LUMEN
ET VIRTUS

Programming logic through robotics: Use of Scratch and Arduino for robot creation and interactive projects

https://doi.org/10.56238/levv15n39-61

Eduka Bytes

ABSTRACT

This article investigates the teaching of programming logic through robotics, using Scratch and Arduino for the creation of robots and interactive projects. Through a literature review, we examine the advantages and challenges of this approach, highlighting how the integration of visual programming tools and hardware can enrich student learning. The integration of Scratch and Arduino in education emerges as an innovative approach to teaching programming logic. Scratch, a visual programming language, and Arduino, an open-source electronics platform, provide students with a dynamic and interactive learning environment. Through hands-on projects and real-time feedback, students can explore abstract concepts in a tangible way, enhancing their understanding and knowledge retention. Despite the challenges, such as teacher training and the availability of resources, the benefits of using Scratch and Arduino outweigh the difficulties, providing students with engaging and meaningful learning experiences. This article reviews the literature on the use of Scratch and Arduino in education, highlighting their contributions to programming education and project-based learning. The results indicate that the use of Scratch and Arduino promotes a dynamic and engaging learning environment, facilitating the understanding of programming and electronics concepts. Future research should focus on strategies to improve teacher education and expand access to these tools, ensuring that all students have the opportunity to develop essential digital literacy skills. By empowering students to become creators and innovators, Scratch and Arduino pave the way for a future where technology is not only understood, but is also actively shaped and utilized for positive change.

Keywords: Scratch, Arduino, Programming Education, Project-Based Learning, Teacher Training.

INTRODUCTION

The growing importance of programming skills in today's world has driven the search for innovative and effective teaching methods. The teaching of programming is considered essential for the development of twenty-first century skills, such as critical thinking, problem solving and creativity (Wing, 2006). In this context, educational robotics, using platforms such as *Scratch* and Arduino, emerges as a promising solution to engage students and facilitate the learning of programming logic.

Scratch is a visual programming language developed by the *MIT Media Lab*, which allows users to create programs by assembling blocks, making it easier to understand fundamental programming concepts (Resnick et al., 2009). Arduino, on the other hand, is an open-source electronic prototyping platform that allows the creation of interactive projects by combining *software* and *hardware* (Banzi & Shiloh, 2014). The combination of these two tools makes it possible to build robots and interactive projects, providing practical and contextualized learning.

This article reviews the existing literature on the use of *Scratch* and Arduino in education, highlighting their contributions to the development of interactive projects and robots. The practical and visual approach offered by these tools can facilitate the understanding of complex programming and electronics concepts, as well as promote active and engaging learning.

JUSTIFICATION

The integration of *Scratch* and Arduino in programming education offers a practical and visual approach, which can make complex concepts easier to understand. *Project-Based* Learning (PBL), especially in robotics, has been shown to be effective in increasing students' interest and improving their *problem-solving* skills (Krajcik & Blumenfeld, 2006). Studies (QUAL???) indicate that educational robotics promotes active learning, where students apply theoretical knowledge in practical contexts, developing essential skills for the twenty-first century (Mataric et al., 2007).

Additionally, educational robotics can help overcome the abstraction barrier often associated with teaching programming by making concepts more tangible and understandable. By seeing their codes come to life in physical robots, students are able to better visualize and understand the consequences of their programmatic actions (Bers, 2010). This is particularly important for young students, as they may feel unmotivated in the face of traditional programming teaching methods, which rely heavily on abstract theoretical concepts.

Thus, this study is justified by the need to identify effective pedagogical practices that prepare students for an increasingly technological future. Through the literature review, we seek to understand how *Scratch* and Arduino can be effectively integrated into the school curriculum, promoting not only the acquisition of technical skills, but also the development of socio-emotional skills, such as collaboration, communication, and persistence.

METHODOLOGY

This article adopts a bibliographic approach, based on a systematic review of articles, books and reports on the use of *Scratch* and Arduino in Education. The search was carried out in academic databases, including publications from the last ten years (2004-2024), to ensure the relevance and timeliness of the data. The inclusion criteria involved studies that address the application of Scratch and Arduino in the creation of robots and interactive projects in educational contexts.

LITERATURE REVIEW

ADVANTAGES OF USING SCRATCH AND ARDUINO

Scratch is a visual programming language developed by the *MIT Media Lab*, which allows users to create interactive stories, games, and animations by assembling blocks. This approach is widely used in learning programming concepts due to its intuitive interface and ease of use, which eliminates the complexity associated with textual syntax (Resnick *et al.*, 2009). By abstracting syntax, Scratch allows students to focus on the logic and flow of programs, making programming more accessible, especially for beginners and children (Maloney et al., 2010).

Arduino, on the other hand, is an open-source electronic prototyping platform that simplifies and facilitates the creation of electronic designs. With its combination of easy-to-use *hardware* and *software* , Arduino empowers users to build interactive devices that can, for example, sense and control the physical environment (Banzi & Shiloh, 2014). The open nature of the platform also fosters innovation and collaboration, with a vast community of developers and educators sharing projects and resources online (Kushner, 2011).

The combined use of these tools offers a powerful approach to teaching programming logic and electronics. *Scratch for Arduino* (S4A) is an extension that allows you to program Arduino boards using the Scratch graphical interface, facilitating the integration between programming and electronics in a unified learning environment (Mellis et al., 2012). This integration promotes hands-on, interactive learning in which students can visualize the impact of their code in real-time through physical devices such as LEDs, motors, and sensors.

Studies show that this approach increases student motivation and engagement. For example, Martins et al. (2016) highlight that robotics projects based on *Scratch* and Arduino not only make learning programming more fun but also help students develop problem-solving and critical thinking skills. Students are challenged to apply theory in practical contexts, which can lead to a deeper understanding of concepts and greater knowledge retention (Papert, 1980).

In addition, project-based learning, facilitated by the use of *Scratch* and Arduino, fosters a collaborative environment where students work as a team to solve problems and create projects. This type of collaborative learning is beneficial for the development of social and communication skills, which are essential for success in the modern workplace (Barron & Darling-Hammond, 2008). The ability to work effectively in groups and to communicate ideas clearly and concisely is often mentioned as one of the key skills required by employers (Trilling & Fadel, 2009).

Another significant benefit is the ability to tailor projects to students' individual needs and interests. Personalization of learning, enabled by the use of flexible tools such as *Scratch* and Arduino, can increase students' intrinsic motivation and encourage them to explore and experiment independently (Deci & Ryan, 2000). This autonomy in learning is crucial for developing a growth mindset and a proactive approach to problem-solving (Dweck, 2006).

Finally, early introduction to programming and robotics can positively influence students' career choices, encouraging more young people to pursue careers in STEM (Science, Technology, Engineering, and Mathematics). Educational programs that utilize Scratch and Arduino have been shown to be effective in arousing students' interest in these areas, especially among underrepresented groups such as girls and minorities (Margolis & Fisher, 2002). This is crucial for diversifying the field of technology and ensuring that a broader spectrum of perspectives and experiences are represented in the future workforce (Cheryan et al., 2017).

CHALLENGES IN IMPLEMENTATION

Despite the benefits of *Scratch* and Arduino in teaching programming and robotics, implementing these tools in the classroom faces several challenges. One of the main obstacles is the need for adequate training for teachers and the availability of material resources. Many educators have no prior experience with these tools, which can limit their effectiveness in conveying the concepts to students (Smith, 2018). Continuous and specialized training of teachers is essential to ensure that they feel confident and empowered to integrate these technologies into their pedagogical practices (Koehler & Mishra, 2009).

The complexity of technology can be intimidating for educators who don't have a technical *background*. The lack of technical knowledge can lead to resistance to the adoption of new technological tools (Ertmer, 1999). To overcome this barrier, it is essential for schools to invest in professional development programs that offer hands-on training and ongoing support to teachers. These programs should include not only the use of the tools, but also pedagogical methodologies that promote active and collaborative learning (Darling-Hammond et al., 2017).

Another significant challenge is the availability of material resources. Implementing projects with *Scratch* and Arduino requires an initial investment in *hardware*, such as Arduino boards, sensors, actuators, and other electronic components, as well as computers with *proper software* installed. In many schools, especially in regions with fewer resources, this financial barrier can be a substantial impediment (Becker et al., 2016). In addition to the initial costs, there is the need for maintenance and replacement of components, which can generate ongoing expenses for educational institutions (Balanskat & Engelhardt, 2015).

The technological infrastructure of schools can also be a limiting factor. In this way, projects with *Scratch* and Arduino are successful, it is necessary that schools have adequate computer labs, with sufficient computers and quality internet connectivity. In many schools, especially in rural and underfunded areas, this basic infrastructure is not yet available (OECD, 2015).

School culture and resistance to change can also represent barriers. The integration of new technologies often requires a significant change in pedagogical practices and curriculum organization. Teachers and administrators may be reluctant to adopt new methodologies that are different from the traditional approaches with which they are familiar (Fullan, 2007). To mitigate this resistance, it is crucial to involve all stakeholders in the change process, providing a clear vision of the benefits and providing support during the transition (Hall & Hord, 2015).

In addition, the assessment of learning in robotics and programming projects presents its own challenges. Traditional assessment methodologies, focused on tests and exams, may not effectively capture the skills developed through practical and interactive projects. It is necessary to develop new assessment methods that consider skills such as problem-solving, creativity, collaboration, and critical thinking (Gibson & Clarke, 2010). Authentic assessment, which includes portfolios, presentations, and self-assessment, may be best suited for measuring the impact of project-based learning with *Scratch* and Arduino (Wiggins, 2019)).

Finally, inequality in access to technology can exacerbate educational disparities. While some students have access to technological resources at home and at school, others may not have the same opportunity, which creates a digital divide (Warschauer, 2004). Thus, schools should adopt policies that ensure equitable access to technological tools and resources for all students, regardless of their socioeconomic status (Selwyn, 2011).

EXAMPLES OF INTERACTIVE PROJECTS AND ROBOTS

Several studies document educational projects that use *Scratch* and Arduino to teach programming and engineering concepts in a practical way. These projects offer students the opportunity to apply theories to concrete practices, facilitating the understanding of complex concepts through experimentation and hands-on learning, as well as promoting essential skills such as problem-solving, critical thinking, and creativity (Blikstein, 2013).

For example, in one high school, students developed robots that could follow lines and avoid obstacles by applying sensor and control concepts (Garcia & Gonzalez, 2020). In this project, students used infrared sensors to detect lines and paths and ultrasonic sensors to identify and avoid obstacles. Students connected these sensors to Arduino boards programmed with *Scratch for Arduino* (S4A). The use of sensors teaches principles of *feedback* and control, which are fundamental in many engineering and computing applications (Bateson, 2001). Therefore, this type of project allows students to understand the logic of flow control and the integration of *hardware* and *software*, which are fundamental in robotics and automatic systems (Garcia & Gonzalez, 2020). Additionally, projects like this help develop problem-solving and logical thinking skills, as students must continuously adjust their codes and sensors to optimize the robot's performance (Eguchi, 2014).

Another project analyzed involved the creation of interactive games that responded to physical commands through sensors connected to the Arduino (Johnson & Lee, 2017). Students created games in which in-game actions were controlled by physical inputs, such as buttons and motion sensors. For example, a simple maze game was controlled by tilting a platform equipped with tilt sensors, allowing players to move a virtual ball (Johnson & Lee, 2017). This type of project illustrates how programming can be used to create intuitive physical interfaces, combining elements of game *design* with electronic engineering (Kafai & Burke, 2014). Therefore, this integration between *software* and *hardware* exemplifies the importance of understanding the interface between the digital and physical worlds, a skill that is increasingly relevant in the era of the Internet of Things (IoT) (Atzori, Iera, & Morabito, 2010).

In addition to robots and games, more complex projects may include home automation or home automation systems, in which students program Arduino boards to control lights, fans, and other household appliances through presence sensors and timers (Martin et al., 2016). There are also projects that involve the creation of home automation systems using Arduino and *Scratch*. Students in a high school project developed a system that automates lighting and temperature control in a residential model, using light and temperature sensors to automatically adjust the lights and thermostat (Smith & Thompson, 2019).

These projects not only teach programming and electronics, but also introduce students to the principles of energy efficiency and home automation, areas of increasing importance in sustainable development (Harb, 2018). It also introduces Internet of Things (IoT) concepts, demonstrating how different devices can be connected and controlled remotely (Ashton, 2009).

Hands-on experience with IoT can prepare students for future careers in a field that is rapidly expanding (Greengard, 2015).

In another study, students created autonomous vehicles that could make decisions based on sensory data, such as following a predetermined trajectory or stopping in the face of an unexpected obstacle (Martinez & Stager, 2013). This type of project is particularly useful for teaching navigation and decision-making algorithms, as well as basic artificial intelligence concepts. Through the programming of these vehicles, students learn the importance of conditional logic and real-time signal processing, which are essential in the fields of engineering and computer science (Siegwart et al., 2011).

Another example of a project involves the creation of environmental monitoring systems. Students utilized temperature, humidity, and air quality sensors connected to the Arduino to collect environmental data, which is then visualized through interfaces programmed into *Scratch* (Smith et al., 2018). This example project proves effective in teaching students about data science, data collection and analysis, as well as promoting awareness of environmental issues (Starkweather, 2014).

Such project examples illustrate how the combination of *Scratch* and Arduino can be used to create meaningful learning experiences. The practical approach and interactivity of these projects facilitate the understanding of theoretical concepts, in addition to promoting creativity, collaboration and engagement of students. In addition, by working on projects that have practical applications and real-world relevance, students have the possibility to see the direct impact of their learning, which can increase motivation and interest in STEM (Science, Technology, Engineering, and Mathematics) subjects (Honey et al., 2014).

In addition to the direct educational benefits, these projects provide opportunities for interdisciplinarity, integrating knowledge from various disciplines, such as Science, Mathematics and Technology. By working on projects that combine programming and electronics with real-world applications, students are encouraged to apply theoretical knowledge in practical contexts, reinforcing their learning and increasing their motivation (Hmelo-Silver, 2004).

DISCUSSION

Analysis of the literature reveals that the integration of *Scratch* and Arduino can significantly transform the teaching of programming logic, making it more accessible and effective for students. By offering a practical and interactive approach, these tools make it possible for students to experience abstract concepts in a tangible context, which can facilitate understanding and knowledge retention (Papert, 1980).

Scratch, with its intuitive graphical interface, facilitates the introduction of programming concepts without the complexity of textual syntax (Resnick *et al*., 2009). This approach allows students to focus on developing computational thinking and problem-solving creatively and collaboratively (Brennan & Resnick, 2012).

The Arduino platform complements this experience by connecting the digital world to the physical, allowing students to see the tangible results of their code in action (Banzi & Shiloh, 2014). The combination of these tools promotes experiential and practical learning, which is fundamental for the internalization of abstract concepts, such as algorithms and control structures (Papert, 1980). Studies indicate that Arduino-based projects not only improve students' technical understanding but also encourage the development of practical and engineering skills (Blikstein, 2013).

The project-based approach, facilitated by the use of *Scratch* and Arduino, promotes more meaningful learning, in which students not only absorb information but also apply and build their knowledge in real-world situations (Krajcik et al., 2008). This type of active and constructivist learning is more aligned with the needs and characteristics of students of the current generation, who are used to interacting with technology from an early age (Prensky, 2001).

However, the effective implementation of this approach depends heavily on adequate teacher training and the availability of resources. As mentioned by Koehler & Mishra (2009), educators' lack of prior experience with these technologies can be a significant barrier. Research shows that many teachers feel insecure when using new technological tools due to the lack of specific and continuous training (Ertmer & Ottenbreit-Leftwich, 2010). To overcome this challenge, it is necessary for educational policies to prioritize professional development programs that offer practical training and ongoing support. These programs should address the technical use of the tools, as well as include pedagogical strategies that integrate technology in a meaningful way into the curriculum (Darling-Hammond et al., 2017).

Therefore, continuous professional development programs, which offer specialized training and support, are key to empowering teachers to effectively use these tools in the classroom (Darling-Hammond et al., 2017).

In addition to teacher training, the availability of material resources is also an important issue to be addressed. Implementing projects with *Scratch* and Arduino requires investments in *hardware* and *software*, as well as adequate infrastructure, such as equipped computer labs and internet connectivity (Becker et al., 2016). In many schools, especially in rural and underfunded areas, this infrastructure is still inadequate, which makes it impossible to implement projects with *Scratch* and Arduino (OECD, 2015). Public and private investments are needed to ensure that all schools have access to the resources they need to implement this approach effectively. Thus, it is imperative that education policies and government initiatives prioritize investment in educational technology and ensure equitable access to technological resources in all schools (Balanskat & Engelhardt, 2015).

However, it is important to note that simply providing technology is not enough. It is also necessary to develop and implement effective pedagogical strategies that integrate these tools meaningfully into the school curriculum (Ertmer, 1999). Educators should be encouraged to adopt innovative pedagogical practices that promote student creativity, collaboration, and critical thinking, rather than focusing exclusively on content delivery (Fullan, 2007).

Finally, it is essential that educational policies encourage inclusion and equity in access to technology. Disparity in access to technological tools can exacerbate existing educational inequalities (Warschauer, 2004). Thus, schools should adopt strategies to ensure that all students, regardless of their socioeconomic status, have the opportunity to learn and benefit from these technologies (Selwyn, 2011). This includes the provision of devices, high-quality internet access, and the technical support necessary for the effective implementation of projects.

Another important aspect is the need to adapt evaluation methodologies. Project-based learning, such as the creation of robots and interactive systems, develops skills that go beyond theoretical knowledge, including creativity, collaboration, and complex problem-solving (Wiggins, 2019). Traditional assessment methods, such as tests and exams, may not adequately capture these competencies. Therefore, it is necessary to develop and implement authentic assessment methods that better reflect students' practical skills and critical thinking (Gibson & Clarke, 2010).

In short, the integration of *Scratch* and Arduino in the teaching of programming logic represents an exciting opportunity to transform education and prepare students for the challenges of the twenty-first century. However, for this approach to reach its full potential, a collective commitment from educators, managers, policymakers, and education communities is needed to provide the support and resources needed for its successful implementation.

CONCLUSION

The combined use of *Scratch* and Arduino in education represents an innovative and promising approach to teaching programming logic, providing students with an engaging and meaningful learning experience. By integrating visual and practical elements, these tools make abstract concepts more tangible and accessible, promoting a deeper and more lasting understanding by students (Resnick *et al*., 2009; Banzi & Shiloh, 2014).

As discussed throughout this article, using *Scratch* and Arduino offers a number of advantages, including an intuitive interface, the opportunity for real-time hands-on learning, and the promotion of skills such as problem-solving, critical thinking, and collaboration (Resnick et al., 2009; Banzi & Shiloh, 2014). Studies demonstrate that this approach can increase student motivation and improve engagement in STEM subjects (Martins et al., 2016).

While there are challenges in implementation, such as teacher training and the availability of resources, the advantages offered by this approach far outweigh the difficulties. Scratch and Arduino-based projects provide a dynamic and interactive learning environment where students are encouraged to explore, experiment, and collaborate, developing essential skills for the twenty-first century, such as problem-solving, critical thinking, and creativity (Blikstein, 2013; Krajcik et al., 2008). Thus, it is critical that educators, managers, policymakers, and educational communities work together to overcome these obstacles and harness the full potential of these innovative tools.

To this end, future research should focus on developing effective strategies to improve the training of educators in this area and expand access to these tools in different educational contexts. Additionally, it is important to allocate resources for investments in continuing professional development programs and educational policies that prioritize the integration of technology into the curriculum can help overcome existing challenges and ensure that all students have access to equitable and high-quality learning opportunities (Darling-Hammond et al., 2017; OECD, 2015).

Ultimately, the use of *Scratch* and Arduino in education not only prepares students for the challenges of the twenty-first century but also empowers them to become creators and innovators in an increasingly technological and interconnected world.

REFERENCES

1. Ashton, Kevin. That Internet of Things thing. *RFID Journal*, v. 22, n. 7, p. 97-114, 2009. Available at: http://www.rfidjournal.com/article/view/4986. Accessed on: 15 Feb. 2024.

2. Atzori, Luigi; Iera, Antonio; Morabito, Giacomo. The internet of things: A survey. *Computer Networks*, v. 54, n. 15, p. 2787-2805, Oct. 2010. Available at: https://www.researchgate.net/publication/222571757_The_Internet_of_Things_A_Survey. Accessed on: 15 Feb. 2024.

3. Balanskat, Anja; Engelhardt, Katja. *Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe*. European Schoolnet, 2015. Available at: http://www.eun.org/documents/411753/817341/Computing+our+future_final_2015.pdf/d3780a64-1081-4488-8549-6033200e3c03. Accessed on: 15 Feb. 2024.

4. Banzi, Massimo; Shiloh, Michael. *Make: getting started with Arduino*. 3rd ed. USA: Maker Media, Inc., 2014. Available at: https://www.esc19.net/cms/lib/TX01933775/Centricity/Domain/110/make_gettingstartedwitharduino_3rdedition.pdf. Accessed on: 15 Feb. 2024.

5. Barron, Brigid; Darling-Hammond, Linda. *Teaching for meaningful learning: A review of research on inquiry-based and cooperative learning*. John Wiley & Sons, 2008. Available at: https://files.eric.ed.gov/fulltext/ED539399.pdf. Accessed on: 15 Feb. 2024.

6. Bateson, Robert N. *Introduction to Control System Technology*. Prentice Hall, 2001.

7. Becker, Adams; Freeman, A.; Giesinger Hall, C.; Cummins, M.; Yuhnke, B. *The NMC/CoSN Horizon Report: 2016 K-12 Edition*. Austin, Texas: The New Media Consortium, 2016. Available at: https://files.eric.ed.gov/fulltext/ED570463.pdf. Accessed on: 15 Feb. 2024.

8. Bers, Marina U. *The tangibleK Robotics Program: Applied Computational Thinking for Young Children*. Early Childhood Research & Practice, v. 12, n. 2, p. 1-20, 2010. Available at: https://files.eric.ed.gov/fulltext/EJ910910.pdf. Accessed on: 15 Feb. 2024.

9. Blikstein, Paulo. *Digital fabrication and 'making' in education: The democratization of invention*. Community and Environment, p. 203-221, 2013. Available at: https://www.researchgate.net/publication/281495128_Digital_Fabrication_and_'Making'_in_Education_The_The_Democratization_of_Invention. Accessed on: 15 Feb. 2024.

10. Brennan, Karen; Resnick, Mitchel. *New frameworks for studying and assessing the development of computational thinking*. Annual American Educational Research Association, Vancouver, BC, Canada, 2012. Available at: https://people.cs.vt.edu/~kafura/CS6604/Papers/Framework-Assessing-CT.pdf. Accessed on: 15 Feb. 2024.

11. Cheryan, Sapna; Ziegler, Sianna A.; Montoya, Amanda K.; Jiang, Lily. Why Are Some STEM Fields More Gender Balanced Than Others? *Psychological Bulletin*, v. 143, n. 1, p. 1-35, 2017. Available at: https://research.chicagobooth.edu/-/media/research/cdr/docs/cheryan-paper-1. Accessed on: 15 Feb. 2024.

12. Darling-Hammond, Linda; Hyler, Maria E.; Gardner, Madelyn. *Effective Teacher Professional Development*. Palo Alto, CA: Learning Policy Institute, 2017. Available at: https://learningpolicyinstitute.org/sites/default/files/product-files/Effective_Teacher_Professional_Development_REPORT.pdf. Accessed on: 15 Feb. 2024.

13. Deci, Edward L.; Ryan, Richard M. The "What" and "Why" of Goal Pursuits: Human Needs and the Self-Determination of Behavior. *Psychological Inquiry*, v. 11, n. 4, p. 227-268, 2000. Available at: https://web.archive.org/web/20170810210258id_/http://academic.udayton.edu/jackbauer/Readings%20595/Deci%2000%20goals%20SDT.pdf. Accessed on: 15 Feb. 2024.

14. Dweck, Carol S. *Mindset: The New Psychology of Success*. New York: Random House, 2006. Available at: https://adrvantage.com/wp-content/uploads/2023/02/Mindset-The-New-Psychology-of-Success-Dweck.pdf. Accessed on: 15 Feb. 2024.

15. Eguchi, Amy. Educational robotics for promoting 21st century skills. *Journal of Automation, Mobile Robotics & Intelligent Systems*, v. 8, n. 1, p. 5-11, Jan. 2014. Available at: https://www.researchgate.net/publication/274882640_Educational_Robotics_for_Promoting_21st_Century_Skills. Accessed on: 15 Feb. 2024.

16. Ertmer, Peggy; Ottenbreit-Leftwich, Anee. Teacher technology change: How knowledge, confidence, beliefs, and culture intersect. *Journal of Research on Technology in Education*, v. 42, n. 3, p. 255-284, Mar. 2010. Available at: https://www.researchgate.net/publication/272007146_Teacher_Technology_Change_How_Knowledge_Beliefs_and_Culture_Intersect. Accessed on: 15 Feb. 2024.

17. Ertmer, Peggy A. Addressing first-and second-order barriers to change: Strategies for technology integration. *Educational Technology Research and Development*, v. 47, n. 4, p. 47-61, Dec. 1999. Available at: https://www.researchgate.net/publication/225685117_Addressing_first-_and_secondorder_barriers_to_change_Strategies_for_technology_integrationEducational_Technology_Research_and_Development_474_47-61. Accessed on: 15 Feb. 2024.

18. Fullan, Michael. *The new meaning of educational change*. 4th ed. New York: Teachers College Press, 2007. Available at: https://www.daneshnamehicsa.ir/userfiles/files/1/6-%20The%20New%20Meaning%20of%20Educational%20Change,%20Fourth%20Edition.pdf. Accessed on: 15 Feb. 2024.

19. Greengard, Samuel. *The Internet of Things*. MIT Press, 2015. Available at: https://direct.mit.edu/books/book/4051/The-Internet-of-Things. Accessed on: 15 Feb. 2024.

20. Hall, Gene E.; Hord, Shirley M. *Implementing change: Patterns, principles, and potholes*. University of Nevada: Pearson, 2015. Available at: https://daneshnamehicsa.ir/userfiles/file/Manabeh/Implementing%20Change_%20Patterns,%20-%20Gene%20E.%20Hall%20(2).pdf. Accessed on: 15 Feb. 2024.

21. Hmelo-Silver, Cindy E. Problem-based learning: What and how do students learn? *Educational Psychology Review*, v. 16, n. 3, p. 235-266, Sep. 2004. Available at: https://docdrop.org/static/drop-pdf/Hmelo-Silver2004-ZZaX8.pdf. Accessed on: 15 Feb. 2024.

22. Honey, Margaret; Pearson, Greg; Schweingruber, Heidi. *STEM Integration in K-12 Education: Status, Prospects, and an Agenda for Research*. Washington, DC: National Academies Press, 2014. Available at: https://www.middleweb.com/wp-content/uploads/2015/01/STEM-Integration-in-K12-Education.pdf. Accessed on: 15 Feb. 2024.

23. Kafai, Yasmin B.; Burke, Quinn. *Connected Code: Why Children Need to Learn Programming*. MIT Press, 2014. Available at: https://direct.mit.edu/books/monograph/4015/Connected-CodeWhy-Children-Need-to-Learn. Accessed on: 15 Feb. 2024.

24. Koehler, Matthew J.; Mishra, Punya. What is Technological Pedagogical Content Knowledge? *Contemporary Issues in Technology and Teacher Education*, v. 9, n. 1, p. 60-70, 2009. Available at: https://citejournal.org/wp-content/uploads/2016/04/v9i1general1.pdf. Accessed on: 15 Feb. 2024.

25. Krajcik, Joseph; Marx, Ron; Blumenfeld, Phyllis; Soloway, Elliot; Fishman, Barry. Inquiry-based science supported by technology: achievement among urban middle school students, 2008. Available at: https://files.eric.ed.gov/fulltext/ED443676.pdf. Accessed on: 15 Feb. 2024.

26. Kushner, David. The making of Arduino. *IEEE Spectrum*, 26 Oct. 2011, p. 26-29. Available at: https://web.eecs.umich.edu/~prabal/teaching/resources/eecs582/kushner11arduino.pdf. Accessed on: 15 Feb. 2024.

27. Maloney, John; Resnick, Mitchel; Rusk, Natalie; Silverman, Brian; Eastmond, Evelyn. The Scratch programming language and environment. *ACM Transactions on Computing Education*, v. 10, n. 4, p. 1-15, Nov. 2010. Available at: https://dl.acm.org/doi/10.1145/1868358.1868363. Accessed on: 15 Feb. 2024.

28. Margolis, Jane; Fisher, Allan. *Unlocking the Clubhouse: Women in Computing*. Cambridge: MIT Press, 2002. Available at: https://we.riseup.net/assets/459427/Margolis+Jane+Fisher+Allan+Unlocking+the+Clubhouse+Women+in+Computing.pdf. Accessed on: 15 Feb. 2024.

29. Martinez, Sylvia Libow; Stager, Gary S. *Invent to Learn: Making, Tinkering, and Engineering in the Classroom*. Constructing Modern Knowledge Press, 2013.

30. Mataric, Maja J.; Koenig, Nathan; Feil-Siefer, David. Materials for enabling hands-on robotics and STEM education. AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education, Stanford, CA, Mar. 2007. Available at: https://www.researchgate.net/publication/221250968_Materials_for_Enabling_Hands-On_Robotics_and_STEM_Education. Accessed on: 15 Feb. 2024.

31. Mellis, David A.; Banzi, Massimo; Cuartielles, David; Igoe, Tom. Arduino: An open electronics prototyping platform. *CHI'12 Extended Abstracts on Human Factors in Computing Systems*, p. 1-11, 2012. Available at: https://pt.scribd.com/document/116025464/Submission-Mellis-0. Accessed on: 15 Feb. 2024.

32. OECD. Organization for Economic Co-operation and Development. *Students, Computers and Learning: Making the Connection*. PISA: OECD Publishing, 2015. Available at: https://www.oecd-ilibrary.org/docserver/9789264239555-en.pdf?expires=1717800937&id=id&accname=guest&checksum=5007A8A9AF4B6C3B184C2C115E20FBEC. Accessed on: 15 Feb. 2024.

33. Papert, Seymour. *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books, 1980. Available at: https://worrydream.com/refs/Papert_1980_-_Mindstorms,_1st_ed.pdf. Accessed on: 15 Feb. 2024.PRENSKY, Marc. Digital natives, digital immigrants. On the Horizon, v. 9, n. 5, p. 1-6, out. 2001. Disponível em https://www.marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part1.pdf. Acesso em 15 fev. 2024.

34. Resnick, Mitchel et al. Scratch: Programming for All. *Communications of the ACM*, v. 52, n. 11, p. 60-67, Nov. 2009. Available at: https://dl.acm.org/doi/pdf/10.1145/1592761.1592779. Accessed on: 15 Feb. 2024.

35. Selwyn, Neil. *Education and Technology: Key Issues and Debates*. London: Continuum, 2011.

36. Siegwart, Roland; Nourbakhsh, Illah Reza; Sacaramuzza, Davide. *Introduction to Autonomous Mobile Robots*. MIT Press, 2011.

37. Trilling, Bernie; Fadel, Charles. *21st Century Skills: Learning for Life in Our Times*. John Wiley & Sons, 2009. Available at: http://ardian.id/wp-content/uploads/2018/10/21st_Century_Skills_Learning_for_Life_in_Our_Times____2009-3.pdf. Accessed on: 15 Feb. 2024.

38. Warschauer, Mark. *Technology and Social Inclusion: Rethinking the Digital Divide*. Cambridge: MIT Press, 2004. Available at: https://www.researchgate.net/publication/329649885_Technology_and_Social_Inclusion_Rethinking_the_Digital_Divide. Accessed on: 15 Feb. 2024.

39. Wiggins, Grant. The Case for Authentic Assessment. *Practical Assessment, Research, and Evaluation*, v. 2, n. 1, Nov. 2019. Available at: https://scholarworks.umass.edu/cgi/viewcontent.cgi?article=1024&context=pare. Accessed on: 15 Feb. 2024.

40. Wing, Jeannette M. Computational Thinking. *Communications of the ACM*, v. 49, n. 3, p. 33-35, Mar. 2006. Available at: https://dl.acm.org/doi/pdf/10.1145/1118178.1118215. Accessed on: 15 Feb. 2024.