Design of a Hardware Prototype with Block Programming to Develop Computational Thinking in Recently Admitted Engineering Students

Ronald Paucar-Curasma[1], Abraham Gamarra-Moreno[2], David Rondon[3], Roberto Unsihuay Tovar[4], Freddy Toribio Huayta-Meza[5]

[1] *Universidad Nacional Autónoma de Tayacaja Daniel Hernández Morillo, Jr. Bolognesi N° 416, Pampas, 09000, Perú*
[2] *Universidad Nacional del Centro del Perú, Av. Mariscal Castilla 3909, Huancayo, 12001, Perú*
[3] *Universidad Continental, Calle Alfonso Ugarte 607, Arequipa, 04001, Perú*
[4] *Universidad Nacional Mayor de San Marcos, Av. Venezuela S/N, Lima, 15001, Perú*
[5] *Universidad Nacional de Huancavelica, Paturpampa S/N, Huancavelica, 09001, Perú*

Summary
In the context of university education, students should start their engineering careers with computational thinking and problem-solving skills. This article proposes the design of a hardware prototype with block programming to develop computational thinking skills in recently admitted engineering students. The prototype consists of two components: hardware and software. The hardware component consists of 6 electronic sensors and 5 output devices. The software component is based on a block programming environment; this programming environment is essential for students starting the engineering career; even more so for students who come from rural areas, with limitations in the use of ICT.

Keywords
Computer optimization, university, recent admission, electronic sensors, block programming.

1. Introduction

In Latin America, countries are heterogeneous and each country within it presents differences between rural and urban areas; there are students of high and low socioeconomic levels, there are different cultural levels, and even more there are pronounced differences between rural and urban schools with respect to educational quality [1], [2]. In the Huancavelica region, the majority of students who enter university come from rural educational institutions, with low skills in mathematics, problem solving and teamwork [3], [4].

In recent years, there has been research on the effectiveness of interventions based on educational electronic boards in beginning students of STEM (Science, Technology, Engineering and Mathematics). The use of microcontrollers, programmable toys, educational robots and Arduino board in combination with block programming has been the focus of many studies in STEM teaching and learning. Teaching/learning methods based on these electronic devices have been shown to improve students' academic skills [5]–[7]. With the thinking of experts in problem solving, more than a decade of speech to capture your cognitive essence has resulted in a broad set of skills; teaching beginning students continues to be done through the use of computers, educational robots, educational kids and programming concepts, strengthening students' computational thinking and problem-solving skills [8].

Combining effective pedagogies with educational technologies to introduce computational thinking to beginning students is likely to enhance and inspire creativity and problem-solving, rather than

CITIE 2022: Congreso Internacional de Tecnología e Innovación Educativa, November 08-10, 2022, Arequipa, Peru
EMAIL: rpaucarc@unat.edu.pe (R. Paucar-Curasma); aegamarra@uncp.edu.pe (A. Gamarra-Moreno)); drondon@continental.edu.pe (K. Gamarra-Moreno); runsihuayt@unmsm.edu.pe (R. Unsihuay); freddy.huayta@unh.edu.pe (F. T. Huayta-Meza)
0000-0002-9489-576X (R. Paucar-Curasma); 0000-0002-3953-8014 (A. Gamarra-Moreno); 0000-0003-3506-5309 (D. Rondon); 0000-0002-4086-3527 (R.. Unsihuay); 0000-0001-9606-6343 (F. T. Huayta-Meza)

frustration and discouragement in students [9]. Age-appropriate curriculum and technologies are of key importance; for example, well-designed coding tasks integrated into collaborative, problem-based, and/or thematic curriculum designs can support a wide range of students' thinking abilities [10] [35]. Common computational thinking skills are: abstraction, pattern recognition, decomposition, and algorithmic design [11]–[13]; thus, as concepts, practices and computational perspectives [14] [36] and take greater importance to apply them from the first years of study of the university.

This article designs a hardware prototype with block programming to develop computational thinking skills in beginning engineering students or new entrants to college.

2. Computational thinking and educational technology with block programming

It is essential to define computational thinking in higher education; thus, also the varieties of technologies with block programming that are used to teach programming and develop computational thinking in beginning students of higher education.

2.1. Computational thinking

Computational thinking is one of the most important technological novelties of scientific thought in recent years; computational thinking is considered so that the disciplines of computation are no longer limited only to the use of computer tools, but allow students to become Researchers and innovators in knowledge discovery; computational thinking, also offers limitless possibilities and ways to solve the problem [24].

Several authors express the importance and benefits of computational thinking in higher education. With regard to computational thinking skills, abstraction and algorithmic thinking stand out in strengthening competencies in reading comprehension and in solving complex problems following algorithmic methods [15]–[19]. In a study, it is pointed out that programming is used to perform mathematical tasks, and can improve the student's understanding in abstract topics and with some complexity in its solution; this implies the use of the computer as a tool to solve problems of this discipline [20]. The use of tools based on block programming and interaction with hardware, generates interest and motivation in students; as well as teamwork and problem solving through abstraction, decomposition and algorithmic thinking skills [3]. Also, It directly relates the development of common skills through creative programming and innovation [21], [22]. Finally, critical thinking is added as a form of reasoning and exchange of ideas prior to solving problems through computational thinking skills [23].

2.2. Educational technologies with block programming

Block-based programming languages have been a popular, low-cost method of teaching programming and computational thinking to students and educators with computer science limitations or beginning students. The use of block-based programming languages has been shown to be successful in developing skills in different areas such as computer science and computational thinking [22] [37]. For example, block-based programs such as Scratch or mBlock [25] allow beginning students to create stories of their city in a given historical period, and in such activities students use skills and knowledge of mathematics, technology, communication and social sciences [16], [26]. The most common programming tools used in beginning students are: Scratch, mBlock and App Inventor; the visual assessment tool is also based on blocks, such as Dr. Scratch [26] [38].

Robot programming has shown benefits in the educational context and also shows value as an aid to building computational thinking skills as well, such as teamwork, problem-solving and creativity skills [27]–[30]. However, in several contexts an important limitation imposed by the educative robots is the cost involved in the acquisition and implementation of these technologies; in addition, a robot can be a complex environment for elementary algorithms, or to solve real problems of society.

The creation of custom hardware prototypes using processor boards (Arduino board and esp8266) helps in the acquisition of abstraction, problem solving and algorithmic thinking skills, and have been proved to be low-cost alternatives and are so illustrative as commercial solutions (educational robotics or other educational technologies)[27].

3. Hardware prototype design with block-based programming

Various electronic devices (sensors and output devices) have been considered to design a hardware prototype with block-based programming. Through the hardware prototype, the beginning student will develop the 3 dimensions of computational thinking (concepts, practices and computational perspectives).

3.1. Hardware prototype with block programming

The hardware prototype design proposal with block programming consists of two main components (hardware and software), as shown in Figure 1 and Table 1. The hardware component is composed of a processor (Arduino board and esp8266), sensors, actuators and a module with connectivity to the IoT (Internet of Things). Sensors and actuators are related to features of the real-world environment around us. The electronic sensors are: ambient temperature and humidity sensor (DHT11), water temperature sensor (DS18B20), crop soil moisture sensor, ambient light sensor (LDR), nearby obstacle detection sensor (HC-SR04-PIR-HC-SR501). The output devices are: display of states and values of environmental parameters (screen OLED-7 segments-LEDs), control of servo motors (SG90), emission of sound for the generation of alerts (buzzer) and sending of data using IoT technology. The software component is made up of a development environment based on the mBlock software [31].

With the development of the hardware prototype with block programming, students recently admitted at university will develop activities or projects in the classroom, following the steps of problem solving: understanding the problem, preparing the plan, executing the plan and reviewing the solution [32] [39]. These activities or projects will be related to the real problems of the students' context; during the execution of the project, the students will develop programs, interact with the sensors and show the results in the output devices; these acquired skills will have an impact on their university training by providing skills in critical thinking and teamwork.

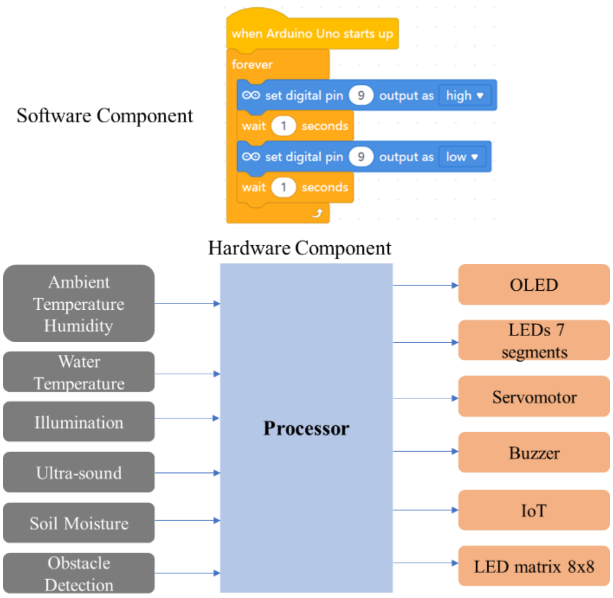
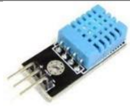
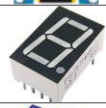


Figure1: Hardware prototype with block programming

Table 1

Electronic sensors and output devices

	Devices	Physical appearance
	Sensors	
1	Ultrasonic Sensor HC-SR04	
2	DHT11 Temperature/Humidity Sensor	
3	HC-SR501 PIR Infrared sensor	
4	LDR Sensor (Light Dependent Resistor)	
5	FC-28 Soil moisture	
6	DS18B20 Water temperature	
	Output devices	
1	7Pin 0.96 Inch OLED	
2	LEDs 7 segments	
3	Servomotor	
4	Buzzer	
5	MAX7219 8X8 LED matrix	

3.2. Developing computational thinking skills with the hardware prototype

The hardware prototype with block programming in recent students will allow the development of computational thinking skills based on the 3 computational dimensions (Table 2): concepts, practices and computational perspectives [14] [34]. Students will develop technological projects using the prototype; these projects will be related to solving problems in their geographical context; since the prototype will allow to implement different solutions based on electronic sensors; for example, ambient temperature sensor, distance sensor, water temperature sensor, obstacle detection sensor, etc., these solutions will have a graphical interface that will be developed through programming in blocks, such as the mBlock, which will allow interacting with the sensors and verifying the results in the output devices (LEDs, servomotor, buzzer, LED matrix, etc.). The use of a permission-based programming interface will allow students to focus on computational concepts rather than the syntax of programming languages; while the presence of electronic sensors and output devices will allow students to enthusiastically view the actual movement/consequence of the program occurring in the physical world, generating immediate visual feedback from programming, motivating beginning students to more easily test their hypotheses and refine their ideas [9], [33].

Table 2

Dimensions of computational thinking [14]

Computational dimensions	Description	Indicators
Computational concepts	Concepts used in the development of the activity	Sequences
		Cycles
		Events
		Parallelism
		Conditional
		Operators
		Data
Computational practices	Problem-solving practices that occur in the process of developing the activity	Experimentation and Iteration
		Testing and Debugging
		Reuse of previous projects
		Abstraction and modularization
Computational perspectives	Students' understanding of themselves, their relationships with others and the technological world around them.	Ways of expressing oneself
		Connection with others
		Questions

4. Conclusions

This article proposes the design of a hardware prototype with block-based programming, as a technological alternative that will allow the development of computational thinking skills in students recently admitted at university; this prototype will help in the teaching/learning process about computational thinking through the creation of real scenarios (i.e. a hardware object that can be touched and controlled) as a final product that interacts with a graphical interface through block programming. By using electronic components or sensors as a low-cost strategy, students will be able to implement successful solutions to solve real problems in their context. Finally, we also believe that the proposed hardware prototype is a cost-effective solution; our approach can be valuable as an educational resource in economically challenged or poor regions, such as Peru's Huancavelica region; in addition, they can be integrated into course curricula related to programming, experimental methods, information management, and other similar disciplines.

5. References

[1]G. Sunkel and D. Trucco, «ICT for education in Latin America. Risks and opportunities", Santiago, Chile, 2010.

[2]D. Trucco, «Education and inequality in Latin America», Santiago de Chile, 2016. [Online]. Available in: https://repositorio.minedu.gob.pe/bitstream/handle/20.500.12799/3053/EducacionydesigualdadenAL.pdf.

[3]C. K. Shyamala, C. S. Velayutham, y L. Parameswaran, «Teaching computational thinking to entry-level undergraduate engineering students at Amrita University», 2017, pp. 1731-1734, doi: 10.1109/EDUCON.2017.7943084.

[4]P. D. L. Sobreira et al., «Usability evaluation of block programming tools in IoT contexts for initial engineering courses», 2020, doi: 10.1109/EDUNINE48860.2020.9149481.

[5]A. Fidai, M. M. Capraro, y R. M. Capraro, «"Scratch"-ing computational thinking with Arduino: A meta-analysis», Think. Ski. Creat., vol. 38, n.o July, p. 100726, 2020, doi: 10.1016/j.tsc.2020.100726.

[6]R. Paucar, N. Jara, H. Paucar, y V. Cruz, «Assessment of Computational Thinking in regular basic education: Case I.E.T.P. "josé Obrero"», Proc. 2019 IEEE 26th Int. Conf. Electron. Electr. Eng. Comput. INTERCON 2019, 2019, doi: 10.1109/INTERCON.2019.8853613.

[7]X. Zhou y L. Chen, «Computational thinking of the university computer course teaching system», J. Adv. Oxid. Technol., vol. 21, n.o 2, 2018, doi: 10.26802/jaots.2018.09320.

[8]O. Yasar, «Computational Thinking , Redefined», en Proceedings of Society for Information Technology & Teacher Education International Conference, 2018, pp. 72-88, [En línea]. Disponible en: https://www.learntechlib.org/primary/p/182505/.

[9]Y. H. Ching, Y. C. Hsu, y S. Baldwin, «Developing Computational Thinking with Educational Technologies for Young Learners», TechTrends, vol. 62, n.o 6, pp. 563-573, 2018, doi: 10.1007/s11528-018-0292-7.

[10]G. Falloon, «What's the difference? Learning collaboratively using iPads in conventional classrooms», Comput. Educ., vol. 84, pp. 62-77, 2015, doi: 10.1016/j.compedu.2015.01.010.

[11] Á. Molina, N. Adamuz, and R. Bracho, «Problem solving based on the Polya method using computational thinking and Scratch with secondary school students», Handb. Educ. Psychol., pp. 287–303, 2020.

[12]B. Ortega and M. Asensio, «Evaluating Computational Thinking through Problem Solving: Validation of an Evaluation Instrument», Rev. Iberoam. Educ.-Evaluation, vol. 14, no. 1, pp. 153-171, 2021, doi: 10.15366/riee2021.14.1.009.

[13]R. Paucar-Curasma, K. O. Villalba-condori, N. Jara, R. Quispe, J. Cabrera, y M. Ponce-Aguilar, «Computational Thinking and Block-Based Programming for Beginning Engineering Students : Systematic Review of the Literature», 2021 XVI Lat. Am. Conf. Learn. Technol., pp. 530-533, 2021.

[14]K. Brennan y M. Resnick, «New frameworks for studying and assessing the development of computational thinking», Proc. 2012 Annu. Meet. Am. Educ. Res. Assoc. Meet. Vancouver, BC, Canada, pp. 1-25, 2012, doi: 10.1.1.296.6602.

[15]F. Alegre, J. Moreno, T. Dawson, E. E. Tanjong, y D. H. Kirshner, «Computational Thinking for STEM Teacher Leadership Training at Louisiana State University», 2020, doi: 10.1109/RESPECT49803.2020.9272455.

[16]K. Harangus y Z. Kátai, «Computational thinking in secondary and higher education», 2020, vol. 46, pp. 615-622, doi: 10.1016/j.promfg.2020.03.088.

[17]T. S. C. da Silva, J. C. B. de Melo, y P. Tedesco, «The creative process in the development of computational thinking in higher education», 2020, vol. 1, pp. 215-226, [En línea]. Disponible en: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85091441415&partnerID=40&md5=ed29efdb47cec9249a9c71fcc5e58648.

[18]J. Wu, Y. Wang, H. Kong, y L. Zhu, «How to cultivate computational thinking-enabled engineers: A case study on the robotics class of zhejiang university», 2019.

[19]R. Delgado, S. Yacchirema, F. Coral, y J. Celi, «Platform for university teaching applying Computational Thinking at the Armed Forces University ESPE », 2019, vol. 2019-June, doi: 10.23919/CISTI.2019.8760819.

[20]J. W. Wilson, M. L. Fernandez, y N. Hadaway, «Mathematical Problem Solving», The University of Georgia, 1993. http://jwilson.coe.uga.edu/emt725/PSsyn/Pssyn.html (accedido mar. 26, 2022).

[21]A.-D. Fang, G.-L. Chen, Z.-R. Cai, L. Cui, y L. Harn, «Research on blending learning flipped class model in colleges and universities based on computational thinking - "Database principles" for example», Eurasia J. Math. Sci. Technol. Educ., vol. 13, n.o 8, pp. 5747-5755, 2017, doi: 10.12973/eurasia.2017.01024a.

[22]J. L. Weese, «Mixed methods for the assessment and incorporation of computational thinking in K-12 and higher education», 2016, pp. 279-280, doi: 10.1145/2960310.2960347.

[23]B. Kules, «Computational thinking is critical thinking: Connecting to university discourse, goals, and learning outcomes», Proc. Assoc. Inf. Sci. Technol., vol. 53, n.o 1, pp. 1-6, 2016, doi: 10.1002/pra2.2016.14505301092.

[24]B. Liu y J. He, «Teaching mode reform and exploration on the University Computer Basic based on Computational thinking training in Network Environment», 2014, pp. 59-62, doi: 10.1109/ICCSE.2014.6926430.

[25]Makeblock, «My Blocks Category», 2022. https://www.mblock.cc/doc/en/block-reference/DIY.html (accedido abr. 03, 2022).

[26]M. Romero, A. Lepage, y B. Lille, «Computational thinking development through creative programming in higher education», Int. J. Educ. Technol. High. Educ., vol. 14, n.o 1, 2017, doi: 10.1186/s41239-017-0080-z.
[27]I. Fronza, L. Corral, y C. Pahl, «Combining block-based programming and hardware prototyping to foster computational thinking», SIGITE 2019 - Proc. 20th Annu. Conf. Inf. Technol. Educ., pp. 55-60, 2019, doi: 10.1145/3349266.3351410.
[28]A. Suárez, D. García, P. A. Martínez, and J. Martos, «Contribution of educational robotics in the acquisition of mathematics knowledge in Primary Education», vol. 30, pp. 9-20, 2018.
[29]L. Daniela y M. D. Lytras, «Educational Robotics for Inclusive Education», Technol. Knowl. Learn., n.o 0123456789, 2018, doi: 10.1007/s10758-018-9397-5.
[30]M. Berland y U. Wilensky, «Comparing Virtual and Physical Robotics Environments for Supporting Complex Systems and Computational Thinking», J. Sci. Educ. Technol., vol. 24, n.o 5, pp. 628-647, 2015, doi: 10.1007/s10956-015-9552-x.
[31]Makeblock, «mBlock - One-Stop Coding Platform for Teaching and Learning», mBlock, 2021. https://mblock.makeblock.com/en-us/ (accedido dic. 05, 2021).
[32]G. Pólya, How to Solve, 2nd ed. 1957.
[33]Y. B. Kafai y Q. Burke, Connected code: why children need to learn programming. MIT Press, Cambridge, 2014.
[34]K. O. Villalba-Condori, S. E. C. Cuba-Sayco, E. P. G. Chávez, C. Deco, C. Bender, Approaches of learning and computational thinking in students that get into the computer sciences career, in: Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality, TEEM'18, Association for Computing Machinery, New York, NY, USA, 2018, p. 36–40. URL: https://doi.org/10.1145/3284179.3284185. doi:10.1145/3284179.3284185.
[35]T. J. Cateriano-Chavez, M. L. Rodríguez-Rios, E. L. Patiño-Abrego, R. L. Araujo-Castillo, K. Villalba-Condori, Competencias digitales, metodología y evaluación en formadores de docentes, Campus virtuales 10 (2021) 153–162.
[36]L. E. H. Benites, K. O. Villalba-Condori, D. Arias-Chávez, M. Berrios-Espezua, S. Cano, Aula invertida en una plataforma virtual para el desarrollo de competencias. caso de estudio: curso de investigación aplicada, Campus Virtuales 10 (2021) 185–193.
[37]L. E. Quispe-Bendezú, R. L. Araujo-Castillo, J. E. García-Tejada, Y. García-Tejada, A. S. Sprock, K. O. Villalba-Condori, Relationship between academic procrastination and attributions of achievement motivation, International Journal of Learning, Teaching and Educational Research 19 (2020) 188–205.
[38]R. Paucar-Curasma, K. Villalba-Condori, D. Arias-Chavez, N.-T. Le, G. Garcia-Tejada, I. Frango-Silveira, Evaluación del pensamiento computacional utilizando cuatro robots educativos con estudiantes de primaria en perú, Education in the Knowledge Society (EKS) 23 (2022). URL: https://revistas.usal.es/tres/index.php/eks/article/view/26161. doi:10.14201/eks.26161.
[39]A. Garcia-Holgado, C. Deco, N. Bredegal-Alpaca, C. Bender, K. O. Villalba-Condori, Perception of the gender gap in computer engineering studies: a comparative study in peru and argentina, in: 2020 IEEE Global Engineering Education Conference (EDUCON), 2020, pp. 1252–1258. doi:10.1109/EDUCON45650.2020.9125224.