

Gaining Python Skills Through Interactive Education Robot Ozobot EVO[†]

Maya Staikova

Institute of Information and Communication Technology, Bulgarian Academy of Sciences,
1000 Sofia, Bulgaria; ctaukoba@gmail.com; Tel.: +359-889-146428

[†] Presented at the 14th International Scientific Conference TechSys 2025—Engineering, Technology and Systems, Plovdiv, Bulgaria, 15–17 May 2025.

Abstract

This paper explores the potential of the Ozobot EVO mobile robot as an educational tool for teaching Python programming. While the robot is currently designed for younger students through color and block programming, it is not yet widely utilized for teaching text-based coding. The Ozobot's compatibility with Python presents a valuable opportunity for students to visualize their programming concepts through the robot's actions, offering a more engaging alternative to console-based learning. The increasing use of the Raspberry Pi, a single-board computer programmed in Python, has necessitated the inclusion of Python in the curriculum. However, students often find learning Python challenging and demotivating. To enhance STEM education and student motivation, this paper proposes sample Python code for the Ozobot EVO, aiming to encourage educators to integrate the robot into their teaching. I suggest some Python code examples for the Ozobot EVO. This is to help educators see how they can use the robot in their lessons. Specifically, code examples for controlling motion, sound, light, and combinations of these functionalities are presented. When students see the robot react immediately to their code, they can understand programming ideas much better than just seeing text in the Python console. The Ozobot EVO mobile robot offers a solid foundation for learning Python programming.

Keywords: Ozobot; Python; education; mobile robot; code; skills



Academic Editors: Nikola G. Shakev,
Valyo N. Nikolov, Nikolay
R. Kakanakov and Sevil
A. Ahmed-Shieva

Published: 4 July 2025

Citation: Staikova, M. Gaining Python Skills Through Interactive Education Robot Ozobot EVO. *Eng. Proc.* **2025**, *100*, 15. <https://doi.org/10.3390/engproc2025100015>

Copyright: © 2025 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Following the natural progression of technological evolution, we have now entered the epoch known as Industry 4.0, characterized by the integration of the Internet of Things (IoT), Big Data, and artificial intelligence (AI). In recent years, the implementation of AI has escalated, leading to a mass adoption of its technologies. The primary objective of Industry 4.0 is to develop increasingly intelligent manufacturing systems, optimizing the automation of processes to minimize human intervention [1–3]. This evolution coincides with the emergence of new generations who have been raised in environments influenced by both Industry 3.0 and 4.0, thereby acclimatizing them to these technological advancements. To enhance their proficiency in interacting with machines and harnessing their creative potential, it is imperative that these younger generations develop a comprehensive understanding of robotics and machinery from an early age. In response to this need, there has been an increasing emphasis on Science, Technology, Engineering, and Mathematics (STEM) education, with programming curricula being introduced at progressively younger ages. The technology education sector has experienced significant advancements with the

deployment of various robotic systems that facilitate the transition into comprehending the technological landscape. Robotics stands out as one of the most sophisticated domains within STEM, underscoring the necessity for educational initiatives to be integrated into primary and secondary curricula using cost-effective robotic solutions [4,5]. This approach not only promotes technical literacy but also prepares future generations for the upcoming sophisticated automated systems, thereby aligning education with the demands of Industry 4.0 [2,4–7].

To achieve effective educational outcomes, the use of high-quality materials is essential. This article introduces Ozobot EVO, a robot particularly well-suited for integration into the learning environment. Constructed from durable materials, the Ozobot's compact design enhances its practicality for educational setups, making it helpful for both individual and paired student activities. Currently designed for a target age group of 6 to 10 years, the Ozobot EVO also provides opportunities for programming in Python 3, which has been incorporated into the curriculum for sixth- and seventh-grade students.

This article presents a practical lesson conducted in a seventh-grade classroom, where the programming is performed in Python. While students receive output through the program console, this approach may be perceived as unengaging and difficult to comprehend for their developmental stage. However, the inclusion of a tangible robot in the lesson significantly enhances student motivation and engagement. By bridging theoretical programming concepts with a physical robot, educational experiences become more interactive and compelling.

2. Hardware

To effectively program a robotic system, it is very important to understand its hardware architecture and components. This knowledge is helpful as it provides an overview of the robot's functional capabilities and also helps in identifying limitations inherent in its design while proposing the potential for future enhancements. In Figure 1, we have an overview picture of the mini robot for illustrative and size perception purposes.



Figure 1. Mobile robot Ozobot EVO.

As we can see from Figure 1, the dimensions of the robot are notably compact, measuring only 2.5 cm × 2.5 cm and possessing a mass of merely 17 g. Notwithstanding its miniature size, Figures 2 and 3 provide some insights into the features and capabilities of the Ozobot EVO.

The Ozobot EVO is constructed within a transparent polycarbonate shell, featuring a power activation button, charging port, top LED light, five controllable color LED lights, sound system/speaker, Bluetooth, proximity, and four optical sensors (IR) and movement wheels [8].



Figure 2. Top view of Ozobot EVO.

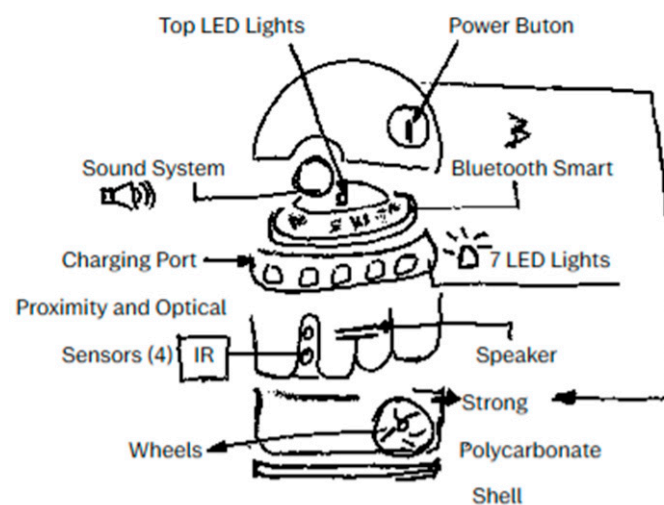


Figure 3. Sensors and Actors.

3. Ozobot EVO Programming

Contemporary education increasingly incorporates methodologies centered on problem-based learning, case studies, and experiential learning frameworks. During early developmental stages, children's engagement with learning often manifests in playful interactions; hence, Ozobot has been specifically designed to accommodate the dexterity of young learners, featuring a compact size that is easy for manipulation by smaller hands while also minimizing spatial requirements within the classroom environment.

Ozobot utilizes a diverse array of programming tools, both device-dependent and device-independent. In the absence of a digital interface, the robot employs color-sensing technology to interpret drawn paths that are simple programs printed in color code that Ozobot can understand. Such color code examples are shown in Figure 4.

At a later stage, when students can move on to using computer-like devices, the Ozobot allows them to work via tablet, phone, or computer. As students learn reading, they can progress to block programming. They will line up pre-made commands, but they will also be able to read them and understand what commands they are using. Figures 5 and 6 show the Ozobot block programming interface. The program is developed in five stages, ranging from pre-reader to master difficulty levels. At school age, students can begin at the beginner level, where, in addition to the basic programs for motion, light, time, and sound, the loop panel is introduced.

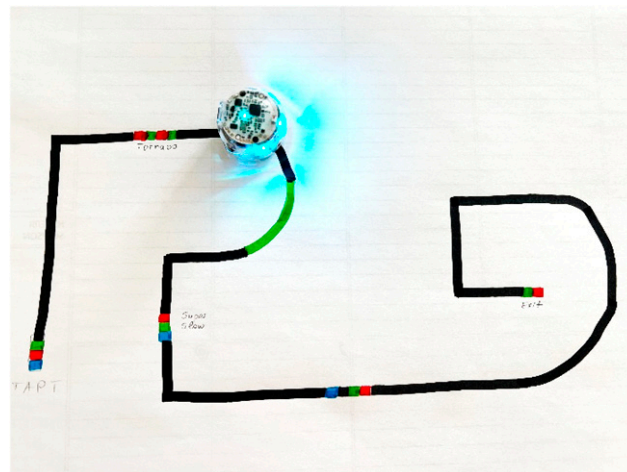


Figure 4. Ozobot color coding.

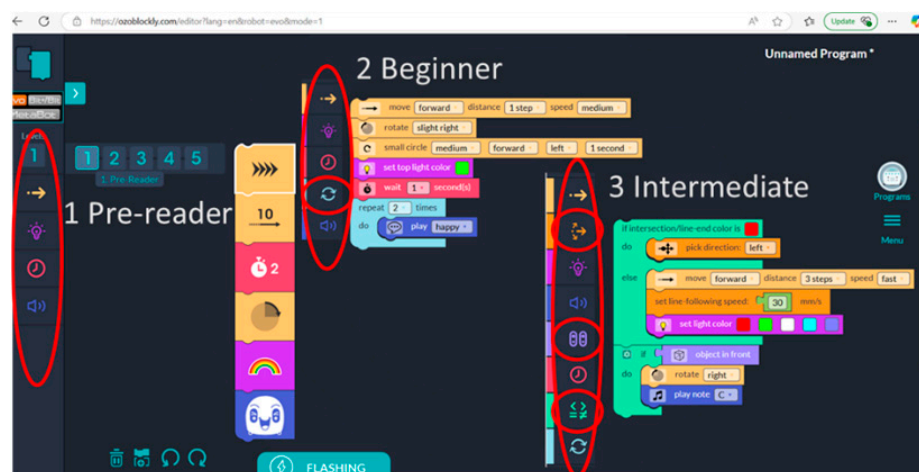


Figure 5. Blockly programming beginner.

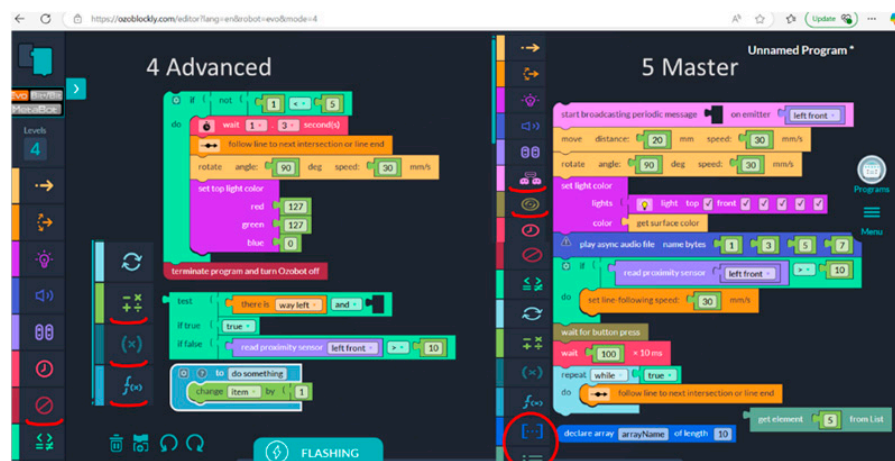


Figure 6. Blockly programming master.

In the next level, intermediate (Level 3), students expand their knowledge by incorporating motion on a line, sensors, and logical conditions (if-then-else). Once they have consolidated their understanding of the new concepts, they can advance to Level 4, where fundamental programming concepts such as math, functions, and variables are covered. By learning the commands through block programming and applying them to the Ozobot,

students can see the results of their programming and understand what each block does. This stage is crucial for grasping the underlying concepts. The connection between programming and the robot's execution increases motivation and encourages reflection on what can be improved. When students observe real implementation, they gain a deeper understanding of what they are creating.

Once students master this interrelationship, learn a set of commands, and develop spatial orientation and abstract thinking—typically around the fifth to sixth grade when they are 12 to 13 years old—they can progress to the actual programming language Python. At this stage, they enter a programming environment where they can write code. This is when they are introduced to libraries, concepts like arrays, and other essential topics, which they will learn gradually. As they grow older, they will also acquire new knowledge in mathematics, which is a fundamental aspect of logical thinking in programming. For these higher-grade students who are familiar with programming languages and coding, a Python library for Ozobot is provided, which can be used for programming to create various, more complex programs for controlling the robots' actions and reading sensors. This library can be easily utilized by students for advanced control and interaction with the robot.

Given its compact size, the Ozobot is suitable for individual or paired activities, facilitating hands-on programming experiences that provide tangible feedback on programmed tasks. This interactive learning tool serves to enhance students' comprehension of technological concepts, elucidating the mechanics of programming and the principles underpinning connectivity in technology. Moreover, Ozobots can be employed as pedagogical instruments to elucidate fundamental concepts of AI, thereby enriching students' understanding of this increasingly relevant field.

4. Challenges

In recent years, the incorporation of the Python programming language into educational curricula has gained significant momentum. One of the educational topics for computer modeling and information technology for seventh-grade students encompasses a series of lessons in Python programming that enhance the foundational knowledge acquired in sixth grade. Presently, trinket.io/python/ is utilized as the platform for code development at school.

In sixth grade, students are introduced to the Python programming language, where they execute a limited number of basic commands. By contrast, seventh-grade instruction facilitates a transition from block-based programming to text-based programming, wherein students engage with fundamental constructs such as loops, functions, and various programming commands. This stage of instruction presents greater complexity in defining and utilizing functions, as students must navigate the intricacies of parameter definition, including data types like integers, floats, and doubles.

Furthermore, students are expected to apply these programming concepts across a range of project-based tasks, which allows for interdisciplinary connections, particularly with software such as Microsoft Excel, thereby deepening their comprehension of programming principles. Despite these efforts, some students continue to struggle with conceptual understanding.

5. Dancing Robot Class Example

To address these challenges, a sample class example is proposed with specific Python code snippets that are compatible with the Ozobot robot, enabling the execution of programmed tasks. One such tutorial involves the development of a motion program designed to replicate the movement of a Thracian choro dance (Buchimis choro). The programming

code necessary for this task is algorithmically structured to create a star-shaped pattern. The corresponding tutorial code is illustrated in Figure 7, while the proposed enhanced code for implementation utilizing Ozobot with its movement capabilities is shown in Figure 8.

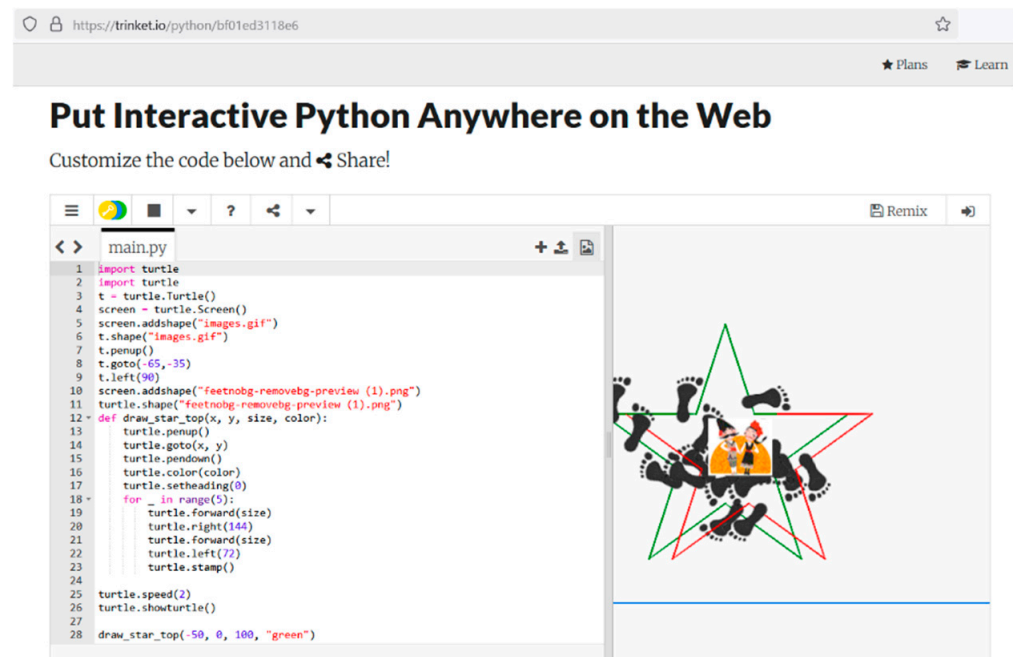


Figure 7. Code for the Thracian choro in Trinket Python.

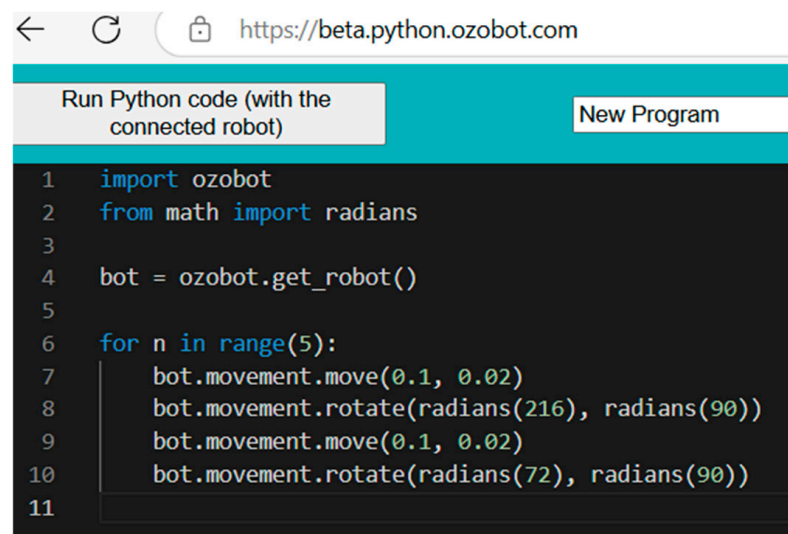


Figure 8. Code for the Thracian choro in Ozobot in Python.

In this context, in addition to physical movement, the student's exploration of computational concepts can be further enriched by integrating musical coding and the implementation of algorithms to generate a Thracian horo–Svornato horo (9/8). For this task, it is essential to identify the musical notes corresponding to the horo and subsequently encode their phonetic representation in a systematic manner. As shown in Figure 9, part of this initiative represents an independent project, as illustrated in Figure 10.



Figure 9. Notes of Svornato horo.

```
1 import ozobot
2
3
4 bot = ozobot.get_robot()
5 takt = 1.125
6 #sound = ozobot.get_robot().sound
7
8 def phrase():
9     bot.sounds.play_note(5, ozobot.Note.G, takt/4)
10    bot.sounds.play_note(5, ozobot.Note.F, takt/8)
11    bot.sounds.play_note(5, ozobot.Note.E, takt/8)
12    bot.sounds.play_note(5, ozobot.Note.G, takt/4)
13    bot.sounds.play_note(5, ozobot.Note.D, takt/4)
14    bot.sounds.play_note(5, ozobot.Note.C, takt/8)
15
16 for count in range(2):
17     phrase()
18
```

Figure 10. Code for music in Svornato horo (9/8) in Ozobot Python.

As an integral component of this project, the subsequent phase involves programming the Ozobot EVO to simulate the behavior of a dancing couple of figures engaged in choir construction and performance. This entails programming the robot to navigate in a star-shaped trajectory while concurrently producing auditory output in the form of a melody reminiscent of a Thracian horo. Relevant code snippets illustrating this functionality are presented in Figures 11 and 12.

```
1 import ozobot
2 import asyncio
3 from math import radians
4
5 bot = ozobot.get_robot(coro = True)
6 movement = bot.movement
7
8 takt = 1.125
9 #sound = ozobot.get_robot().sound
10
11 async def phrase(repeat):
12     for n in range(repeat):
13         await bot.sounds.aplay_note(5, ozobot.Note.G, takt/4)
14         await bot.sounds.aplay_note(5, ozobot.Note.F, takt/8)
15         await bot.sounds.aplay_note(5, ozobot.Note.E, takt/8)
```

Figure 11. Ozobot Python Code for music in Svornato horo (9/8).

```

39 ✓ async def moveLikeAStar(repeat):
40 ✓     for n in range(repeat):
41         await movement.move(0.1, 0.02)
42         await movement.arotate(radians(-(180-34)), radians(90))
43         await movement.move(0.1, 0.02)
44         await movement.arotate(radians(72), radians(90))
45
46
47 ✓ async def main():
48     res = await asyncio.gather(phrase(9), moveLikeAStar(5))
49
50     asyncio.run(main())
51

```

Figure 12. Code for music in Svornato horo (9/8) in Ozobot Python with Async futures.

Moreover, it is essential to employ asynchronous programming paradigms for this project. Initially proposed implementations utilized synchronous coding techniques; however, the current iteration necessitates the utilization of asynchronous libraries to facilitate the desired concurrent operations.

To illustrate the star, a colored pencil is attached to see the movement of the robot according to the set code Figure 13.

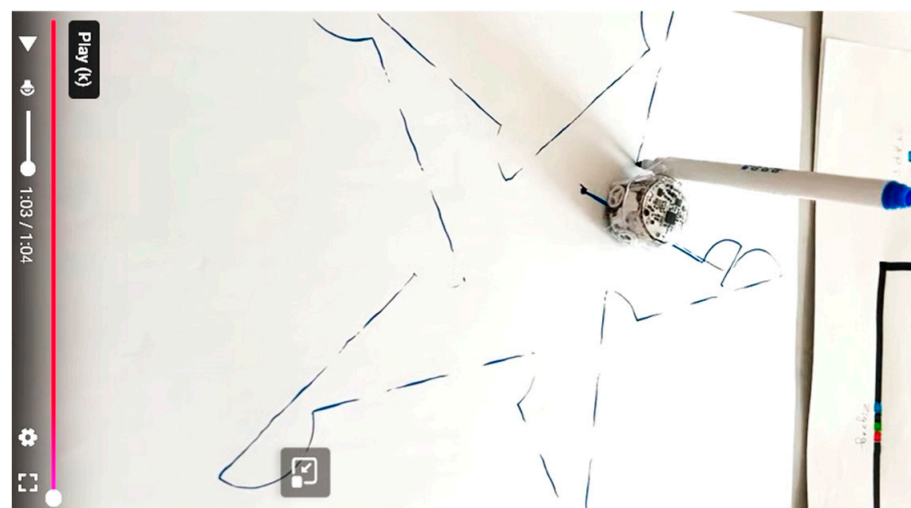


Figure 13. Draw a Thracian choro with Ozobot EVO.

This graphical representation may be pre-calculated, allowing for the subsequent formulation of the controlling code.

6. Conclusions

Ozobot EVO is a compact mobile robotic device that seamlessly integrates into educational curricula across a diverse age range, accommodating students from 6 to 18 children years. Its design facilitates individual or collaborative use among learners, promoting interactive engagement. The robot's battery can be recharged in approximately 20 min, providing an operational duration of 45 to 50 min, sufficient for a standard instructional period. The minimalistic dimensions of Ozobot EVO enable the utilization of multiple units within classroom environments, minimizing spatial constraints while enhancing the potential for hands-on learning experiences.

Its small size is directly related to the materials from which it is constructed, as well as its sustainability profile. However, it demonstrates a notable sensitivity to the substrate upon which it traverses. This phenomenon is particularly evident when a writing implement, such as a pencil, is affixed to it.

Ozobot EVO provides significant flexibility in learning and application, facilitating tasks that range from interpreting color codes to engaging in block-based programming. This progression is structured into five distinct stages, culminating in advanced programming using the Python language.

Ozobot serves as a great tool for enhancing programming education through a block-based coding system, particularly for students at a developmental stage where they seek to comprehend the implications of their commands and the subsequent outcomes of their implementations. While block programming inherently facilitates a visual representation of code and elucidates its underlying rationale, the introduction of an external element—such as a mobile robot—significantly deepens students' grasp of programming principles, logic, and mathematical concepts. This hands-on interaction not only reinforces cognitive integration but also enhances students' motivation to explore further knowledge in these fields. Consequently, this relationship between humans and robotics fosters a robust foundation for the emergence of Industry 5.0, characterized by increased collaboration between human intellect and robotic capabilities in future development.

Numerous tutorials have already been crafted focusing on color coding and device-independent programming; however, there exists an equally significant interest in utilizing devices and implementing programs through robotics.

When engaging with Python, familiarity with the Ozobot EVO platform serves as a foundational prerequisite, enabling learners to discern the distinctions and commonalities among various interfaces associated with Python. This experience enhances comprehension of programming concepts such as libraries, arrays, functions, and the nuances of synchronous versus asynchronous programming.

Despite the limited hours allocated within the syllabus for IT, the integration of Ozobot EVO and Python programming could be synergistically combined with diverse educational subjects, including music, dance, and mathematics, thereby facilitating a broader application within STEM (Science, Technology, Engineering, and Mathematics) curricula.

Funding: This research received is funding from doctoral subsidies.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
AI	artificial intelligence
STEM	Science, Technology, Engineering, and Mathematics

References

1. Huang, S.; Wang, B.; Li, X.; Zheng, P.; Mourtzis, D.; Wang, L. Industry 5.0 and Society 5.0—Comparison, complementation and co-evolution. *J. Manuf. Syst.* **2022**, *64*, 424–428. [CrossRef]
2. Staikova, M.; Ivanova, V.; Chivarov, N. Artie max robot as a STEM education tool through integration technology, math and english language. *AIP Conf. Proc.* **2024**, 3078, 020007. [CrossRef]
3. Staikova, M.; Ivanova, V.; Chivarov, N. Students understanding for AI in different educational levels. *IFAC-PapersOnLine* **2024**, *58*, 182–186. [CrossRef]
4. Khine, M.S.; Ohmer. *Robotics in STEM Education*; Springer: Berlin, Germany, 2017; p. 262.
5. Breiner, J.M.; Harkness, S.S.; Johnson, C.C.; Koehler, C.M. What Is STEM? A Discussion About Conceptions of STEM in Education and Partnerships. *Sch. Sci. Math.* **2012**, *112*, 3–11. [CrossRef]
6. Vega, J.; Canas, J.M. Open vision system for low-cost robotics education. *Electronics* **2019**, *8*, 1295. [CrossRef]
7. Weinberg, J.B.; Yu, X. Robotics in education: Low-cost platforms for teaching integrated systems. *IEEE Robot. Autom. Mag.* **2003**, *10*, 4–6. [CrossRef]
8. Ozobot Educator's Guide. Available online: <https://ozobot.com> (accessed on 17 May 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.