electronics

MDPI

Article

Design, Implementation and Functional Verification of an Educational Prototype with a Visual Programming Interface for Contextualized STEM Activities

Ronald Paucar Curasma [1,*], Roberto Florentino Unsihuay Tovar [2], Claudia Acra-Despradel [3] and Augusto Caceres Nuñez [4]

[1] Grupo de Investigación TIC Aplicadas a la Sociedad, Universidad Nacional Autónoma de Tayacaja Daniel Hernández Morillo, Pampas 09156, Peru
[2] Facultad de Ingeniería Electrónica y Eléctrica, Universidad Nacional Mayor de San Marcos, Lima 15001, Peru
[3] Vicerrectorado de Investigación, Universidad Nacional Pedro Henríquez Ureña, Santo Domingo 10100, Dominican Republic
[4] Facultad de Ciencias e Ingenierías, Físicas y Formales, Universidad Católica de Santa María, Arequipa 04001, Peru
* Correspondence: rpaucarc@unat.edu.pe

Abstract

The integration of open-source hardware and visual programming technologies provides an effective approach for developing contextualized STEM activities in higher education. This study presents the design, implementation, and functional verification of an educational prototype that combines customized electronic boards, sensors, and a visual programming interface developed in mBlock. The proposed platform integrates three educational boards—the Agriculture Board, Environment Board, and Aquaculture Board—for monitoring soil moisture, air-quality conditions, and water turbidity, respectively. The hardware implementation included the design and fabrication of customized printed circuit boards, followed by electrical continuity testing, power supply verification, and firmware execution. Customized programming blocks and communication extensions were then developed to enable interaction between the educational boards and the mBlock environment. Finally, the complete prototype was functionally verified during classroom implementation through contextualized STEM activities based on problem understanding, activity development, activity execution, and solution review. The results confirmed the correct operation of the hardware platform, successful communication between the educational boards and the visual programming environment, and real-time acquisition and visualization of sensor measurements. The integrated functional verification demonstrated correct interaction among the hardware platform, software extensions, connected sensors, and customized programming blocks under authentic classroom conditions. The proposed prototype provides a functional technological platform for implementing contextualized sensor-based STEM activities and demonstrates the feasibility of integrating open-source hardware and visual programming into higher education.

Keywords: STEM education; open-source hardware; visual programming; mBlock; educational prototype; functional verification; contextualized STEM activities

Check for updates

Academic Editor: Stefano Scanzio

Received: 19 June 2026
Revised: 4 September 2026
Accepted: 16 September 2026
Published: 21 September 2026

Copyright: © 2026 by the authors. 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.

1. Introduction

Higher education faces the challenge of preparing students who are capable of applying scientific and technological knowledge to solve real-world problems within their

local contexts [1,2]. Although undergraduate programs provide theoretical foundations in science and engineering, first-year university students frequently experience difficulties in connecting academic concepts with practical applications, particularly when they are required to integrate knowledge from science, technology, engineering, and mathematics (STEM) to solve interdisciplinary problems [3]. This situation highlights the need for educational approaches that promote active learning, experimentation, and the application of technological resources in authentic learning environments [4].

In recent years, STEM education has become an important educational approach for promoting critical thinking, creativity, collaborative learning, and technological problem-solving skills [2]. Rather than teaching science, technology, engineering, and mathematics as isolated disciplines, STEM education encourages their integration through the analysis and solution of contextualized problems [5]. Several studies have shown that meaningful STEM activities allow students to investigate real situations, design technological solutions, evaluate alternatives, and apply scientific reasoning to address challenges encountered in their surrounding environment [6,7]. Recent advances in open-source hardware, embedded systems, and sensors have created new opportunities for the development of educational technologies that support contextualized STEM learning. Platforms such as Arduino and ESP8266, together with environmental and agricultural sensors, provide flexible and affordable solutions for designing electronic systems capable of monitoring physical and environmental variables [4].

In parallel, block-based visual programming environments have significantly reduced the complexity associated with introductory programming. Platforms such as mBlock allow novice users to develop applications through graphical programming blocks instead of text-based programming languages [8]. This approach facilitates the configuration of sensors, acquisition of environmental data, control of electronic devices, and visualization of experimental results while lowering the entry barrier for students with limited programming experience. Consequently, visual programming has become an effective interface for integrating hardware platforms into STEM education.

Most existing studies focus either on isolated hardware developments or on educational experiences, whereas limited attention has been given to integrated educational prototypes that combine customized electronic boards, sensor technologies, and visual programming within a unified platform for contextualized STEM activities [9]. This gap limits the availability of validated technological solutions that can be readily adopted by educational institutions.

Unlike general-purpose educational electronics platforms that typically require the assembly and configuration of separate microcontroller boards, sensor modules, communication components, and programming libraries, the proposed prototype was conceived as an integrated hardware–software platform for contextualized STEM activities. Its technical contribution lies in the coordinated development of three dedicated electronic boards—the Agriculture Board, Environment Board, and Aquaculture Board—together with sensor-specific interfaces and customized mBlock extensions and programming blocks. This architecture provides a unified interface between physical sensing, data acquisition, communication, and block-based visual programming. The contribution of this study is therefore focused on the design, implementation, integration, and functional verification of the complete prototype rather than on demonstrating educational effectiveness or metrological performance.

Therefore, the objective of this study was to design, implement, and functionally verify an educational prototype with a visual programming interface for contextualized STEM activities. The proposed prototype integrates three customized educational electronic boards for agriculture, environment, and aquaculture, together with sensors and customized pro-

gramming blocks developed for the mBlock environment. The functional verification was conducted in a real classroom environment through STEM activities involving soil moisture monitoring, air quality assessment, and water turbidity measurement.

The main contribution of this work is the development of an integrated educational technology platform that combines open-source hardware, customized electronic boards, sensors, and a visual programming interface within a single educational ecosystem. Unlike previous studies that primarily focus on either hardware implementation or classroom applications, this work presents the complete engineering process, from prototype design and implementation to functional verification in an authentic educational setting. The results demonstrate the technical feasibility of integrating hardware and software components to support contextualized STEM activities and provide a reusable technological platform for future educational and research applications.

The educational electronic boards used in the present study are the same hardware boards previously reported by the authors. However, the present study addresses a different and more specifically delimited contribution, focusing on their integrated hardware–software architecture, implementation with customized mBlock extensions, and functional verification at the hardware, software, and integrated-system levels. Accordingly, the contribution reported here does not arise from the development of new board designs, but from the technical integration and functional-verification framework examined in the present study.

The remainder of this paper is organized as follows. Section 2 reviews the related literature on STEM education, block-based programming, open-source hardware, and educational prototype verification. Section 3 describes the design and implementation of the proposed hardware and software components. Section 4 presents the functional verification of the prototype and its classroom demonstration through contextualized STEM activities. Finally, Section 5 summarizes the main conclusions and outlines future research directions.

2. Literature Review

2.1. STEM Education and the Integration of STEM Disciplines

The primary purpose of education is to enable individuals to apply knowledge to improve their quality of life and address problems within their communities. However, many students completing higher education encounter difficulties in transferring acquired knowledge to real-life situations or require extended periods to develop a scientific and technological perspective aimed at solving societal challenges [10]. In response to this issue, Science, Technology, Engineering, and Mathematics (STEM) education has gained increasing attention across educational disciplines and educational levels, particularly when these areas are integrated into the teaching and learning process [11].

STEM education focuses on the integration of four fundamental disciplines with the goal of enabling learners to solve real-world problems through interdisciplinary knowledge and practical application [12]. Nevertheless, integrating these disciplines can be challenging for many students because it requires critical thinking, mathematical reasoning, technological literacy, and the ability to design effective solutions [13]. Consequently, introducing STEM experiences during the early stages of higher education is considered beneficial, as it promotes analytical thinking, practical knowledge application, and active participation in addressing challenges within local and global contexts [14,15].

The STEM acronym encompasses four complementary domains. Science (S) refers to the systematic study of the natural world through observation, experimentation, and analysis [16]. Technology (T) involves the application of tools, software, technological processes, and electronic devices to solve problems and improve everyday life [17,18]. Engineering

(E) is concerned with the design, construction, and optimization of systems, structures, and technological products through systematic problem-solving approaches [19]. Finally, Mathematics (M) provides the analytical language and quantitative methods required for modeling, data analysis, pattern recognition, and problem-solving across the other STEM disciplines.

From an educational perspective, STEM is understood as an interdisciplinary approach that integrates engineering and technology to enhance the learning of science and mathematics while promoting active student participation [7]. Contemporary STEM education emphasizes student-centered learning, collaborative work, interdisciplinary integration, and the application of knowledge to authentic situations across all educational levels [20,21]. However, several authors argue that STEM integration should not be regarded as an end in itself. Rather, meaningful integration should be guided by authentic problems and explicit learning objectives, since increasing the level of disciplinary integration alone does not necessarily produce better educational outcomes [6].

Researchers have identified three essential characteristics of effective STEM education: the reduction in traditional disciplinary boundaries, the meaningful integration of STEM disciplines, and the development of learning activities situated in real-world contexts; consequently, STEM education is closely associated with educational methodologies such as problem-solving, problem-based learning, and project-based learning, where students investigate authentic challenges using technological resources, including both software and hardware tools [22]. Through these experiences, learners assume active roles as innovators, designers, logical thinkers, and technologically literate individuals capable of proposing sustainable solutions [23].

Globally, STEM education is viewed from complementary perspectives. On one hand, it is considered an economic imperative because modern societies require professionals capable of working in technology-driven and highly competitive environments. On the other hand, it is regarded as a social imperative that prepares responsible citizens to address environmental, technological, and societal challenges through informed decision-making and innovation [24].

In the present study, STEM education is understood as an interdisciplinary approach in which science, technology, engineering, and mathematics are integrated through contextualized problem-solving activities supported by open-source hardware, sensors, and a visual programming interface. Under this perspective, students interact with real-world data acquired from electronic sensors, develop technological solutions through block-based programming, and analyze experimental results within authentic learning environments.

Overall, the reviewed literature indicates that successful STEM implementation depends not only on integrating multiple disciplines but also on providing authentic contexts, appropriate technological resources, and opportunities for active experimentation. These principles constitute the conceptual foundation of the proposed educational prototype, which integrates customized electronic boards, environmental sensors, and a visual programming interface into a single platform for developing contextualized STEM activities.

2.2. Educational Technologies with Block-Based Programming Interfaces

Block-based programming environments have become one of the most widely adopted educational technologies for introducing programming, computational thinking, and STEM concepts to novice learners. By replacing text-based syntax with graphical programming blocks, these environments reduce programming complexity and enable students to focus on algorithmic reasoning, logical thinking, and problem solving [25]. Consequently, block-based programming has become an effective approach for supporting introductory programming courses and technology-enhanced STEM education.

Among the most widely adopted educational platforms is MIT App Inventor, originally developed by Google and currently maintained by the Massachusetts Institute of Technology (MIT). App Inventor enables users to develop Android applications through a graphical drag-and-drop interface while incorporating fundamental programming constructs such as variables, loops, and conditional statements. Its cloud-based environment and integrated emulator facilitate application development and testing without requiring extensive programming experience, making it an effective tool for introducing computational thinking and mobile application development [26].

Another widely used educational platform is mBlock, which was specifically developed to support STEM education through the integration of block-based and text-based programming [22]. Unlike traditional visual programming environments, mBlock supports the programming of Arduino boards, educational robots, Internet of Things (IoT) devices, and artificial intelligence (AI) applications within a unified development environment. Furthermore, recent versions allow the creation of customized extensions and Python 3.12 based functionalities, enabling developers to incorporate new hardware devices and educational applications into the platform [27]. These capabilities make mBlock particularly suitable for educational environments that require direct interaction between programming interfaces and electronic systems.

Open-source hardware platforms, particularly Arduino, are commonly integrated with block-based programming environments because of their flexibility, accessibility, and ease of use [28]. Arduino enables students to design electronic circuits, interface sensors and actuators, and implement embedded applications while gaining practical experience in electronics, automation, data acquisition, and control systems. Such experiences also introduce students to emerging technological concepts associated with Industry 4.0, including the Internet of Things (IoT), cloud computing, machine learning, big data, and cybersecurity.

Sensors represent another essential component of educational electronic platforms. These devices convert physical or chemical phenomena into digital signals that can be processed by embedded systems, enabling the monitoring of variables such as temperature, humidity, pressure, light intensity, air quality, distance, noise, and motion. Their integration with open-source hardware platforms allows students to acquire and analyze real-world data while developing practical engineering solutions for applications in agriculture, environment, industry, and other technological domains.

Several studies have demonstrated that block-based programming environments provide an effective and cost-efficient approach for teaching programming and computational thinking to students with limited computing backgrounds [25]. Platforms such as Scratch, App Inventor, and mBlock have been successfully applied to develop interactive projects integrating mathematics, science, engineering, and technology while promoting creativity, collaboration, and algorithmic reasoning [29,30]. In addition, educational robotics has shown positive effects on computational thinking and teamwork; however, the relatively high cost of commercial robotic platforms often limits their adoption, particularly in resource-constrained educational institutions [31–34].

An alternative approach consists of developing customized educational hardware based on open-source microcontrollers such as Arduino and ESP8266. Previous studies indicate that personalized hardware platforms can provide learning experiences comparable to commercial educational robotics while significantly reducing implementation costs [31]. Moreover, customized platforms offer greater flexibility for integrating specific sensors, communication modules, and educational applications tailored to local learning contexts.

Overall, the reviewed literature demonstrates that block-based programming environments, open-source hardware, and sensors constitute complementary technologies for supporting contextualized STEM education. However, most previous studies focus on

individual software platforms, commercial robotics, or isolated hardware implementations. Comparatively fewer studies describe integrated educational platforms that combine customized electronic boards, sensor technologies, and visual programming extensions within a unified ecosystem. This gap motivates the present study, which proposes the design, implementation, and functional verification of an educational prototype integrating customized electronic boards and mBlock extensions to support contextualized STEM activities in authentic classroom environments.

2.3. Open-Source Hardware for Contextualized STEM Educational Applications

Open-source hardware platforms based on low-cost microcontrollers, such as Arduino, have significantly expanded opportunities for developing educational technologies because of their affordability, flexibility, and ease of integration with sensors, actuators, and communication modules. These characteristics have made open-source hardware an attractive alternative for designing educational prototypes that support hands-on STEM learning. Barbosa [35] reported that the functionality of Arduino-based platforms can be extended through modular shields, allowing additional hardware capabilities to be incorporated into a single electronic platform. Similarly, Rahman [36] developed an Arduino-based shield for acquiring bioelectrical signals, including ECG, EMG, EEG, and EOG, demonstrating the adaptability of open-source hardware for specialized sensing applications. Plaza [37] further emphasized the educational value of modular electronic platforms, highlighting their ability to facilitate practical STEM activities through the integration of science, technology, engineering, and mathematics.

In the field of agriculture, open-source hardware has been extensively employed for monitoring environmental variables and implementing precision agriculture solutions. Bhadani and Vashisht [38] developed an Arduino-based monitoring system capable of measuring air temperature, relative humidity, and soil moisture using DHT11 and FC28 sensors. Likewise, Prasojo [39] proposed an automated irrigation system integrating soil moisture sensors, relay modules, and solenoid valves to optimize water consumption during dry seasons. Beyaz and Gül [40] evaluated low-cost light sensors, including the TSL2561, BH1750, and TEMT6000, for monitoring artificial illumination in controlled agricultural environments. Tuysuzoglu [41] combined Arduino Uno with multiple environmental sensors and machine-learning techniques to estimate soil temperature profiles, while Poojitha et al. developed an IoT-based smart agriculture platform incorporating Arduino Uno, environmental sensors, wireless communication, and cloud-based monitoring. Similarly, Botero-Valencia [42] designed a low-cost meteorological station integrating renewable energy, wireless communication, and environmental sensing for precision agriculture.

Open-source hardware has also demonstrated considerable potential in aquaculture applications. Saha [43] proposed an IoT-based monitoring system integrating Arduino Uno, water-quality sensors, Raspberry Pi, and mobile technologies for monitoring water temperature, pH, electrical conductivity, and watercolor. Likewise, Riansyah [44] developed an aquaponics monitoring system incorporating pH and total dissolved solids sensors together with automated fish feeding based on Arduino and ESP8266 technologies. Sarwar [45] presented a real-time monitoring platform for aquaculture management capable of measuring water level, temperature, pH, and dissolved oxygen through Arduino-based embedded systems and wireless communication.

Environment has become another important application area for open-source hardware. Sunny [46] developed an indoor environment system integrating DHT22 sensors, particulate matter sensors (PM1.0, PM2.5, and PM10), and CO_2 sensors to evaluate indoor air quality.

Although these studies demonstrate the versatility of open-source hardware in agriculture, aquaculture, and environment, most developments have focused on specific applications or isolated monitoring systems. Comparatively fewer studies have reported the development of integrated educational platforms that combine customized electronic boards, multiple sensor technologies, and visual programming environments specifically designed for contextualized STEM education. Moreover, existing educational implementations generally rely on commercial development boards or generic programming environments without providing customized software extensions that facilitate direct interaction between students and sensor-based systems.

Overall, the reviewed literature confirms that open-source hardware, sensors, and IoT technologies provide an appropriate technological foundation for contextualized STEM education. However, a gap remains in the development of integrated educational prototypes that combine customized electronic boards, sensor technologies, and visual programming within a unified platform. This gap provides the technological motivation for the present study, which proposes the design, implementation, and functional verification of an educational prototype integrating three customized electronic boards and dedicated mBlock extensions to support contextualized STEM activities in authentic classroom environments.

2.4. Functional Verification of Educational Technology Prototypes in Real-World Environments

From a technological development perspective, one of the major challenges in STEM educational technologies is demonstrating that educational prototypes operate reliably under authentic application conditions. While numerous studies report pilot implementations or laboratory-based experiments, comparatively fewer investigations describe systematic functional verification processes carried out in real educational environments. Such verification is essential to confirm that hardware and software components operate correctly before educational technologies can be adopted in classroom settings [47].

Educational technology prototypes generally evolve through successive stages of technological maturity, progressing from conceptual design and laboratory testing toward implementation in operational environments. During these stages, prototypes must demonstrate stable operation, correct communication between hardware and software, and adequate performance under realistic conditions before broader educational deployment can be considered [48,49]. Consequently, functional verification represents a critical stage in the development of educational technologies because it provides evidence that the proposed system satisfies its intended technical requirements.

Functional verification of electronic prototypes can involve testing at different levels of system integration to confirm the correct operation of individual components and their interaction within the integrated system [50]. In educational electronic systems, these aspects are particularly important because failures in hardware or software integration may compromise classroom activities and reduce the usability of the proposed technology.

Beyond technical performance, implementing prototypes in authentic educational environments provides an opportunity to evaluate their operational feasibility under realistic classroom conditions. Factors such as classroom organization, availability of technological resources, user interaction, connectivity, and instructional time may influence prototype performance and reveal practical limitations that are not evident during laboratory testing [51]. Consequently, classroom deployment complements laboratory verification by demonstrating whether the proposed technology can be integrated into regular teaching activities.

Recent studies have emphasized the importance of systematically documenting the engineering process associated with educational technologies, including hardware design, software implementation, prototype verification, and classroom deployment. Such documentation contributes to prototype reproducibility, continuous technological improvement,

and future technology transfer to educational institutions [52]. Furthermore, systematic functional verification provides the technical evidence required to support subsequent developments and larger-scale educational implementations.

Although previous studies have reported educational applications based on open-source hardware, relatively few investigations describe the complete engineering workflow encompassing prototype design, implementation, customized software development, functional verification, and classroom deployment within a single educational platform. This limitation is particularly evident in systems integrating customized electronic boards, sensors, and visual programming interfaces. Therefore, the present study addresses this gap by presenting the design, implementation, and functional verification of an educational prototype that integrates customized electronic boards, sensor technologies, and mBlock-based visual programming to support contextualized STEM activities in an authentic university classroom environment.

Based on the functional-verification approaches identified in the reviewed literature, the verification protocol adopted in this study was structured to examine the prototype at complementary hardware, software, and integrated-system levels. At the hardware level, the procedure considered PCB continuity, power-architecture compatibility, and processor/firmware operation where documented. At the software level, the execution of the customized mBlock extensions was examined using the Environment Board as a representative implementation. Finally, integrated functional verification examined the interaction among the educational boards, sensors, customized programming blocks, and visual interface under the reported implementation conditions. This protocol is described in detail in Section 3.

3. Design and Implementation of the Proposed Educational Prototype

3.1. Overall Architecture

The proposed educational prototype was designed as an integrated technological platform that combines open-source hardware, customized educational electronic boards, sensors, and a block-based programming interface. The prototype was conceived to support contextualized STEM activities by enabling students to acquire environmental data, process sensor measurements, and develop technology-based solutions through visual programming. Figure 1 presents the overall architecture of the proposed educational prototype.

The prototype consists of two complementary components: a hardware component and a software component. The hardware component comprises three customized educational electronic boards designed for agriculture, aquaculture, and environment. Each board integrates a microcontroller, communication interfaces, environmental sensors, actuators, and visualization elements that enable students to acquire and analyze real-world data. The software component is based on the mBlock visual programming environment, version 5.4.3 (Makeblock Co., Ltd., Shenzhen, China), and incorporates customized extensions, graphical programming blocks, communication libraries, and device drivers that enable interaction between the educational boards and the connected sensors without requiring prior knowledge of text-based programming languages.

The development of the proposed prototype was supported by an extensive review of scientific publications, technical reports, and previous technological developments related to open-source hardware, educational electronics, and block-based programming environments, following the recommendations of Trilles et al. [53]. Furthermore, studies in Educational Robotics, Computational Thinking, and STEM education were analyzed to identify the functional and educational requirements of the proposed platform. In particular, the systematic review and meta-analysis reported by Fidai et al. [54] provided

useful evidence regarding the implementation of block-based programming technologies in educational contexts.

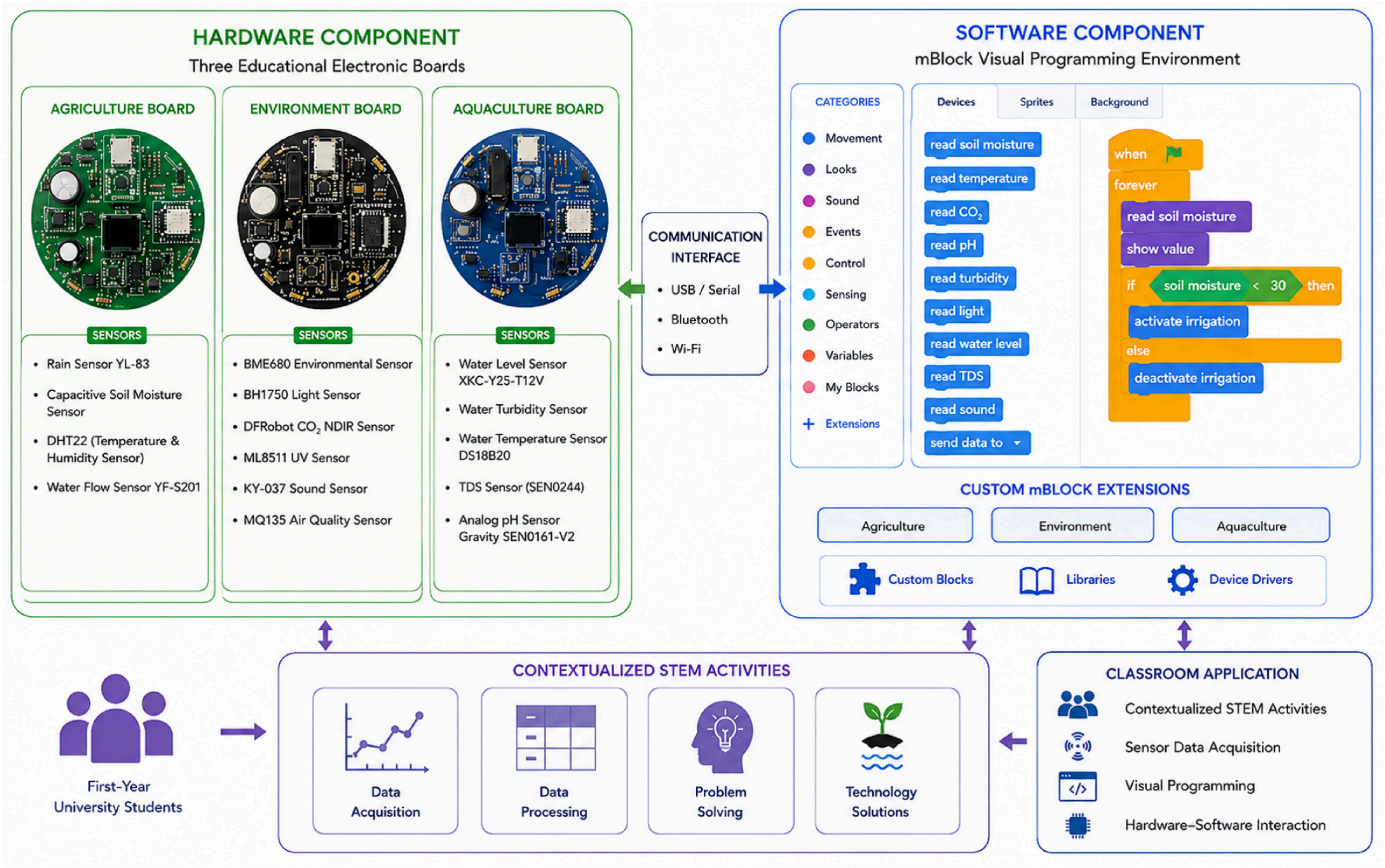


Figure 1. Overall architecture of the proposed educational hardware–software prototype for contextualized STEM activities.

The engineering process was organized through a structured sequence of design, implementation, integration, and functional verification activities. The Design Science Research (DSR) framework [55] served as a conceptual reference for the development of the technological artifact. For the hardware component, the development workflow comprised functional design, electronic design, PCB fabrication and assembly, and functional verification. For the software component, a sequential workflow comprising requirements specification, extension architecture design, coding and integration, and functional verification was adopted. These workflows provided a structured development process while allowing the prototype to be adapted to the requirements of contextualized STEM activities.

Based on this overall architecture, the following subsections describe the engineering process adopted for the development of the proposed educational prototype. Section 3.2 presents the hardware design and implementation, including functional design, electronic design, PCB fabrication, assembly, and operational testing. Section 3.3 describes the software architecture, the development of the customized mBlock extensions, and their functional verification.

3.2. Hardware Design and Implementation

The hardware component was developed following a structured engineering workflow comprising functional design, electronic design, PCB implementation, fabrication and assembly, and functional verification. The workflow was applied consistently to the three educational electronic boards developed for agriculture, environment, and aquaculture.

The hardware development process c”mpri’ed four sequential stages: (i) functional design, (ii) electronic design, (iii) PCB fabrication and assembly, and (iv) hardware functional verification. Figure 2 illustrates the engineering workflow adopted for the development of the proposed hardware platform.

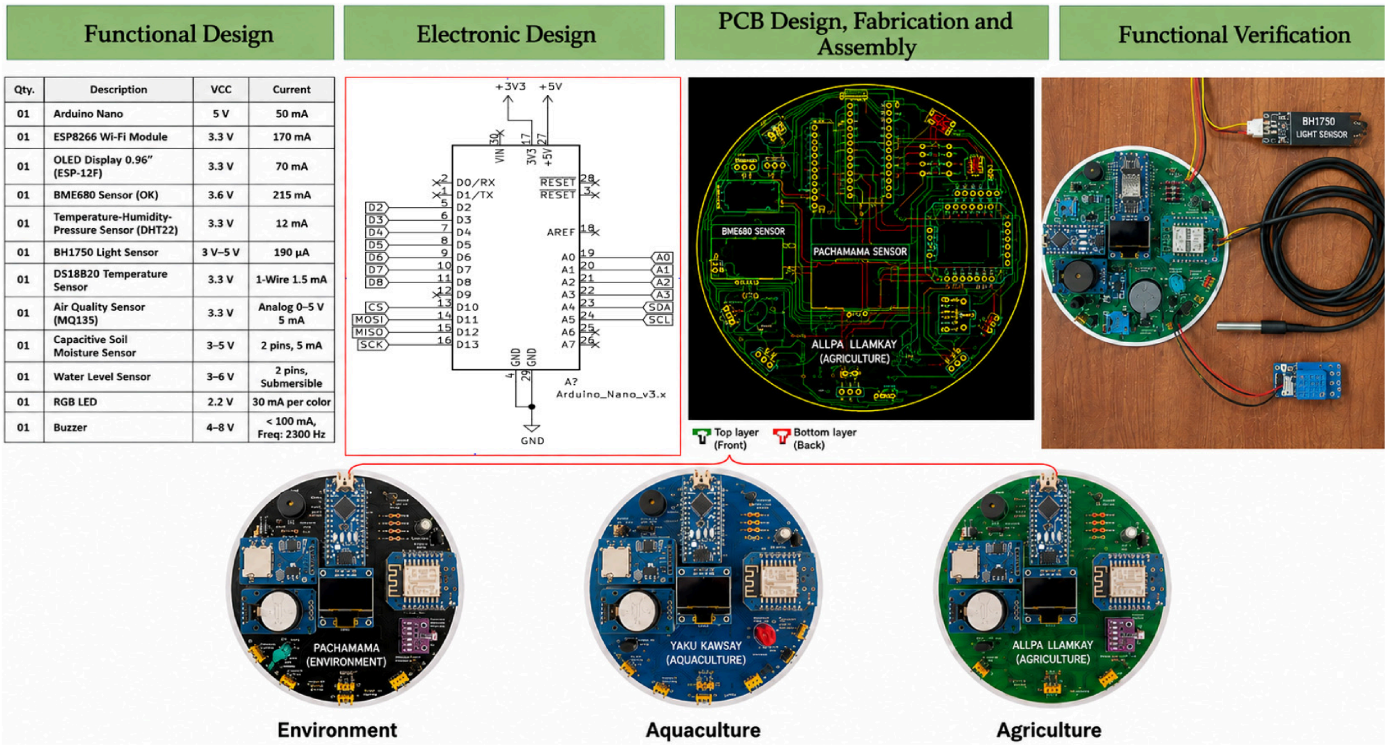


Figure 2. Hardware design, implementation, and functional verification workflow of the proposed educational prototype.

3.2.1. Functional Design

The functional design stage defined the technical requirements of the educational prototype according to the intended STEM applications in agriculture, environment, and aquaculture. At this stage, the physical variables to be measured, the sensing devices, communication interfaces, power requirements, and educational functionalities were identified before the electronic implementation of the system.

A modular hardware architecture was adopted to maximize flexibility and facilitate future expansion. Although the three educational boards were developed for different application domains, they share a common hardware architecture composed of a microcontroller, power regulation circuitry, communication interfaces, sensor connectors, display modules, and expansion ports. This modular approach simplifies maintenance, reduces development costs, and enables the integration of additional sensors for future educational applications.

The environment board was designed to acquire atmospheric variables, including temperature, relative humidity, atmospheric pressure, air quality, carbon dioxide concentration, ultraviolet radiation, sound level, and light intensity. The aquaculture board was intended to monitor water-quality parameters, such as water temperature, pH, and turbidity, whereas the agriculture board focused on soil moisture and environmental variables associated with crop monitoring. These functional specifications established the basis for the subsequent electronic design and hardware implementation stages.

3.2.2. Electronic Design

Following the definition of the functional requirements, the electronic design stage focused on developing the circuit architecture for each educational board. Electronic schematics were created by selecting and interconnecting the microcontroller, communication modules, environmental sensors, signal conditioning circuits, voltage regulation units, display devices, and peripheral components required for each application.

The electronic architecture was organized into six functional modules: processing, power management, communication, sensor acquisition, actuation, and visualization. This modular organization improved circuit readability, simplified debugging, and facilitated hardware scalability while ensuring electrical compatibility among all integrated components.

The electronic schematics and printed circuit board (PCB) layouts were developed during 2023 using the web-based EasyEDA electronic design automation platform (JLC Group, Shenzhen, China; https://easyeda.com/editor, accessed on 15 September 2026). During this process, electrical connections, pin assignments, communication buses (UART, I2C, and SPI), and power distribution were validated before PCB fabrication.

3.2.3. PCB Fabrication and Assembly

Based on the validated electronic schematics, dedicated printed circuit boards (PCBs) were designed for the three educational applications. The PCB layout considered component placement, routing optimization, connector positioning, power distribution, and accessibility of external interfaces to facilitate classroom experimentation.

After completing the PCB design, the boards were manufactured and assembled by integrating microcontrollers, wireless communication modules, environmental sensors, voltage regulators, connectors, passive components, and display modules. Each board was then inspected to verify soldering quality, electrical continuity, and mechanical integrity before functional testing.

The resulting hardware consists of three customized educational electronic boards dedicated to environment, aquaculture, and agriculture, respectively (Figure 2). The use of dedicated PCBs reduced the need for external wiring and provided a more integrated hardware configuration during classroom activities, thereby simplifying the physical interconnection of the electronic components. Furthermore, the modular design enables future hardware upgrades through the incorporation of additional sensors and communication modules.

3.2.4. Hardware Functional Verification

Following PCB fabrication and assembly, a systematic hardware functional verification procedure was conducted to confirm the electrical integrity and operational performance of the developed educational boards before software integration and classroom implementation. The verification process was designed to ensure that the hardware complied with the design specifications and operated correctly under normal conditions. The verification consisted of three complementary stages:

PCB Continuity Verification

The first stage evaluated the electrical continuity of the printed circuit board (PCB) using a digital multimeter (Fluke Corporation, Everett, WA, USA) operating in continuity mode. Measurements were performed on the 5 V, 3.3 V, and ground distribution traces to detect possible open circuits or unintended short circuits resulting from PCB fabrication or component assembly.

As illustrated in Figure 3, the continuity test confirmed the electrical integrity of the PCB traces, indicating that the power distribution network was correctly implemented before energizing the board.

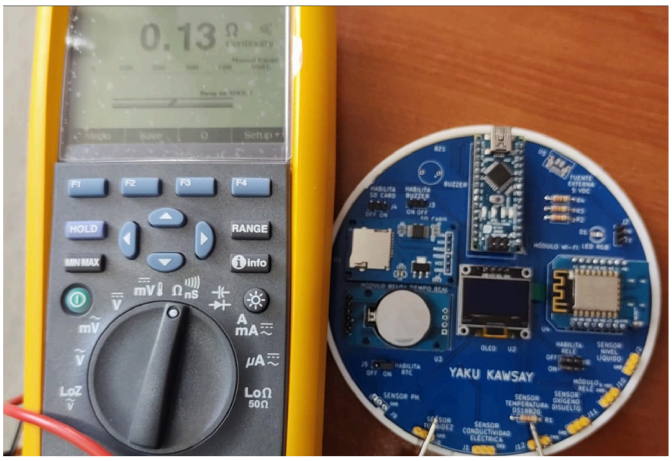


Figure 3. PCB continuity verification of the Aquaculture Board using a digital multimeter.

Power Supply Verification

The hardware functional verification included continuity/impedance checks of the PCB connections associated with the 5 V, 3.3 V, and ground lines using a digital multimeter. These tests were performed to identify unintended open circuits or short circuits in the assembled electronic boards. In addition, the power architecture was checked for compatibility with the specified operating-voltage ranges of the installed electronic components. The voltage values reported in Table 1 therefore correspond to the specified operating requirements of the components and should not be interpreted as a complete record of experimentally measured supply voltages.

Table 1. Specified operating-voltage requirements of the main electronic components used in the educational boards.

Electronic Device	Specified Operating Voltage/Range (VDC)
DS3231 Real-Time Clock	3.3–5.0 VDC
OLED Display	3.0–5.5 VDC
SD Card Module	3.3–5.0 VDC
Buzzer	3.3–8.0 VDC
Arduino Nano	5.0 VDC
ESP8266 Wi-Fi Module	3.3 VDC

The verification confirmed PCB continuity and the compatibility of the nominal 5 V and 3.3 V power architecture with the specified operating-voltage requirements of the installed components. A complete retained record of experimentally measured voltages at each component was not available.

Firmware Upload and Operational Verification

The final stage consisted of uploading the test firmware to the Arduino Nano microcontroller and verifying the correct operation of the embedded electronic modules.

As shown in Figure 4, the successful execution of the firmware confirmed the correct operation of the Arduino Nano, OLED display, ESP8266 communication module, SD-card interface, and peripheral circuits integrated into the educational board. This procedure also verified the communication between the microcontroller and the connected hardware before integrating the customized mBlock extensions.

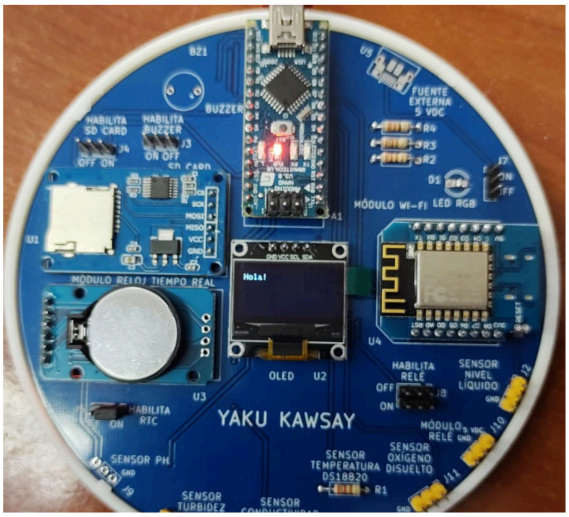


Figure 4. Firmware upload and operational verification of the Aquaculture Board.

The hardware verification procedures were successfully completed for the Agriculture Board, Environment Board, and Aquaculture Board. The results confirmed the electrical integrity of the printed circuit boards, correct power distribution, and proper operation of the embedded hardware components, providing a correct platform for the subsequent software implementation and classroom functional verification.

To improve the traceability of the hardware functional verification, the available laboratory test evidence was consolidated into a verification matrix, as shown in Table 2. The matrix summarizes the main hardware functions evaluated for the three educational boards, including PCB continuity/impedance checks, compatibility of the power architecture with the operating-voltage requirements of the installed components, and, where documented, processor and firmware operation. For each verification item, the test procedure, acceptance criterion, observed result, available test record, and corresponding outcome are reported.

Table 2. Hardware functional-verification matrix for the three educational boards.

Board	Function/ Requirement	Test Procedure	Acceptance Criterion	Observed Result	Available Test Record	Outcome
Agriculture Board	PCB electrical continuity	Continuity/impedance checks on 5 V, 3.3 V, and ground-related PCB connections using a digital multimeter	No unintended open or short circuits in the evaluated connections	PCB connections were checked according to the implemented circuit	Technical report, Figure S2 (p. 4)	Pass
Agriculture Board	Power architecture	Supply architecture checked against specified operating ranges of installed components	5 V and 3.3 V architecture compatible with component requirements	Compatibility with the specified component operating ranges was verified	Technical report, Table S1 (p. 4)	Pass
Environment Board	PCB electrical continuity	Continuity/impedance checks on 5 V, 3.3 V, and ground-related PCB connections using a digital multimeter	No unintended open or short circuits in the evaluated connections	PCB connections were checked according to the implemented circuit	Technical report, Section S1.2 (p. 5); individual photographic record not retained	Pass
Environment Board	Power architecture	Supply architecture checked against specified operating ranges of installed components	5 V and 3.3 V architecture compatible with component requirements	Compatibility with the specified component operating ranges was verified	Technical report, Table S2 (p. 6)	Pass

Table 2. *Cont.*

Board	Function/ Requirement	Test Procedure	Acceptance Criterion	Observed Result	Available Test Record	Outcome
Aquaculture Board	PCB electrical continuity	Continuity/impedance checks on 5 V, 3.3 V, and ground-related PCB connections using a digital multimeter	No unintended open or short circuits in the evaluated connections	PCB connections were checked according to the implemented circuit	Technical report, Figure S5 (p. 7)	Pass
Aquaculture Board	Power architecture	Supply architecture checked against specified operating ranges of installed components	5 V and 3.3 V architecture compatible with component requirements	Compatibility with the specified component operating ranges was verified	Technical report, Table S3 (p. 7)	Pass
Aquaculture Board	Processor and firmware operation	Arduino Nano powered and firmware uploaded during laboratory testing	Processor powers correctly and firmware executes	Arduino Nano powered and firmware execution was documented	Technical report, Figure S6 (p. 8)	Pass

The available technical records correspond to documented laboratory functional checks rather than to a repeated-trial reliability campaign. Consequently, complete repetition counts, failure rates, and long-duration reliability indicators cannot be reconstructed from the retained records. The results reported in Table 2 should therefore be interpreted as evidence of successful operational checks under the documented laboratory conditions and not as evidence of long-term reliability or system stability.

The retained laboratory records document the execution and outcome of the reported functional checks; however, a complete numerical record of the individual voltage measurements obtained during these tests was not retained. Therefore, no reconstructed experimental voltage values are reported. The reported verification is limited to the documented continuity/impedance checks, nominal power-architecture compatibility, and processor/firmware operation where applicable.

3.3. Software Component Implementation

The customized visual programming extensions were developed and tested using mBlock version 5.4.3. The extension-development process was performed using the mBlock 5 Extension Builder (Makeblock Co., Ltd., Shenzhen, China; https://ext.mblock.cc, accessed on 15 September 2026), which was used to define the customized programming blocks and generate the corresponding .mext extension files. The software architecture extends the mBlock programming environment through customized extensions, graphical programming blocks, communication libraries, and device drivers that allow students to interact with sensors and actuators without requiring prior knowledge of text-based programming.

The development focused on creating application-specific programming blocks for the three educational boards—environment, aquaculture, and agriculture—allowing learners to initialize sensors, acquire environmental data, control actuators, and visualize measurements through graphical commands. The software implementation incorporated customized C++ libraries and communication drivers that established interaction between the programming environment and the hardware platform [56].

Figure 5 illustrates the engineering workflow adopted for the development of the software component. The workflow comprises four sequential stages: requirements specification, block and extension design, coding and integration, and functional verification. This process enabled the systematic implementation of customized programming extensions fully compatible with the proposed educational hardware platform.

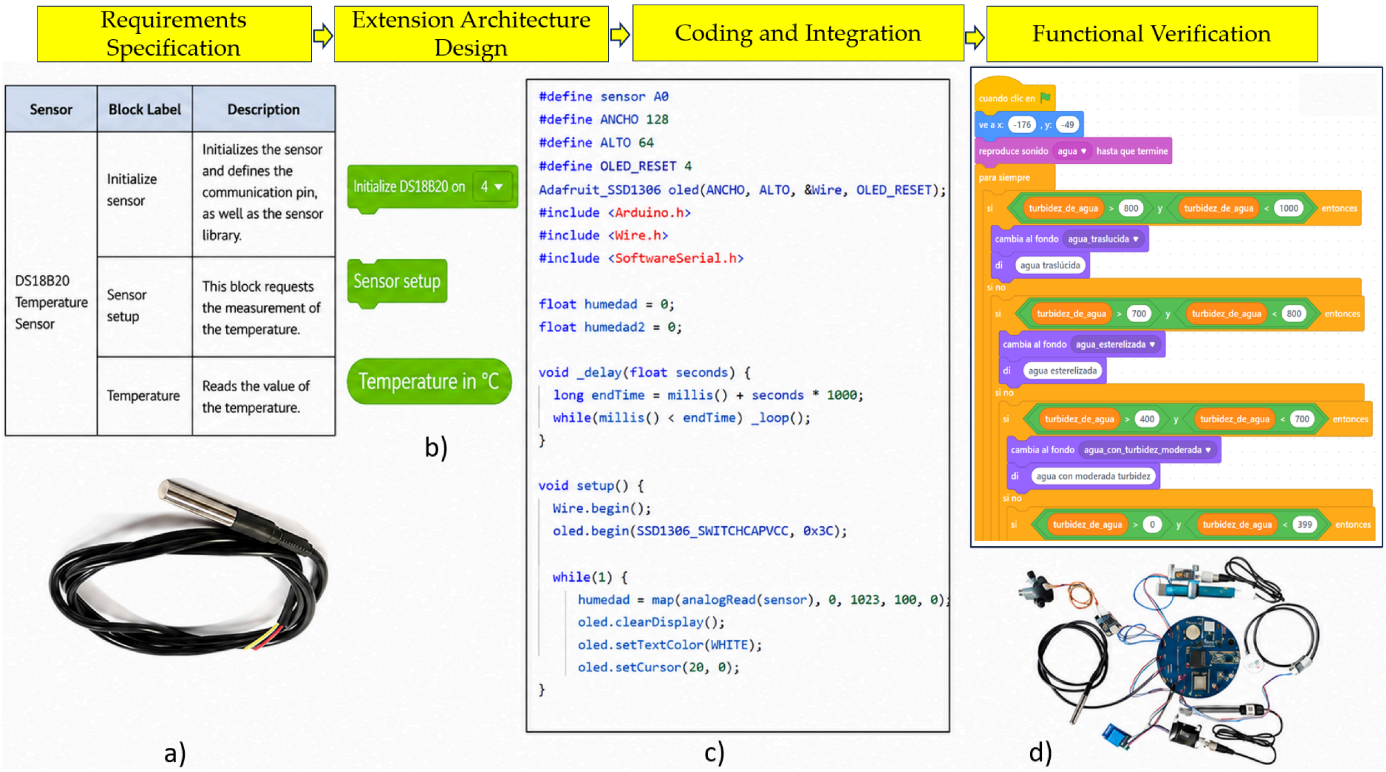


Figure 5. Software development workflow for the implementation and functional verification of customized mBlock extensions: (**a**) DS18B20 temperature sensor; (**b**) extension architecture design; (**c**) coding and integration; and (**d**) functional verification.

3.3.1. Requirements Specification

The first stage consisted of identifying the functional requirements of the software platform according to the educational objectives of the proposed STEM activities. Functional requirements included sensor initialization, environmental data acquisition, actuator control, serial communication, graphical data visualization, and compatibility with the three educational electronic boards.

Special attention was given to the design of intuitive graphical programming blocks that could be easily understood by first-year university students. Each programming block was associated with a specific hardware function, allowing users to configure sensors, acquire measurements, and control electronic devices through visual commands rather than text-based programming.

3.3.2. Extension Architecture Design

The software architecture was designed as a layered extension framework that enables seamless communication between the proposed educational hardware platform and the mBlock visual programming environment. Its primary objective is to provide an intuitive programming interface through customized graphical blocks that allow students to interact with sensors and electronic devices without requiring prior knowledge of text-based programming.

As shown in Figure 6, the architecture is organized into four functional layers that collectively support communication, data acquisition, and visual programming. This modular organization improves software maintainability, facilitates the integration of new sensors and educational boards, and promotes the reuse of software components across different STEM applications.

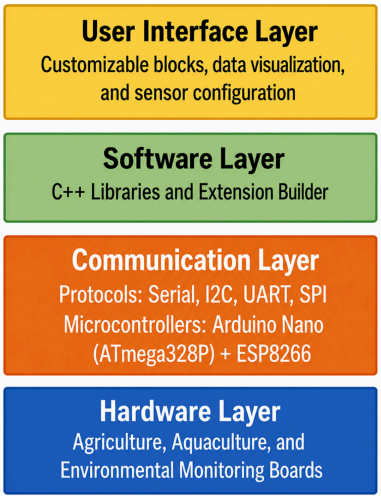


Figure 6. Architecture of the customized mBlock extension.

The Hardware Layer comprises the educational electronic boards and the associated sensors and actuators used in agriculture, aquaculture, and environment. These devices acquire physical and environmental variables and execute control actions through the embedded microcontroller, providing the physical interface between the educational platform and the surrounding environment.

The Communication Layer establishes correct data exchange between the hardware platform and the software environment through standard communication protocols, including Serial/UART, I^2C, and SPI. By abstracting the communication process, this layer enables efficient interaction between the microcontroller, peripheral devices, and the customized programming environment while ensuring compatibility among different hardware modules.

The Software Layer contains the customized extensions developed using the mBlock Extension Builder, together with C++ libraries, communication drivers, and configuration files required to support the educational boards. This layer translates graphical programming instructions into executable commands that initialize sensors, acquire measurements, process data, and control the hardware platform.

Finally, the User Interface Layer provides the graphical programming environment presented to students and teachers. Through customized programming blocks, users can configure sensors, acquire environmental data, visualize measurements in real time, and develop STEM applications without writing conventional source code. This layer simplifies interaction with the educational hardware and facilitates experimentation, programming, and problem-solving activities within the proposed learning environment.

3.3.3. Extension Coding

After defining the software architecture and the functional requirements of the programming blocks, the coding phase was carried out to implement the customized extensions for the three educational electronic boards. The extensions were developed using the mBlock Extension Builder framework and C++-based libraries, enabling communication between the visual programming environment and the sensors integrated into the agriculture, aquaculture, and environment boards.

The implementation process involved the development of customized programming blocks, communication drivers, sensor libraries, and data acquisition functions. Each block was designed to perform specific tasks, including sensor initialization, parameter configuration, data acquisition, signal processing, and visualization of environmental variables. This approach allowed students to interact with hardware devices through

graphical programming commands without requiring advanced knowledge of text-based programming languages.

Table 3 presents the programming blocks developed for the environment educational board. These customized blocks provide functions for sensor initialization, data acquisition, and real-time monitoring of variables such as temperature, relative humidity, atmospheric pressure, air quality, carbon dioxide concentration, ultraviolet radiation, light intensity, and sound levels.

The coding process also included the implementation of labels in Spanish to facilitate accessibility and contextualized learning. In addition, debugging and integration tests were conducted throughout the development process to ensure compatibility between the software extensions, the mBlock environment, and the educational hardware platform.

Table 3. Customized programming blocks implemented for the Environment Board.

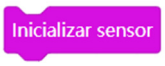
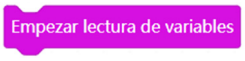
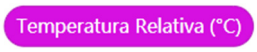
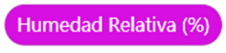
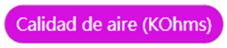
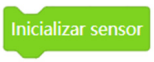
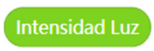
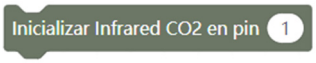
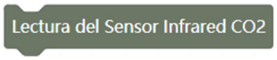
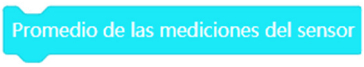
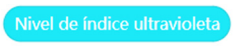
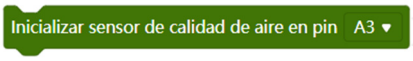
Sensor	Programming Block	Output Variable	Unit	Description
BME680 environmental sensor		-	-	Initializes the sensor, configures the communication interface, and loads the required libraries.
		Sensor measurements	-	Acquires the measurements provided by the BME680 sensor.
		Temperature	°C	Returns the measured temperature.
		Relative humidity	%	Returns the measured relative humidity.
		Atmospheric pressure	hPa	Returns the atmospheric pressure.
		Gas resistance	kΩ	Returns the gas-sensor resistance measured by the BME680.
		Estimated altitude	m	Returns the estimated altitude calculated from atmospheric pressure.
BH1750 light sensor		-	-	Initializes the sensor, configures the communication interface, and loads the required libraries.
		Illuminance	lx	Returns the measured light intensity.
DFRobot CO_2 NDIR Sensor		-	-	Initializes the sensor and configures the analog communication pin.
		Sensor output	-	Acquires the average sensor readings.
		Voltage	mV	Returns the analog output voltage.
		CO_2 concentration	ppm	Returns the CO_2 concentration value provided by the sensor-specific processing implemented in the extension. The measurement was not independently calibrated against a reference CO_2 instrument in the present study.

Table 3. *Cont.*

Sensor	Programming Block	Output Variable	Unit	Description
ML8511 UV sensor		-	-	Initializes the sensor and configures the analog communication pin.
		Sensor measurements	-	Acquires the average sensor readings.
		Voltage	V	Returns the analog output voltage.
		UV index	Index	Returns the calculated ultraviolet index.
		UV category	-	Returns the ultraviolet exposure category.
KY-037 sound sensor		-	-	Initializes the sensor, defines the analog communication pin, and calls the required sensor libraries.
		Analog sensor output	ADC value	Returns the analog output generated by the sound sensor.
		Time	ms	Returns the sampling time used during acquisition.
MQ135 air quality sensor		-	-	Initializes the sensor and configures the analog communication pin.
		Relative sensor response	ADC value	Returns the analog response of the MQ135 sensor. Quantitative gas concentration requires calibration.

As a result, a set of customized software extensions was obtained, enabling correct communication between the educational boards and the visual programming environment. These extensions support the implementation of STEM activities based on real-world data acquisition, experimentation, and problem-solving in educational contexts.

3.3.4. Functional Verification of Software Extensions with Sensors

The functional verification of the customized software extensions was performed using the Environment Board as a representative implementation example. This board was selected to illustrate the interaction between the customized mBlock blocks and multiple sensor interfaces within the proposed software architecture. The verification focused on the execution of the graphical programming blocks for sensor initialization, data acquisition, processing of sensor outputs where applicable, and visualization of the corresponding sensor responses within the mBlock environment.

For this representative verification, the customized programming blocks were executed using the BH1750 light sensor, DFRobot CO_2 NDIR sensor, ML8511 ultraviolet sensor, KY-037 sound sensor, and MQ135 air-quality sensor. The original manufacturer information for some of the commercially available generic sensor modules was not retained in the project records; therefore, these modules are identified by their model or functional designation. The procedure included sensor initialization, communication establishment, data acquisition, variable assignment, and visualization of the corresponding sensor outputs through the graphical programming interface. Figure 7 presents representative programming sequences used during this software-level functional verification.

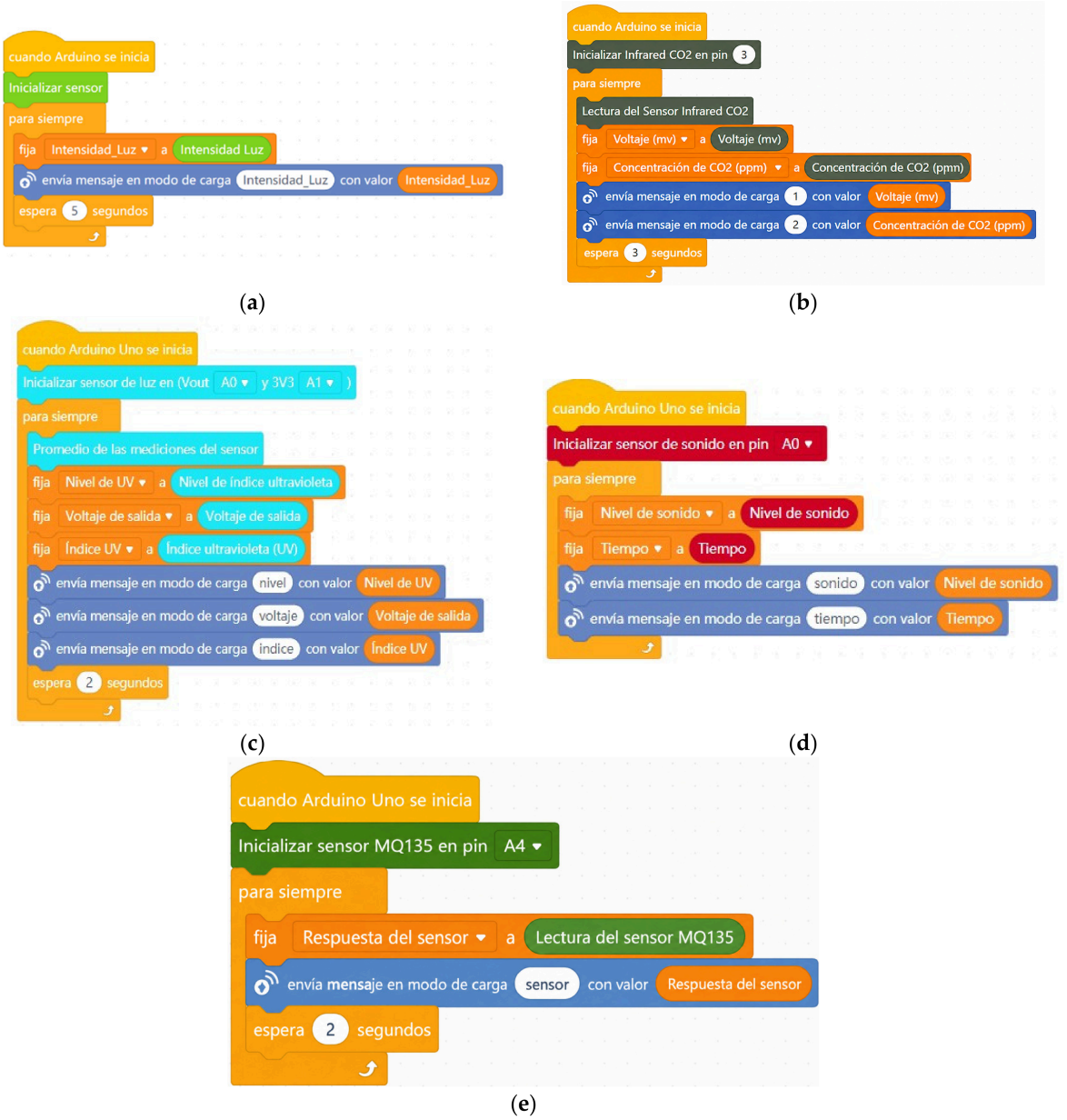


Figure 7. Functional verification of the software extensions using customized programming blocks for the Environment Board: (**a**) BH1750 light sensor; (**b**) DFRobot CO_2 NDIR sensor; (**c**) ML8511 ultraviolet sensor; (**d**) KY-037 sound sensor; and (**e**) MQ135 air-quality sensor.

As illustrated in Figure 7, each software extension was verified by executing representative programming sequences associated with the corresponding sensor. The tests confirmed the correct interaction between the customized programming blocks, the communication drivers, and the hardware platform, demonstrating correct acquisition and visualization of sensor data within the mBlock environment.

The functional verification demonstrated that the developed software extensions provide a stable interface between the visual programming environment and the Environment Board, allowing students to interact with environmental sensors through intuitive graphical blocks. This functionality supports the implementation of contextualized STEM activities involving environmental monitoring, experimentation, and technology-based problem solving without requiring conventional text-based programming.

The results obtained with the Environment Board provide a representative example of the operation of the customized mBlock extension framework with multiple sensor

interfaces. This software-level verification was intended to demonstrate the execution of the implemented programming blocks rather than to constitute an exhaustive verification of all sensors integrated into the three educational boards. The integrated operation of the Agriculture Board, Environment Board, and Aquaculture Board is addressed separately in Section 4.3.

4. Functional Verification of the Proposed Educational Prototype in a Real-World Classroom Environment

The classroom implementation involved 69 first-year Nursing students enrolled in the Introduction to the Experimental Method course at the Universidad Nacional Autónoma de Tayacaja Daniel Hernández Morillo (UNAT) during the 2025 academic year. The implementation was conducted over 16 academic weeks, comprising 16 weekly sessions of four hours each. Students worked collaboratively in groups of approximately five members under continuous instructor supervision and received ongoing technical and methodological guidance throughout the activities. One unit of each educational board—the Agriculture Board, Environment Board, and Aquaculture Board—was available for the classroom implementation and was shared among the student groups during the corresponding activities. Previous programming experience was not used as a participant-selection criterion and was not formally assessed as part of the present study.

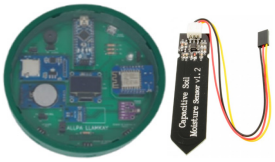
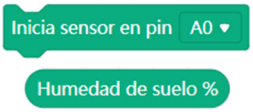
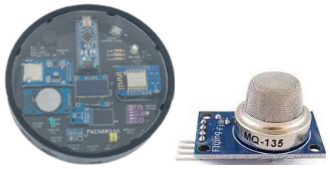
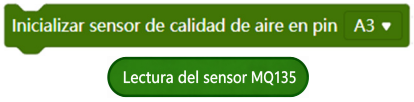
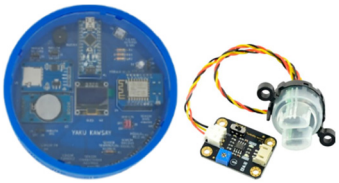
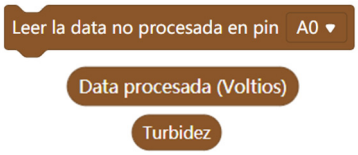
The classroom activities followed a project-based learning approach organized into four sequential phases: Problem Understanding, Activity Development, Activity Execution, and Solution Review. Contextualized activities addressing agriculture, environmental monitoring, and aquaculture were developed using the corresponding educational boards and associated sensors. During the Activity Execution phase, students were introduced to visual programming using mBlock and used the customized programming blocks to interact with the educational boards and their associated sensors. The block-based interface enabled the construction of programming sequences without requiring conventional text-based programming syntax.

During the classroom implementation, students initialized the sensors, acquired and visualized sensor data through the customized mBlock programming blocks, and interpreted the resulting sensor responses within the context of the corresponding activities. This implementation provided a real classroom scenario for examining the functional interaction among the Agriculture Board, Environment Board, Aquaculture Board, associated sensors, customized software extensions, and visual programming interface. In particular, the classroom activities provided an operational setting for verifying sensor data acquisition and visualization, execution of the customized programming blocks, and hardware–software interaction under the reported implementation conditions.

It should be emphasized that the classroom implementation was intended to examine the functional operation and practical feasibility of the proposed educational prototype under real classroom conditions rather than to evaluate learning outcomes or educational effectiveness. Consequently, no educational outcomes related to research competencies, computational thinking, STEM competencies, or other educational constructs are reported or interpreted as part of the functional-verification results presented in this study.

Table 4 summarizes the contextualized STEM activities implemented during the classroom verification process, together with the corresponding educational boards, sensors, and customized programming blocks.

Table 4. Contextualized STEM activities used for the functional verification of the proposed educational prototype.

Contextualized STEM Activity	Educational Board and Sensor	Customized Programming Blocks
Monitoring soil moisture for crop cultivation in a local agricultural area. Students acquired soil-moisture measurements using the Agriculture Board and analyzed the sensor output through the customized mBlock interface.	 Agriculture Board and Capacitive Soil Moisture Sensor	
Monitoring environmental conditions associated with smoke generated by agricultural burning. Students acquired MQ135 sensor responses and visualized the measurements using customized programming blocks.	 Environment Board and MQ135 Gas Sensor	
Monitoring water turbidity in local drinking-water sources. Students acquired turbidity measurements using the Aquaculture Board and analyzed the sensor output through the customized mBlock interface.	 Aquaculture Board and Water Turbidity Sensor.	

4.1. Development of Classroom Activities

Several classroom activities were implemented with first-year university students following a four-stage problem-solving methodology consisting of problem understanding, activity development, activity execution, and solution review. These activities were designed to verify the operational use of the proposed educational prototype while engaging students in contextualized STEM tasks involving environmental monitoring, sensor-based experimentation, and visual programming.

As an illustrative example, the activity entitled "Monitoring Water Turbidity in Drinking-Water Sources Using Sensors" is presented. Students used the Aquaculture Board, a water turbidity sensor, and the customized programming blocks developed for the mBlock environment. The activity was conducted under authentic classroom conditions to demonstrate the interaction between the educational board, the sensor, and the visual programming interface during the acquisition and visualization of environmental data.

Problem Understanding:

During this stage, students explored scientific information related to water quality and turbidity using academic databases, including Google Scholar and Scopus, together with ChatGPT GPT-4o, OpenAI, San Francisco, CA, USA) as a generative assistant for organizing and summarizing information. Information generated with ChatGPT was not used as an independent scientific source. Students were instructed to contrast AI-generated information with scientific publications retrieved from academic databases and search engines. The verification process involved comparing the main concepts and statements generated by the AI tool with information reported in the consulted scientific literature, under instructor supervision. Only information that could be supported by the consulted academic sources was incorporated into the classroom activities.

Students identified water turbidity as the study variable and analyzed published information describing factors associated with water quality. Based on this information, each group prepared a cause-and-effect diagram to organize the relationships reported in the literature.

As illustrated in Figure 8, the resulting diagram summarizes literature-based factors associated with water turbidity in drinking-water sources. Rather than representing measured relationships obtained during this study, the diagram synthesizes information reported in previous publications concerning possible causes of increased turbidity and its reported environmental and drinking-water quality implications. The literature references visible within the diagram correspond to sources consulted and incorporated by the students during the activity and are shown only as part of the original artifact. They should therefore not be interpreted as references used by the authors to support the findings or claims of the present study. This activity enabled students to identify the study variable, interpret scientific information, and define the technological problem addressed during the subsequent implementation stages.

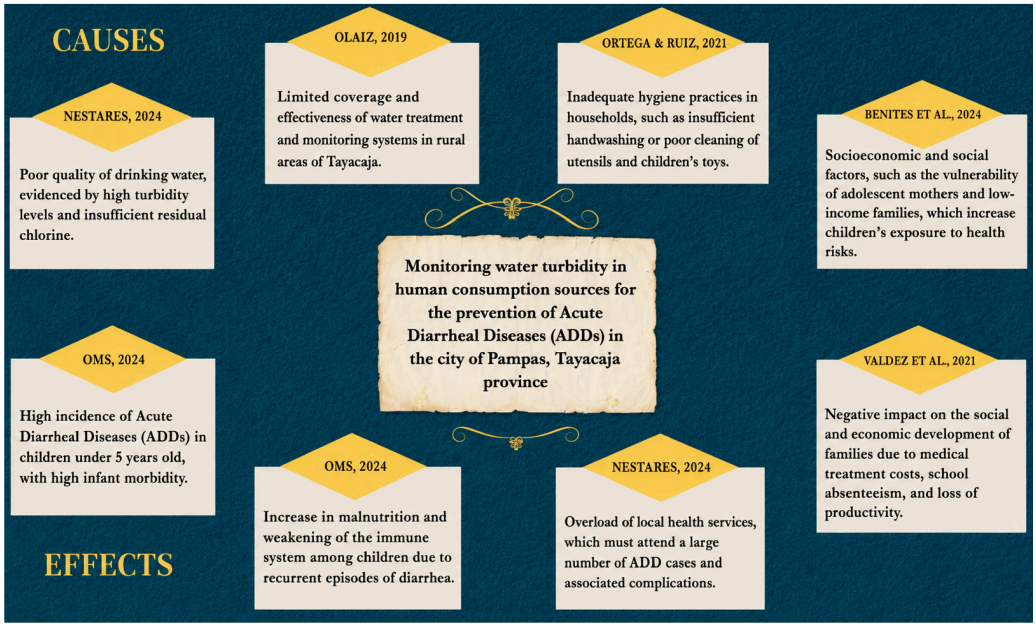


Figure 8. Cause-and-effect diagram developed by students from the literature review on water turbidity. The author–year references visible within the diagram are part of the students' original work and are reproduced as they appeared in the classroom artifact; they are not citations introduced by the authors of this manuscript.

Design Activities:

During this stage, students reviewed scientific and technological publications related to sensor-based water-quality monitoring and similar educational developments. Academic sources, including Google Scholar and Scopus, together with ChatGPT as a generative assistant, were consulted to identify existing technological solutions and implementation strategies. Information generated by ChatGPT was contrasted with scientific publications before being incorporated into the activity.

Based on the reviewed information, students defined the sequence of tasks required to implement the proposed educational prototype using the Aquaculture Board and the water turbidity sensor. The planned activities included:

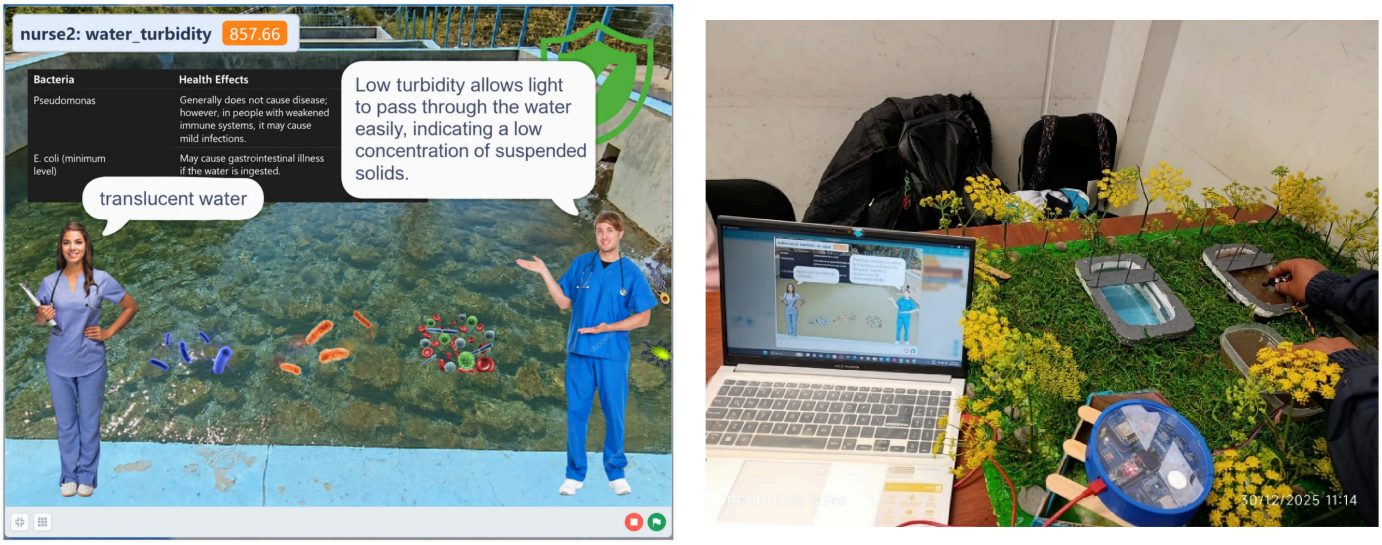
- Identifying representative locations associated with water quality in the Atocc water source.

- Selecting the electronic components required for the activity, including the Aquaculture Board and the water turbidity sensor.
- Developing a visual programming application in mBlock to acquire and display turbidity-related sensor data.
- Constructing a physical model representing the water distribution system.
- Integrating the hardware and software components to verify the acquisition and visualization of sensor data.

Implementation Activities:

During this stage, students implemented the planned activities and operated the proposed educational prototype using the Aquaculture Board, the water turbidity sensor, and the customized programming blocks developed for the mBlock environment.

As shown in Figure 9a, students developed a visual programming application that acquires sensor data and displays the corresponding variables through an interactive graphical interface. The application integrates customized characters, graphical elements, and explanatory messages to facilitate the interpretation of the acquired sensor readings during the classroom activity.



(**a**) (**b**)

Figure 9. Functional implementation of the proposed prototype: (**a**) customized mBlock application and (**b**) integration of the Aquaculture Board with the physical model.

Figure 9b illustrates the physical model developed by the students to simulate a local water distribution system. The prototype integrates the Aquaculture Board, the water turbidity sensor, the customized mBlock application, and water samples with different turbidity conditions. This implementation provided a classroom scenario for verifying the interaction between the hardware platform, the customized software extensions, and the visual programming interface during real-time data acquisition and visualization.

Solution Review:

During the final stage, each student group reviewed the operation of the proposed educational prototype by executing the customized programming blocks, acquiring sensor data, and observing the interaction among the Aquaculture Board, the water turbidity sensor, the physical model, and the mBlock application. Students compared the acquired measurements under different water conditions and examined the behavior of the integrated hardware and software components throughout the activity.

As illustrated in Figure 10, students demonstrated the operation of the proposed prototype by integrating the Aquaculture Board, the customized mBlock application, the water turbidity sensor, and the physical model developed during the implementation stage. This classroom activity provided an opportunity to verify the functional operation of the educational prototype under authentic classroom conditions and to identify practical aspects related to system operation, sensor integration, and possible improvements for future versions of the platform.



Figure 10. Classroom functional verification of the proposed prototype using the Aquaculture Board.

The review stage also enabled students to discuss the observed sensor responses, compare the acquired data with the expected behavior of the prototype, and reflect on the applicability of the proposed platform for supporting contextualized STEM activities based on environmental monitoring.

4.2. Integration of STEM Disciplines During the Functional Verification of the Educational Boards

Table 5 summarizes how the four STEM disciplines were integrated during the classroom implementation of the proposed educational prototype. The activities were developed using the Agriculture Board, Environment Board, and Aquaculture Board, allowing students to investigate contextualized problems, acquire real-time sensor data, develop visual programming applications in mBlock, construct physical models, and analyze the measurements obtained during the functional verification of the prototype. This interdisciplinary approach enabled the hardware platform, software extensions, and customized programming blocks to be used within authentic STEM learning activities under real classroom conditions.

Table 5. Integration of STEM disciplines during the functional verification of the Agriculture, Environment, and Aquaculture Boards.

STEM Discipline	Agriculture Board 	Environment Board 	Aquaculture Board 
Science	Search and analysis of information on soil moisture, plant water requirements, and spinach cultivation. Identification of soil moisture as the study variable and development of a cause-and-effect analysis associated with the agricultural problem.	Search and analysis of information on air quality, smoke generated by hillside burning, and environmental implications. Identification of air quality as the study variable and development of a cause-and-effect analysis associated with the environmental problem.	Search and analysis of information on water quality and turbidity. Identification of water turbidity as the study variable and development of a cause-and-effect analysis associated with the water-quality problem.
Technology	Use of the Agriculture Board, capacitive soil moisture sensor, and mBlock to acquire sensor data, visualize soil-moisture measurements, and display contextualized messages related to crop conditions.	Use of the Environment Board, MQ135 gas sensor, and mBlock to acquire sensor readings, visualize measurements, and display contextualized messages related to air-quality monitoring.	Use of the Aquaculture Board, water turbidity sensor, and mBlock to acquire sensor data, visualize turbidity measurements, and display contextualized messages related to water conditions.
Engineering	Design and construction of a physical model simulating soil-moisture monitoring for spinach cultivation. Selection of components, sensor integration, system assembly, and functional verification of the prototype.	Design and construction of a physical model representing an environment affected by smoke from hillside burning. Selection of components, sensor integration, system assembly, and functional verification of the prototype.	Design and construction of a physical model simulating the flow of water from the source to the local community. Selection of components, sensor integration, system assembly, and functional verification of the prototype.
Mathematics	Reading and interpretation of soil-moisture sensor responses, comparison of sensor-output ranges, and analysis of data variation during functional testing.	Reading and interpretation of MQ135 sensor readings, comparison of measurement ranges, and analysis of data variation during functional testing.	Reading and interpretation of turbidity-sensor responses, comparison of sensor-output ranges, and analysis of data variation during functional testing.

4.3. Integrated Functional Verification of the Proposed Educational Prototype

Following the independent verification of the hardware and software components, an integrated functional verification was conducted to evaluate the operation of the complete educational prototype in an authentic classroom environment. This stage focused on verifying the interaction among the educational boards, the connected sensors, the customized programming blocks, and the mBlock visual programming environment during the execution of contextualized STEM activities.

The verification involved the three educational boards developed in this study: the Agriculture Board, Environment Board, and Aquaculture Board. Each board was connected to its corresponding sensor and controlled through the customized programming blocks previously implemented in mBlock. During classroom activities, students initialized the sensors, acquired measurements, and visualized the corresponding variables in real time through the graphical programming interface.

For the Agriculture Board, the integrated verification showed functional interaction between the capacitive soil-moisture sensor and the customized programming blocks

responsible for sensor initialization and soil-moisture acquisition. The sensor response was successfully displayed in the mBlock interface, allowing students to observe changes in soil moisture under different experimental conditions.

For the Environment Board, the functional verification showed successful communication between the MQ135 gas sensor and the customized programming blocks. The relative MQ135 sensor response was acquired and displayed through the visual programming environment, demonstrating functional interaction between the hardware platform and the software extension. Because the MQ135 sensor was not calibrated against a reference gas concentration in this study, its output was interpreted as a relative sensor response rather than as a quantitative gas concentration in ppm.

For the Aquaculture Board, the customized programming blocks successfully acquired the raw sensor output and the corresponding processed voltage from the water turbidity sensor. These values were displayed in real time through the graphical interface, allowing students to observe and compare the sensor response obtained from different water samples during the classroom activities. Since a reference calibration procedure for quantitative turbidity determination was not performed in this study, the sensor output was interpreted as an indicative response rather than as a validated turbidity measurement in NTU.

Table 6 summarizes the integrated functional verification performed for the three educational boards. The results show successful interaction among the hardware platform, customized software extensions, and the visual programming environment under the reported test conditions. The prototype successfully executed sensor initialization, data acquisition, and visualization during the reported functional tests, providing evidence of functional operation of the integrated system under the evaluated conditions.

Table 6. Integrated functional verification of the educational boards and customized programming blocks.

Board	Function Tested	Test Condition/Procedure	Acceptance Criterion	Observed Result	Test-Record Availability	Outcome
Agriculture Board 	Soil-moisture sensor acquisition 	Sensor connected to the board and corresponding customized mBlock block executed	Sensor response acquired and displayed through mBlock	Sensor response successfully acquired and displayed.	Individual supporting record not retained; repetition count not retained.	Pass
Environment Board 	MQ135 sensor acquisition 	MQ135 sensor connected to the board and corresponding customized mBlock block executed	Relative sensor response acquired and displayed through mBlock	Sensor response successfully acquired and displayed.	Individual supporting record not retained; repetition count not retained.	Pass
Aquaculture Board 	Turbidity-sensor acquisition 	Turbidity sensor connected and corresponding customized mBlock block executed	Raw sensor output and corresponding processed voltage acquired and displayed through mBlock	Sensor response successfully acquired and displayed.	Individual supporting record not retained; repetition count not retained.	Pass

Table 6. *Cont.*

Board	Function Tested	Test Condition/Procedure	Acceptance Criterion	Observed Result	Test-Record Availability	Outcome
Agriculture Board 	Hardware–software integration	Board, sensor, and customized extension operated together	Successful communication, acquisition, and visualization	Communication between the board, soil-moisture sensor, and customized mBlock extension was established, and the sensor response was acquired and displayed.	Individual supporting record not retained; repetition count not retained.	Pass
Environment Board 	Hardware–software integration	Board, sensors, and customized extension operated together	Successful communication, acquisition, and visualization	Communication between the board, connected sensors, and customized mBlock extension was established, and the sensor responses were acquired and displayed.	Individual supporting record not retained; repetition count not retained.	Pass
Aquaculture Board 	Hardware–software integration	Board, sensor, and customized extension operated together	Successful communication, acquisition, and visualization	Communication between the board, turbidity sensor, and customized mBlock extension was established, and the sensor response was acquired and displayed.	Individual supporting record not retained; repetition count not retained.	Pass

Following the laboratory-level hardware verification described in Section 3.2.4, integrated functional verification was performed to determine whether the educational boards, connected sensors, customized mBlock extensions, and visual programming interface operated together as intended. The verification focused on sensor initialization, data acquisition, communication with the corresponding educational board, execution of the customized programming blocks, and visualization of the resulting sensor response.

The Pass outcomes reported in Table 6 indicate that the predefined functional acceptance criteria were met during the corresponding classroom executions. However, individual supporting records and repetition counts for these executions were not retained. Accordingly, these outcomes should be interpreted as reported functional checks under the implementation conditions rather than as independently auditable repeated-test results.

A systematic numerical record of the individual sensor readings obtained during each classroom execution was not retained. Therefore, the reported results are limited to the functional responses observed during the reported activities, and no reconstructed measurement values are presented.

From an engineering perspective, the proposed architecture differs from a conventional educational setup based on a general-purpose microcontroller board connected to multiple independent sensor modules. In a conventional configuration, sensors and communication modules are typically connected individually through external wiring and configured through separate libraries or programming routines. In the proposed prototype, the required sensor interfaces are organized around three application-oriented electronic boards—the Agriculture Board, Environment Board, and Aquaculture Board—while the corresponding software functions are exposed through customized mBlock programming blocks. This integration provides a unified hardware–software interface for sensor data acquisition and visualization during the implemented STEM activities. However, the present

study did not conduct a controlled quantitative comparison with conventional Arduino-based modular configurations in terms of cost, assembly time, reliability, or robustness. Therefore, no claims of superiority regarding these parameters are made.

5. Discussion

The functional-verification results demonstrate the integration of three application-oriented educational boards, sensor interfaces, and block-based visual programming within a unified hardware–software architecture. The Agriculture Board, Environment Board, and Aquaculture Board supported sensor-data acquisition and interaction with customized mBlock programming blocks under the reported implementation conditions. These results are consistent with previous studies showing the flexibility of open-source microcontroller platforms for integrating sensors, communication modules, and application-specific electronic components [35–37]. However, whereas previous developments have commonly focused on individual monitoring applications or specific hardware configurations, the present prototype integrates three application-oriented educational boards and their corresponding visual programming functions within a common technological architecture.

The application-oriented organization of the prototype is also consistent with the broad use of open-source hardware and sensors reported in agriculture, environmental monitoring, and aquaculture. Previous studies have employed Arduino-based systems for monitoring soil moisture, temperature, environmental variables, and agricultural conditions [38–42]. Similarly, Arduino-based platforms have been applied to water-quality and aquaculture monitoring using sensors for variables such as temperature, pH, total dissolved solids, water level, and other water-quality parameters [43–45]. Environmental monitoring systems have likewise integrated multiple sensors with embedded platforms for the acquisition of air-quality and environmental data [46]. In contrast to these domain-specific implementations, the present work organizes agriculture, environment, and aquaculture applications as three components of a single educational hardware–software platform.

The customized mBlock extensions represent another relevant component of the proposed architecture. Block-based programming environments have been reported as suitable tools for introducing programming and computational thinking to users with limited computing backgrounds [28–30]. In the present prototype, the visual programming interface was extended beyond the use of generic programming blocks by incorporating customized functions for interaction with the developed electronic boards and associated sensors. The Environment Board was used as a representative case for software-level functional verification, demonstrating the execution of customized blocks for sensor initialization, data acquisition, processing where applicable, and visualization of sensor responses. At the integrated-system level, the three educational boards were operated with their corresponding hardware and software components during the reported classroom activities.

The classroom implementation provided an additional operational context for examining the prototype beyond laboratory-level hardware checks. Previous research has emphasized that the implementation of technological prototypes under realistic conditions can reveal operational factors that may not be evident during laboratory testing and can contribute to assessing their practical feasibility [47–52]. In the present study, the prototype was implemented over 16 academic weeks with 69 first-year Nursing students working collaboratively in groups of approximately five members. The activities provided a real classroom setting for examining sensor-data acquisition, execution of the customized programming blocks, and hardware–software interaction. Nevertheless, these observations should be interpreted as evidence of functional operation and practical feasibility under the reported conditions rather than as evidence of educational effectiveness.

Several limitations should be considered when interpreting the results. First, the available functional-verification records correspond primarily to documented operational checks rather than to a repeated-trial reliability campaign; consequently, failure rates, long-duration stability, and engineering reliability were not established. Second, complete metrological validation of all integrated sensors was outside the scope of this study. Sensor outputs for which reference calibration was not performed should therefore be interpreted as indicative responses under the evaluated conditions rather than as validated quantitative measurements. Third, the classroom implementation was conducted in a single course and institutional setting, limiting the generalization of the implementation conditions to other educational contexts. Fourth, only one unit of each educational board was available and shared among the student groups. Finally, the proposed dedicated-board architecture was not quantitatively compared with conventional microcontroller-and-sensor-module configurations in terms of cost, assembly time, maintainability, reliability, or classroom robustness.

Future work should include repeated functional test campaigns, long-duration operation and stress testing, sensor characterization and calibration where quantitative physical measurements are required, and controlled engineering comparisons with conventional modular configurations. Additional studies should examine the scalability of the hardware–software architecture, the incorporation of new sensors and communication technologies, and its implementation across different courses and educational contexts. Educational effectiveness should be investigated separately through appropriately designed studies and validated assessment instruments.

6. Conclusions

This study presented the design, implementation, and functional verification of an educational prototype with a visual programming interface for contextualized STEM activities. The proposed prototype integrates three application-oriented educational boards—the Agriculture Board, Environment Board, and Aquaculture Board—with sensor interfaces, open-source hardware platforms, and customized programming blocks developed for the mBlock environment.

The functional-verification process provided technical evidence of prototype operation at the hardware, software, and integrated-system levels under the reported conditions. Hardware-level verification included PCB continuity/impedance checks, assessment of the compatibility of the power architecture with the operating-voltage requirements of the installed components, and documented firmware operation. At the software level, the Environment Board was used as a representative implementation to verify the execution of customized mBlock programming blocks and their interaction with multiple sensor interfaces. At the integrated-system level, the three educational boards operated with their corresponding sensors and customized software components, enabling the acquisition and visualization of sensor data associated with soil moisture, air-quality monitoring, and water turbidity.

The principal contribution of this work is the integration of application-oriented educational boards, sensor interfaces, and block-based visual programming within a unified hardware–software architecture. The classroom implementation further demonstrated the practical feasibility of operating the prototype during contextualized activities related to agriculture, environmental monitoring, and aquaculture. These findings are limited to the functional operation and practical feasibility of the prototype under the reported laboratory and classroom conditions and should not be interpreted as evidence of educational effectiveness, metrological performance, long-term reliability, or superiority over conventional modular microcontroller configurations.

Overall, the proposed prototype provides a functional technological platform for the implementation of contextualized sensor-based STEM activities through visual programming. The results establish a technical basis for subsequent studies involving extended functional testing, sensor characterization, engineering comparisons, scalability, and the separate evaluation of educational outcomes across different educational contexts.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/electronics15184329/s1.

Author Contributions: Conceptualization, R.P.C., R.F.U.T., C.A.-D. and A.C.N.; methodology and formal analysis, R.P.C., R.F.U.T. and A.C.N.; investigation, R.P.C., R.F.U.T. and A.C.N.; resources and data curation, R.F.U.T., R.P.C. and C.A.-D.; writing—original draft preparation, R.P.C. and R.F.U.T.; project administration and funding acquisition, R.P.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: The classroom implementation was conducted during the 2025 academic year as part of standard academic activities. Subsequently, the Research Ethics Committee of the Universidad Nacional Autónoma de Tayacaja Daniel Hernández Morillo issued an Ethical Exemption Letter on 20 August 2026, confirming that formal ethical approval was not required for the study because the activities involved no clinical, biomedical, or invasive procedures and the reported data were collected anonymously and voluntarily.

Informed Consent Statement: Participation in the reported data collection was voluntary and followed an opt-in approach. All participants were students enrolled in the Introduction to the Experimental Method course and agreed to participate in the study. The collected data were handled anonymously. Photographs included in the manuscript were anonymized by pixelating students' faces to prevent participant identification.

Data Availability Statement: The technical resources retained from the development of the prototype, including the PCB design files, Gerber files, bill of materials, customized mBlock extension files (.mext), and associated source-code resources, are available from the corresponding author upon reasonable request. These materials are not currently deposited in a public repository.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zawacki-Richter, O.; Marín, V.I.; Bond, M.; Gouverneur, F. Systematic review of research on artificial intelligence applications in higher education—Where are the educators? *Int. J. Educ. Technol. High. Educ.* **2019**, *16*, 39. [CrossRef]
2. Contente, J.; Galvão, C. STEM Education and Problem-Solving in Space Science: A Case Study with CanSat. *Educ. Sci.* **2022**, *12*, 251. [CrossRef]
3. Fadillah, M.A.; Hirahmah, A.; Asrizal; Usmeldi. The Effect of STEM-Based Interventions on Learning Outcomes in High School: A Meta-Analysis. *J. Innov. Phys. Teach.* **2024**, *2*, 102–112. [CrossRef]
4. Sarı, U.; Çelik, H.; Pektaş, H.M.; Yalçın, S. Effects of STEM-focused Arduino practical activities on problem-solving and entrepreneurship skills. *Australas. J. Educ. Technol.* **2022**, *38*, 140–154. [CrossRef]
5. Osborne, J.; Simon, S.; Collins, S. Attitudes towards science: A review of the literature and its implications. *Int. J. Sci. Educ.* **2003**, *25*, 1049–1079. [CrossRef]
6. Thibaut, L.; De Loof, H.; De Meester, J.; Goovaerts, L. Integrated STEM Education: A Systematic Review of Instructional Practices in Secondary Education. *Eur. J. STEM Educ.* **2018**, *3*, 2. [CrossRef] [PubMed]
7. Maass, K.; Geiger, V.; Romero, M.; Goos, M. The Role of Mathematics in interdisciplinary STEM education. *ZDM-Math. Educ.* **2019**, *51*, 869–884. [CrossRef]
8. Zhang, J.; Liu, J. Construction of scaffolding instruction mode for mblock for arduino maker course based on design thinking. In Proceedings of the ACM International Conference Proceeding Series, Hohhot, China, 22–24 October 2018; pp. 1–6. [CrossRef]
9. Salvador-Carulla, L.; Woods, C.; de Miquel, C.; Lukersmith, S. Adaptation of the technology readiness levels for impact assessment in implementation sciences: The TRL-IS checklist. *Heliyon* **2024**, *10*, e29930. [CrossRef] [PubMed]

10. Martynenko, O.O.; Pashanova, O.V.; Korzhuev, A.V.; Prokopyev, A.I.; Sokolova, N.L.; Sokolova, E.G. Exploring attitudes towards STEM education: A global analysis of university, middle school, and elementary school perspectives. *Eurasia J. Math. Sci. Technol. Educ.* **2023**, *19*, em2234. [CrossRef] [PubMed]
11. Kareem, J.; Thomas, R.S.; Nandini, V.S. A Conceptual Model of Teaching Efficacy and Beliefs, Teaching Outcome Expectancy, Student Technology Use, Student Engagement, and 21st-Century Learning Attitudes: A STEM Education Study. *Interdiscip. J. Environ. Sci. Educ.* **2022**, *18*, e2282. [CrossRef]
12. Rayner, G.M.; Papakonstantinou, T. Employer perspectives of the current and future value of STEM graduate skills and attributes: An Australian study. *J. Teach. Learn. Grad. Employab.* **2015**, *6*, 100–115. [CrossRef]
13. Mcdonald, C.V. STEM Education: A review of the contribution of the disciplines of science, technology, engineering and mathematics. *Sci. Educ. Int.* **2016**, *27*, 530–569.
14. Dugger, W.E. Science, technology, engineering, and mathematics (STEM) as a learning approach to improve 21st century skills: A review. *Int. J. Online Biomed. Eng.* **2020**, *16*, 95–104. [CrossRef]
15. Masalimova, A.R.; Khairullina, E.R.; Lapidus, N.I.; Orekhovskaya, N.A.; Zheltukhina, M.R.; Baranova, E.A. Exploring Preservice STEM Teachers' Smartphone Addiction. *Contemp. Educ. Technol.* **2022**, *14*, ep371. [CrossRef] [PubMed]
16. Capraro, R.; Slough, S. *STEM Project-Based Learning: An Integrated Science, Technology, Engineering, and Mathematics (STEM) Approach*; Sense Publishers: Rotterdam, The Netherlands, 2013; pp. 1–5.
17. Bruce, B.C.; Levin, J.A. Educational technology: Media for inquiry, communication, construction, and expression. *J. Educ. Comput. Res.* **1997**, *17*, 79–102. [CrossRef]
18. Dugger, W. Evolution of STEM in the United States. In Proceedings of the 6th Biennial International Conference on Technology Education Research in Australia, Queensland, Australia, 8–11 December 2010; pp. 117–123.
19. Burbaité, R.; Drasuté, V.; Stuikys, V. Integración of Computational Thinking Skills in STEM-Driven Computer Science Education. In Proceedings of the 2018 IEEE Global Engineering Education Conference (EDUCON), Santa Cruz de Tenerife, Spain, 17–20 April 2018; Volume 1, pp. 1824–1832. [CrossRef]
20. Master, A.; Cheryan, S.; Moscatelli, A.; Meltzoff, A.N. Programming experience promotes higher STEM motivation among first-grade girls. *J. Exp. Child Psychol.* **2017**, *160*, 92–106. [CrossRef] [PubMed]
21. Herschbach, D.R. The STEM Initiative: Constraints and Challenges. *J. STEM Teach. Educ.* **2011**, *48*, 96–122. [CrossRef]
22. Makeblock. "mBlock—One-Stop Coding Platform for Teaching and Learning," mBlock. Available online: https://mblock.cc/ (accessed on 5 December 2021).
23. Wang, M.-T.; Degol, J.L. Gender Gap in Science, Technology, Engineering, and Mathematics (STEM): Current Knowledge, Implications for Practice, Policy, and Future Directions. *Educ. Psychol. Rev.* **2017**, *29*, 119–140. [CrossRef] [PubMed]
24. Smith, K.; Maynard, N.; Berry, A.; Stephenson, T.; Spiteri, T.; Corrigan, D.; Mansfield, J.; Ellerton, P.; Smith, T. Principles of Problem-Based Learning (PBL) in STEM Education: Using Expert Wisdom and Research to Frame Educational Practice. *Educ. Sci.* **2022**, *12*, 728. [CrossRef]
25. Weese, J.L. Mixed methods for the assessment and incorporation of computational thinking in K-12 and higher education. In *Proceedings of the 2016 ACM Conference on International Computing Education Research, Melbourne, Australia, 8–12 September 2016*; Association for Computing Machinery, Inc.: New York, NY, USA, 2016; pp. 279–280. [CrossRef]
26. Sarmiento, M.I. Lenguajes y Entornos de Programación para Fortalecer El Desarrollo de Competencias Concernientes al Pensamiento Computacional. *Hamut'Ay* **2020**, *7*, 86. [CrossRef]
27. Makeblock Education. "What is mBlock-Python Editor?," Makeblock. Available online: https://python.mblock.cc/ (accessed on 2 April 2022).
28. Arduino. "What is Arduino?," Arduino. Available online: https://www.arduino.cc/ (accessed on 5 December 2021).
29. Harangus, K.; Kátai, Z. Computational thinking in secondary and higher education. *Procedia Manuf.* **2020**, *46*, 615–622. [CrossRef]
30. Romero, M.; Lepage, A.; Lille, B. Computational thinking development through creative programming in higher education. *Int. J. Educ. Technol. High. Educ.* **2017**, *14*, 42. [CrossRef]
31. Fronza, I.; Corral, L.; Pahl, C. Combining block-based programming and hardware prototyping to foster computational thinking. In *SIGITE 2019, Proceedings of the 20th Annual Conference on Information Technology Education, Tacoma, WA, USA, 3–5 October 2019*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 55–60. [CrossRef]
32. Suárez, A.; García, D.; Martínez, P.A.; Martos, J. Contribución de la robótica educativa en la adquisición de conocimientos de matemáticas en la Educación Primaria. *Magister Rev. Miscelánea Investig.* **2018**, *30*, 9–20. [CrossRef]
33. Daniela, L.; Lytras, M.D. Educational Robotics for Inclusive Education. *Technol. Knowl. Learn.* **2018**, *24*, 219–225. [CrossRef]
34. Berland, M.; Wilensky, U. Comparing Virtual and Physical Robotics Environments for Supporting Complex Systems and Computational Thinking. *J. Sci. Educ. Technol.* **2015**, *24*, 628–647. [CrossRef]
35. Barbosa, O.; Dias, S. Using Shields as Educational Tools for Firmware Development on Engineering Courses. In Proceedings of the Workshop on Sustainable Computing and Technology Reuse, São Paulo, Brazil, 26 November 2025; pp. 13–16. [CrossRef]

36. Rahman, M.K.K.; Subashini, M.M.; Nasor, M.; Tawfik, A. Development of Bio-Shields for Arduino Uno. In *2018 Advances in Science and Engineering Technology International Conferences (ASET)*; IEEE: New York, NY, USA, 2018; pp. 1–5. [CrossRef]
37. Plaza, P.; Sancristobal, E.; Carro, G.; Castro, M.; Blazquez, M.; Garcia-Loro, F. Lighting through educational robotics. In Proceedings of the 2018 Technologies Applied to Electronics Teaching, TAEE 2018, La Laguna, Spain, 20–22 June 2018. [CrossRef]
38. Bhadani, P.; Vashisht, V. Soil Moisture, Temperature and Humidity Measurement Using Arduino. In *2019 9th International Conference on Cloud Computing, Data Science and Engineering*; IEEE: New York, NY, USA, 2019; pp. 567–571.
39. Prasojo, I.; Maseleno, A.; Tanane, O.; Shahu, N. Design of automatic watering system based on arduino. *J. Robot. Control (JRC)* **2020**, *1*, 55–58. [CrossRef]
40. Beyaz, A.; Gül, V. Determination of Low-Cost Arduino Based Light Intensity Sensors Effectiveness for Agricultural Applications. *Braz. Arch. Biol. Technol.* **2022**, *65*, e22220172. [CrossRef]
41. Tuysuzoglu, G.; Birant, D.; Kiranoglu, V. Soil Temperature Prediction via Self-Training: Izmir Case. *Tarim Bilim. Derg.* **2022**, *28*, 47–62. [CrossRef]
42. Botero-Valencia, J.S.; Mejia-Herrera, M.; Pearce, J.M. Low cost climate station for smart agriculture applications with photovoltaic energy and wireless communication. *HardwareX* **2022**, *11*, e00296. [CrossRef]
43. Saha, S.; Hasan, R.; Kabir, S. IoT Based Automated Fish Farm Aquaculture Monitoring System. In *2018 2nd International Conference onon Innovations in Science, Engineering and Technology (ICISET)*; IEEE: New York, NY, USA, 2018; pp. 201–206.
44. Riansyah, A.; Mardiati, R.; Effendi, M.R.; Ismail, N. Fish feeding automation and aquaponics monitoring system base on IoT. In *Proceedings of the 2020 6th International Conference on Wireless and Telematics, ICWT 2020, Yogyakarta, Indonesia, 3–4 September 2020*; IEEE: New York, NY, USA, 2020. [CrossRef]
45. Sarwar, A.; Iqbal, M. IoT-Based Real-Time Aquaculture Health Monitoring System. *Eur. J. Electr. Eng. Comput. Sci.* **2022**, *6*, 44–50. [CrossRef]
46. Sunny, A.I.; Zhao, A.; Li, L.; Kanteh Sakiliba, S. Low-cost IoT-based sensor system: A case study on harsh environment. *Sensors* **2021**, *21*, 214. [CrossRef] [PubMed]
47. Hijón-Neira, R.; Connolly, C.; Pizarro, C.; Pérez-Marín, D. Prototype of a Recommendation Model with Artificial Intelligence for Computational Thinking Improvement of Secondary Education Students. *Computers* **2023**, *12*, 113. [CrossRef]
48. Saldivia, B.S.; Calderón, J.M. Digital Technologies in Service Learning for the Formal Education of the New Millennium Citizenship. *RIED-Rev. Iberoam. Educ. Distancia* **2020**, *23*, 129–148. [CrossRef]
49. Vidal, D. Factores que intervienen en el uso de las plataformas educativas. *Rev. Educ. Super.* **2020**, *1*, 81–88. [CrossRef]
50. Walsh, S.; Murphy, D.; Doyle, M.; Reilly, J.; Thompson, J.; Dunwoody, R.; Erkal, J.; Finneran, G.; Fontanesi, G.; Mangan, J.; et al. Development of the EIRSAT-1 CubeSat through functional verification of the engineering qualification model. *Aerospace* **2021**, *8*, 254. [CrossRef]
51. Anderson, T.; Shattuck, J. Design-based research: A decade of progress in education research? *Educ. Res.* **2012**, *41*, 16–25. [CrossRef]
52. Fishman, B.J.; Penuel, W.R.; Allen, A.-R.; Cheng, B.H.; Sabelli, N. Design-Based Implementation Research: An Emerging Model for Transforming the Relationship of Research and Practice. *Natl. Soc. Study Educ.* **2013**, *112*, 136–156.
53. Trilles, S.; Monfort-Muriach, A.; Cueto-Rubio, E.; Granell, C.; Juan, P. Sucre4Stem: Internet of things in classrooms. In *Proceedings of the 15th International Conference of Technology, Learning and Teaching of Electronics, TAEE 2022—Proceedings, Teruel, Spain, 29 June–1 July 2022*; IEEE: New York, NY, USA, 2022; pp. 6–9. [CrossRef]
54. Fidai, A.; Capraro, M.M.; Capraro, R.M. 'Scratch'-ing computational thinking with Arduino: A meta-analysis. *Think. Ski. Creat.* **2020**, *38*, 100726. [CrossRef]
55. Steuck, P.-F.; Gottschewski-Meyer, P.O.; Bierschwale, D.; Knackstedt, R.; Steuck, P. Questionnaires in Design Science Research—A Systematic Overview Questionnaires in Design Science Research-a Systematic Overview Research Paper. 2024. Available online: https://www.researchgate.net/publication/382088080 (accessed on 6 February 2025).
56. Makeblock. "My Blocks Category.". Available online: https://support.makeblock.com/hc/en-us/articles/15235802242199-Configuration-Configure-Blocks (accessed on 20 April 2026).

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.