their children in

robots' development. And that was this research's motivation to go on designing and developing an RP suitable for the whole educational community (students, teachers and parents). RP's specifications were extracted by the combination of quantitative and qualitative data collection and analysis tools, to summarize the most important of them: (i) the vast majority of the participants (54%) prefer to program the RP using any device (smartphone, tablet, PC), (ii) they want (92%) RP's software-hardware to be open-source, (iii) they would like (50%) to build their RP using a 3D printer, and (iv) the RP should be compatible with old technology devices (40%). As to RP's total cost, two are the most prevalent trends: (i) a 36% of the participants want the RP's cost to range between 31€-50€, and (ii) a 32% of them would prefer a cost range between 51 €-100€. This last specification is also the research's milestone in RP's design since the goal was to keep its cost close to 50€, as opposed to the high cost (180€) of the corresponding Lego WeDo II. To recap and following the above remarks came the initial specifications of the RP such as: (i) has to be low-cost and open-source, (ii) has to be programmed by any device (and also be compatible with old technology), and (iii) can be built using a 3D printer.

3 Robotic Platform's Architecture

RP was designed and developed with ease of construction and use. Several considerations had to be done to fulfill the educational community needs:

- Keep the hardware cost as low as possible (close to 50€).
- Keep the hardware free from "exotic" difficult to find electronic parts.
- Use open-source hardware and software the most.
- Can be easily be assembled by students, teachers, and parents.
- It can be programmed by almost any device: smartphone, tablet, PC.
- No need for software to download and install. A *Visual Programming Language (VPL)* will be embedded into the RP.

The RP's construction consists of a mechanical chassis, 2 servo motors converted to DC, sensors, actuators, 2 microcontrollers (μC), electronic parts for interfacing, and various mechanical parts such as wheels, M4 screws, nuts, and washers, etc. A complete part list is shown below in Table 1.

All the electronic schematics, mechanical designs, and software were designed and developed by the researchers and would be freely available to the educational community under the Attribution-By Creative Commons (CC-BY) license [19]. This implementation based on the use of perforated aluminum plates (Fig. 1), offering a modular construction with the possibility to extend the robot frame in both three dimensions. Besides, the above construction has been introduced into educational activities and has been positively evaluated [20], however, the final version will have a frame of plastic parts that can be easily manufactured using a 3D printer. The Arduino Uno was chosen as the RP's main μC because of its low cost and high availability (there are also many Uno's cheap clones in the market) [21]. Although Uno is programmed in C/C++ through the Arduino IDE [11], this implementation uses a different innovative approach; it uses the ESP 32 μC [22] as an intermediate web server to integrate Blockly

Table 1. RP’s components list.

Qty	Part	Qty	Part
1	Base (perforated aluminum plate)	2	Side plate (perforated aluminum plate)
2	Motor support plate (perforated aluminum plate)	1	Third wheel support plate (perforated aluminum plate)
1	Front plate (perforated aluminum plate)	2	Servo motors (converted to DC)
2	Wheel with tire (3D printed)	1	Ball caster wheel
1	Basic electronics board	1	ESP32 microcontroller (webserver)
1	Arduino Uno microcontroller (RP’s main “brain”)	1	Logic level converter (to interface ESP’s TX, RX signals with UNO’s)
1	Sonar HC-SR04 (for distance meter)	1	LDR (Light Dependent Resistor)
1	L293D motor driver IC	2	Push buttons
1	Buzzer	2	LED (Red color)
7	Resistors (1 × 100 Ω, 2 × 220 Ω, 2 × 1K, 2 × 10K)	80	Dupont Wires (male to male, male to female, various cm)
2	USB cable (1 × type A to B, 1 × type B to micro USB)	1	Power supply (220VAC 50 Hz Output: 12 V @ 2 A)
1	Breadboard (to connect experimental electronic circuits)	80	Fasteners and hardware (M4 screws, nuts, washers, board spacers)

[23], an open-source Block based VPL focus to young age students. In this way, it innovates in that Uno programming is implemented through any device (smartphone, tablet, PC) equipped with Internet connectivity (Wi-Fi card and browser software) and not through specialized software or app that the user has to install on his device.

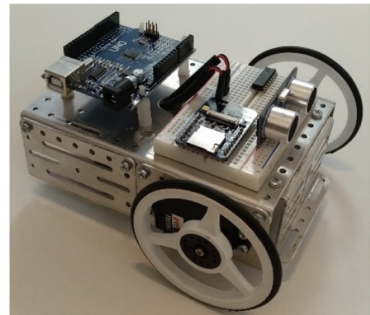
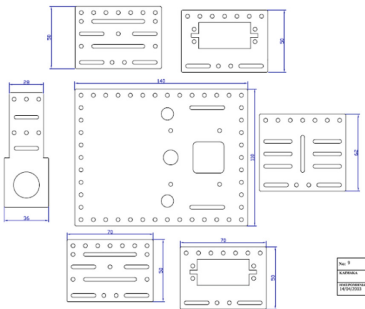


Fig. 1. Bottom view of RP’s mechanical parts and RP’s photo.

In Figs. 2 and 3 the RP’s electronic schematic and block electronic diagram are respectively presented. Arduino Uno is used as the basic programming controller (μC) for the RP. It is connected to motors through the L239D motor driver to support the needed current (350–500 mA per motor) [24].

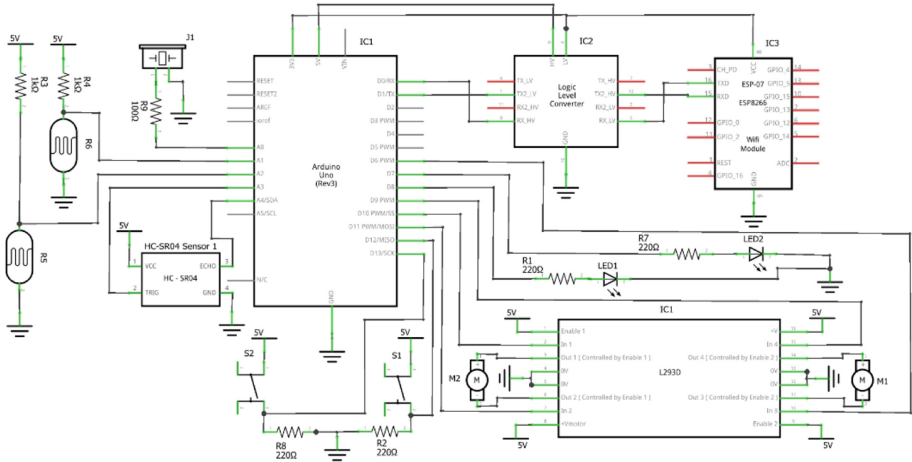


Fig. 2. RP's electronic schematic

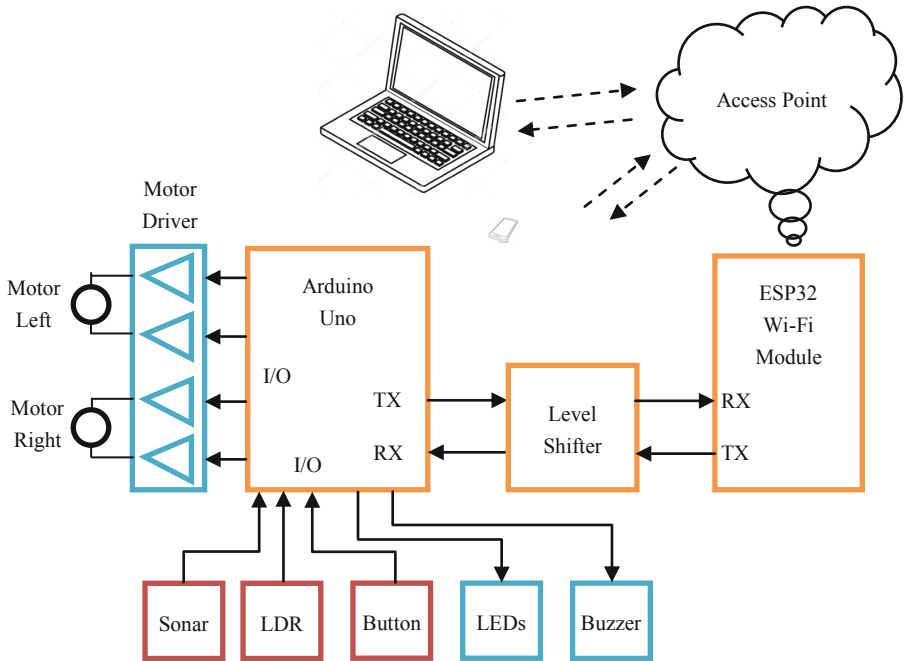


Fig. 3. Robot's hardware block diagram.

It is also connected to the sensors (Sonar, LDR and buttons) and the actuators (LEDs, buzzer) through electronic-circuit interfaces [25]. ESP μ C is primarily used as a web server. It is connected to Uno through the TX, RX signals to upload syntactically correct C/C++ code. A logic level converter circuit is necessary as an intermediate between the two μ Cs' because of their different voltages use. ESP is responsible for (i) client-server connection, (ii) listen and answer client's requests, (iii) send Blockly's HTML, CSS and JS code to the client, (iv) convert Blocks to C/C++ code and (v) upload code to Uno. Uno is responsible for (i) control motors, (ii) read digital (sonar, button) and analog sensors (LDR), and (iii) switch on/off actuators (LEDs, buzzer). Arduino Uno - ESP32 combination is chosen mainly because of the low cost. However, there are many other more expensive open-source μ Cs that may be used instead such as Arduino Uno Wi-Fi, Arduino MKR Wi-Fi, Arduino YUN, etc.

4 Software Implementation

RP's software implementation includes: (i) Arduino's functions for reading and storing sensors data, and for driving RP's actuators (motors, LEDs, buzzer), (ii) ESP's web-server software to serve client (devices) requests, (iii) webpage software and Blockly library, implemented in HTML, CSS, JavaScript, which are responsible for webpage's UI and visual programming language (VPL). Blockly is integrated into the ESP μ C, to provide RP's ability to be programmed without the need for the client's software installation. Blockly allows users to create programs through graphical manipulation [26], by dragging blocks around a screen and using flow diagrams, state diagrams, another component wiring, and icons or non-text representation. Blockly is an open-source developer library that provides a block editor UI and a framework for generating code in text-based languages such as JavaScript, Lua, PHP, Dart, and Python; custom generators for other text languages may also be created [26]. Blockly code is represented by *blocks*, which may be dragged around the screen and have connection points where they can be attached to other blocks and chained together (Fig. 4). For generating Arduino's C/C++ code there is a dedicated Blockly version called Blockly@duino [27].

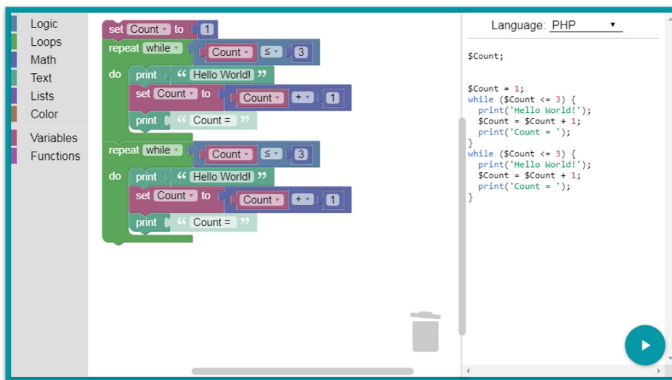


Fig. 4. Blockly's blocks and generating PHP code

Blockly’s downside is that does not provide a full vocabulary of blocks or a runtime environment, so it needs to integrate it with some form of output which was this research’s case.

5 Robotic Platform’s Kinematics

This RP employs a differential drive of a two-wheel vehicle to achieve mobility in two-dimensional space (Fig. 5). In this way, the RP can move in all directions (forward, backward, left and right) by directly controlling the speed and direction’s rotation of the DC motors [11].

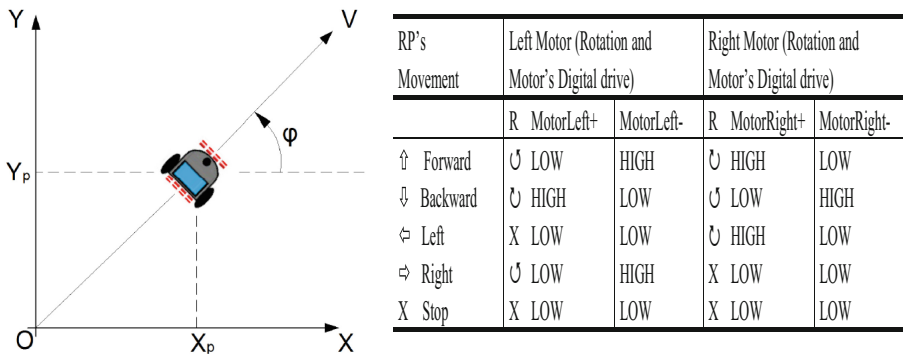


Fig. 5. RP’s dynamics-coordinates and RP’s movements concerning its motors’ rotation

RP’s posture as seen in the above figure is described by the following mathematics [11]:

$$\begin{bmatrix} \dot{X}_p \\ \dot{Y}_p \\ 0 \end{bmatrix} = \begin{bmatrix} \cos \varphi & 0 \\ \sin \varphi & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} V \\ \omega \end{bmatrix} = F(\varphi)u \tag{1}$$

Where: V is the RP’s linear velocity, ω is the RP’s rotational velocity, φ is RP’s orientation, X_p and Y_p are the RP’s coordinates of the center of its mass, and the vector u is the control command depending on the right and left wheels’ speeds. RP’s linear (V) and rotational (ω) velocities can be obtained from the right and left wheels’ velocities, according to the next functions:

$$\omega = \frac{d\varphi}{dt} = \frac{V_R - V_L}{b} \tag{2}$$

$$V = \frac{V_R + V_L}{2} \tag{3}$$

Where: V_R and V_L are the velocity of the right and left wheels respectively, and b is the distance between the centers of the right and left wheels. These equations are implemented in the RP's movement control functions; a sample code of RP's variable-speed forward movement with respect to its motors' rotation (Fig. 5) is presented:

```
const int MotorLeftP=10; const int MotorLeftM=11;
const int MotorRightP=9; const int MotorRightM=6;
pinMode(MotorLeftP,OUTPUT); pinMode(MotorLeftM,OUTPUT);
pinMode(MotorRightP,OUTPUT); pinMode(MotorRightM,OUTPUT);
void loop() {forwardSpeed(128); delay(1000)}
void forwardSpeed(int RPspeed) {
  //PR speed range value: 0-255 (RP's speed: 0-100%)
  digitalWrite(MotorLeftPlus,LOW);
  analogWrite(MotorLeftPlus,RPspeed);
  digitalWrite(MotorRightPlus,RPspeed);
  analogWrite(MotorRightMinus,LOW);}
```

6 Discussion

According to researchers, educators, and Papert, ER has numerous advantages and benefits for students [1, 8, 28–30]: (i) improves concentration and the overall learning process, (ii) increases motivation to learn, (iii) offers hands-on exposure to a wide range of subjects (mechanical, electrical, computer engineering) and is a useful aid for STEM, (iv) remains students' high levels of attention and curiosity (v) develops cognitive and social skills including teamwork, problem-solving, creativity, and robot design, (vi) attracts students to technological and scientific studies, and (vii) encouraged students to promote their interest and improve their English ability. From a technical perspective, the first step towards ER and STEM is to choose and use an RP [1]. It may seem simple however the above ER event's survey (see section *Problem Statement*) evidence a lack of students' engagement in ER due to the high RPs cost. In this research, a low-cost RP implementation was thoroughly presented, where its specifications came from the 1st cycle of action research results. According to the educational community's needs, the RP's specifications should be the following: low-cost (<50 €), open-source, should be programmed by any device, should be compatible with old technology devices, and should be built by a 3D printer.

7 Perspectives and Future Work

The next step of this research is the RP's evaluation. For this purpose, a future workshop and an experiential seminar are scheduled to introduce students and teachers to the RP and measure theirs' acceptance, thoughts, and suggestions. These results would be used to improve RP's design and repeat an action research circle. The main goal is for this RP to have widespread acceptance by the educational community, so the future step is a large scale evaluation study to collect quantitative and qualitative data.

Although it's building architecture based on low-cost hardware, electronic and software solutions, it still ensures the efficiency of the platform for many beyond educational applications. While the current design focuses on using the RP by younger students – due to the embedded VPL-, it can also be used in conjunction with other programming environments (e.g. Arduino IDE, MATLAB, etc.). For this purpose, future releases of the RP can be evolved by incorporating the following specifications: (i) adding a camera for image processing, (ii) swapping μ C to Raspberry Pi [28] with Ubuntu [29] for machine learning support, (iii) adding a speech synthesizer circuit and Bluetooth connectivity, plus a large-scale evaluation.

References

1. Chatzopoulos, A., Papoutsidakis, M., Kalogiannakis, M., Psycharis, S.: Action research implementation in developing an open source and low cost robotic platform for STEM education. *Int. J. Comput. Appl.* **178**, 33–46 (2019)
2. From Wikipedia: Science, technology, engineering, and mathematics. https://en.wikipedia.org/wiki/Science,_technology,_engineering,_and_mathematics
3. Gonzalez, H.B., Kuenzi, J.J.: Science, technology, engineering, and mathematics (STEM): a primer, pp. 1–15. Congressional Research Service (2012)
4. Ioannou, M., Bratitsis, T.: Utilizing Sphero for a speed related STEM activity in Kindergarten. In: Hellenic Conference on Innovating STEM Education. Athens (2016)
5. Tsupros, N., Kohler, R., Hallinen, J.: STEM Education in Southwestern Pennsylvania the missing components (2009)
6. Texley, J., Ruud, R.M.: Teaching STEM Literacy: a Constructivist Approach for Ages 3 to 8. Redleaf Press, St. Paul (2018)
7. Psycharis, S., Kotzampasaki, E.: The impact of a STEM inquiry game learning scenario on computational thinking and computer self-confidence. *Eurasia J. Math. Sci. Technol. Educ.* **15** (2019). <https://doi.org/10.29333/ejmste/103071>
8. Daniela, L., Lytras, M.D.: Educational robotics for inclusive education. *Technol. Knowl. Learn.* **24**, 219–225 (2019). <https://doi.org/10.1007/s10758-018-9397-5>
9. Mavrovounioti, V., Chatzopoulos, A., Papoutsidakis, M., Piromalis, D.: Implementation of an 2-wheel educational platform for STEM applications. *J. Multidiscip. Eng. Sci. Technol.* **5**, 8944–8948 (2018)
10. Melkonian, S., Chatzopoulos, A., Papoutsidakis, M., Piromalis, D.: Remote control via android for a small vehicle' s 2-wheels balancing. *J. Multidiscip. Eng. Sci. Technol.* **5**, 8964–8967 (2018)
11. Xatzopoulos, A., Papoutsidakis, M., Chamilothis, G.: Mobile robotic platforms as educational tools in mechatronics engineering. In: International Scientific Conference eRA – 8, Pireaus, pp. 41–51 (2013)
12. Ospennikova, E., Ershov, M., Iljin, I.: Educational robotics as an innovative educational technology. *Procedia - Soc. Behav. Sci.* **214**, 18–26 (2015). <https://doi.org/10.1016/j.sbspro.2015.11.588>
13. Atmatzidou, S., Markelis, I., Demetriadis, S.: The use of LEGO Mindstorms in elementary and secondary education : game as a way of triggering learning. In: Workshop Proceedings of International Conference on Simulation, Modeling, and Programming for Autonomous Robots, pp. 22–30 (2008)

14. Dorouka, P., Papadakis, S.J., Kalogiannakis, M.: Tablets and apps for promoting robotics, mathematics, STEM education and literacy in early childhood education. *Int. J. Mob. Learn. Organ.* **14**, 255–274 (2020). <https://doi.org/10.1504/IJMLO.2020.10026334>
15. Kalogiannakis, M., Papadakis, S.: Evaluating a course for teaching introductory programming with Scratch to pre-service kindergarten teachers. *Int. J. Technol. Enhanc. Learn.* **11**, 231 (2019). <https://doi.org/10.1504/ijtel.2019.10020447>
16. Kalogiannakis, M., Papadakis, S.: Evaluating pre-service kindergarten teachers' intention to adopt and use tablets into teaching practice for natural sciences. *Int. J. Mob. Learn. Organ.* **13**, 113–127 (2019). <https://doi.org/10.1504/IJMLO.2019.096479>
17. Vamvakopoulos, P.: Event for Educational Robotics, with great interest. <http://proodeutiki.gr/?p=85159>
18. LEGO® Education: Education WeDo Toolbox. 31 (2017)
19. Butcher, N.: A Basic Guide to Open Educational Resources (OER). United Nations Educational, Scientific and Cultural Organization, Paris, France (2015)
20. Priniotakis, G., Chronis, I., Chatzopoulos, A., Louris, E.: Lilipad Arduino platform for a multifunctional garments and wearables laboratory course. In: 15th AUTEX World Textile Conference 2015, Bucharest, pp. 10–13 (2015)
21. Papoutsidakis, M., Tanwar, R., Chatzopoulos, A., Tseles, D.: Custom made embedded automation systems for smart homes - part 2: the implementation. *Int. J. Eng. Appl. Sci. Technol.* **2**, 16–19 (2017)
22. Papoutsidakis, M., Chatzopoulos, A., Piromalis, D.: Distance control of water temperature via android devices. *J. Multidiscip. Eng. Sci. Technol.* **6**, 11240–11244 (2019)
23. Google: Blockly—Google Developers (2017). <https://developers.google.com/blockly/>
24. Vordos, G., Chatzopoulos, A., Papoutsidakis, M., Piromalis, D.: Balance control of a small scale sphere with an innovative android application. *J. Multidiscip. Eng. Sci. Technol.* **5**, 8957–8963 (2018)
25. Papoutsidakis, M., Chatzopoulos, A., Drosos, C., Kalovrektis, K.: An Arduino family controller and its interactions via an intelligent interface. *Int. J. Comput. Appl.* **179**, 5–8 (2018)
26. Pasternak, E., Fenichel, R., Marshall, A.N.: Tips for creating a block language with blockly. In: Proceedings of the 2017 IEEE Blocks Beyond Work. B&B 2017, November 2017, pp. 21–24 (2017). <https://doi.org/10.1109/BLOCKS.2017.8120404>
27. Métayer, O.: Blockly@rduino Create Code with Blocks. <https://create.arduino.cc/projecthub/libreduc/blockly-rduino-create-code-with-blocks-b6d3e4>
28. Alimisis, D.: Educational robotics: Open questions and new challenges. *Themes Sci. Technol. Educ.* **6**, 63–71 (2013)
29. Miller, D.P., Nourbakhsh, I.: Robotics for education. In: Siciliano, B., Khatib, O. (eds.) *Springer Handbook of Robotics*, pp. 2115–2134. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-32552-1_79
30. Sullivan, A., Bers, M.U.: Dancing robots: integrating art, music, and robotics in Singapore's early childhood centers. *Int. J. Technol. Des. Educ.* **28**, 325–346 (2018). <https://doi.org/10.1007/s10798-017-9397-0>