Eduweb, 2026, enero-marzo, v.20, n.1. ISSN: 1856-7576
Doi: https://doi.org/10.46502/issn.1856-7576/2026.20.01.17

Original Article

Technology integration with Arduino and block-based coding to support preservice science teachers' self-efficacy in electrical circuits

Integración de la tecnología con Arduino y programación basada en bloques para fortalecer la autoeficacia de los futuros docentes de ciencias en circuitos eléctricos

Emre Kırtoklu[1] **Mustafa Ergun**[2]

Afiliación autor:
[1] M.Sc. in Science Education, Ondokuz Mayıs University, Science Education, Graduate School of Education, Turkey. ek1905@icloud.com
[2] PhD in Chemistry Education, Ondokuz Mayis University, Associate Professor, Faculty of Education, Samsun, Turkey. (Corresponding author) mergun@omu.edu.tr

Recibido: 05/01/26
Aceptado: 05/03/26

Cómo citar:
Kırtoklu, E., & Ergun, M. (2026). Technology integration with Arduino and block-based coding to support preservice science teachers' self-efficacy in electrical circuits. *Revista Eduweb, 20*(1), 285-302. https://doi.org/10.46502/issn.1856-7576/2026.20.01.17

Abstract

This study explores the effects of integrating block-based coding and Arduino into preservice science teacher education from an information and communication technologies (ICT) perspective. The study, conducted with 37 preservice science teachers enrolled in a science teacher education program at a state university in Türkiye, adopted an explanatory sequential mixed methods design. Quantitative data were collected using achievement tests in algorithmic thinking, basic electronics, Arduino, and mBlock, along with a programming self-efficacy perception scale. Qualitative data were obtained through semi-structured interviews to explain the quantitative findings. Results indicated significant improvements in participants' learning outcomes, sustained achievement on retention tests, and notable increases in programming self-efficacy. Qualitative findings revealed that block-based coding and Arduino were perceived as effective ICT tools for material design, active participation, and the concretization of abstract science concepts, despite some initial technical challenges. Overall, the study highlights the pedagogical potential of ICT supported coding environments in strengthening both the cognitive and affective dimensions of preservice science teacher training.

Keywords: Arduino, block-based coding, electrical circuits, preservice science teacher education, programming self-efficacy.

Resumen

Este estudio explora los efectos de la integración de la programación basada en bloques y Arduino en la formación inicial de profesores de ciencias desde una perspectiva de tecnologías de la información y la comunicación (TIC). El estudio se llevó a cabo con 37 futuros profesores de ciencias matriculados en un programa de formación docente en una universidad pública de Türkiye y adoptó un diseño de métodos mixtos secuencial explicativo. Los datos cuantitativos se recopilaron mediante pruebas de rendimiento sobre pensamiento algorítmico, electrónica básica, Arduino y mBlock, así como una escala de percepción de autoeficacia en programación. Los datos cualitativos se obtuvieron a través de entrevistas semiestructuradas con el fin de explicar los resultados cuantitativos. Los resultados mostraron mejoras significativas en el rendimiento académico de los participantes, así como una retención del aprendizaje a largo plazo y un aumento notable en las percepciones de autoeficacia en programación. Los hallazgos cualitativos indicaron que la programación basada en bloques y Arduino fueron percibidos como herramientas TIC eficaces para el diseño de materiales didácticos, la participación activa y la concreción de conceptos científicos abstractos, a pesar de algunos desafíos técnicos iniciales. En conjunto, el estudio destaca el potencial pedagógico de los entornos de programación apoyados por TIC en el fortalecimiento de las dimensiones cognitivas y afectivas de la formación inicial del profesorado de ciencias.

Palabras clave: Arduino, autoeficacia en programación, circuitos eléctricos, formación inicial del profesorado de ciencias, programación basada en bloques.

https://revistaeduweb.org/
ISSN: 1856-7576
Open Acces

Introduction

Block-based coding has emerged as an accessible alternative to text-based programming languages, which often pose significant entry barriers for novice learners. Through visual, drag-and-drop interfaces, environments such as Scratch, mBlock, and Code.org enable learners to engage with core programming concepts without the cognitive burden imposed by complex syntax (Dat et al., 2025; Rachmatullah & Wiebe, 2023). Research consistently shows that block-based coding supports motivation, problem-solving, and algorithmic thinking, while also fostering positive affective outcomes such as engagement and confidence (Coşkunserçe, 2023; Nannim et al., 2024; Şentürk & Sari, 2025).

For preservice science teachers, block-based coding represents not only a technical competence but also a pedagogical resource for designing inquiry-oriented and technology-integrated learning environments. Science topics such as electricity are often taught through abstract representations that students find difficult to conceptualize. Coding-enabled physical computing tools, such as Arduino, allow teacher candidates to transform these abstractions into observable and manipulable phenomena through circuit construction, testing, and experimentation. In this respect, block-based coding supports key competencies expected of contemporary science teachers, including designing instructional materials, integrating digital tools with curricular goals, and facilitating active student participation. Compared with text-based programming, block-based environments lower barriers to entry for novices and provide early experiences of success, which may encourage sustained engagement with technology-integrated science teaching.

The present study is theoretically grounded in self-efficacy theory, which emphasizes individuals' beliefs about their capabilities to organize and execute the actions required to achieve specific performance outcomes. Within teacher education, programming self-efficacy is particularly critical because it influences whether preservice teachers are willing to engage with educational technologies, persist when technical difficulties arise, and transfer newly acquired skills into instructional practice. Block-based coding environments are well suited to fostering self-efficacy by providing incremental task structures, immediate feedback, and opportunities for collaboration. When combined with hands-on circuit construction, such environments are also aligned with constructivist and experiential learning perspectives, in which knowledge is actively built through experimentation and reflection. From this perspective, programming self-efficacy functions as a key psychological mechanism linking coding experiences to pedagogical readiness in science education.

Despite the increasing integration of digital technologies in teacher education, preservice science teachers often experience difficulties in transforming technological knowledge into pedagogically meaningful classroom practices. In particular, while block-based coding and Arduino are widely used as instructional tools, there is limited evidence on how such experiences contribute to preservice teachers' programming self-efficacy and their ability to design instructional materials within specific science content areas such as electrical circuits. Therefore, this study approaches digital technologies not as ends in themselves, but as pedagogical tools that support instructional design and teacher learning. This study was conducted within a science teacher education program at a state university in Türkiye, where preservice teachers participated in a structured, practice based training process focusing on technology integrated instruction in electrical circuits.

Eduweb
https://revistaeduweb.org/
ISSN: 1856-7576
Open Acces

Theoretical Backgroud and Literature Review

Block-Based Coding in Teacher Education

Research in teacher education indicates that block-based coding environments positively influence preservice teachers' programming skills, attitudes, and confidence. Studies report that engagement with tools such as Scratch, LEGO Mindstorms, and mBlock enhances programming competence while increasing preservice teachers' self-efficacy beliefs (Bower et al., 2017; Sanusi et al., 2025). Compared with text-based programming, block-based approaches are also associated with reduced anxiety and cognitive load, making them particularly suitable for novice learners in teacher education contexts (Jeon & Kwon, 2024; Ga & Chang, 2025; Weintrop & Wilensky, 2017). However, recent research suggests that technology oriented teacher training often remains at a theoretical level and does not always translate into teachers' confidence for classroom implementation, underscoring the importance of hands-on, practice-based approaches in teacher education (Hur, 2021; Tsai, 2023).

Beyond technical skill development, block-based coding has been linked to broader pedagogical outcomes, including increased creativity, motivation, and interest in STEM-oriented instruction, particularly when tasks involve designing meaningful products or addressing real-world problems (Şentürk & Sari, 2025). Despite these advantages, challenges related to limited curriculum integration and insufficient instructional time persist in teacher education programs (Bower et al., 2017). These findings suggest that the pedagogical impact of block-based coding depends on its systematic integration into teacher training curricula.

Programming Self-Efficacy and Teacher Training

Programming self-efficacy has been identified as a critical factor shaping preservice teachers' engagement with coding and their willingness to integrate technology into teaching practice (Sun & Zou, 2024). Empirical studies show that block-based coding can significantly enhance perceptions of self-efficacy by fostering early success and reducing fear of failure (Bower et al., 2017; Hur, 2021; Şentürk & Sari, 2025; Wawire et al., 2025). Mashishi & Ramaila (2024) found that integrating blok-based coding into science instruction promoted active learning by increasing student engagement and fostering essential skills such as critical thinking and collaboration among preservice teachers. Similarly, robotic coding tasks positively affected teacher candidates' confidence and instructional readiness (Muşlu Kaygisiz et al., 2020).

Recent syntheses of the literature further emphasize that while computational thinking and programming are increasingly incorporated into teacher education, many studies lack coherent pedagogical frameworks that explicitly connect programming practices with disciplinary learning and affective outcomes such as self-efficacy (Rodrigues et al., 2025; Yuana et al., 2025). This gap highlights the need for instructional designs that integrate programming not as an isolated skill, but as a meaningful component of science teaching that supports both cognitive and motivational dimensions of learning.

Arduino, mBlock, and Science Education

A growing body of research has examined the use of Arduino and block-based interfaces such as mBlock in science education, particularly in preservice teacher training. Coding supported activities have been shown to help concretize abstract

https://revistaeduweb.org/
ISSN: 1856-7576
Open Acces

scientific concepts and facilitate meaningful learning experiences (Hsu et al., 2018; Xu et al., 2023). Scratch and mBlock based practices are also reported to enhance student participation and diversify instructional strategies in science classrooms (Ga & Chang, 2025).

Studies focusing on Arduino-based applications report generally positive outcomes alongside notable challenges (Sarı et al., 2022). In preservice science teacher education, Arduino supported activities have been associated with improved attitudes toward programming, problem-solving skills, and technology integration (Fjukstad et al., 2018). However, findings regarding self-efficacy are not uniformly positive. Türkoğuz & Sefer (2019) found that although Arduino-supported chemistry experiments did not significantly enhance preservice science teachers' domain-specific self-efficacy, they increased confidence in using information technologies. In physics education, Bulus Kirikkaya & Basaran (2019) demonstrated that Arduino-supported electrical experiments positively influenced attitudes toward laboratory work and technology integration.

Nevertheless, several studies indicate that preservice teachers initially experience difficulties due to limited prior knowledge, technical complexity, or resource constraints (Vasconcelos & Kim, 2020; Woo & Faloon, 2025; Yucelyigit, 2023). These challenges often decrease with experience, leading to more favorable dispositions toward technology integration over time (Kozcu Cakir & Guven, 2026).

Research Gap

Taken together, existing studies demonstrate that Arduino and mBlock supported activities positively influence engagement, attitudes, and learning outcomes in teacher education (Hur, 2021). Nevertheless, important gaps remain. Many studies examine coding or robotics without situating interventions within a specific science content domain, such as electrical circuits. Although self-efficacy is frequently mentioned as an outcome, it is rarely examined as a central psychological mechanism linking coding experiences to pedagogical readiness. Finally, research integrating block-based coding and Arduino in preservice science teacher education often relies on single-method designs, limiting explanatory depth regarding how and why learning gains occur.

To address these gaps, the present study focuses on preservice science teachers' experiences with block-based coding and Arduino in the context of teaching electrical circuits. By adopting an explanatory sequential mixed methods design, the study examines changes in programming self-efficacy and learning outcomes, as well as the experiential factors that help explain these changes.

The Purpose of the Study

This study aims to examine changes in preservice science teachers' (PSTs) programming self-efficacy and learning outcomes related to algorithmic thinking, basic electronics, Arduino, and mBlock following a structured block-based coding and Arduino training program. In addition, it explores PSTs' views on integrating mBlock- and Arduino-supported activities into science teaching, particularly for designing instructional materials on electrical circuits.

Accordingly, the study addresses the following research questions:

https://revistaeduweb.org/
ISSN: 1856-7576
Open Acces

RQ1. Is there a significant change in PSTs' perceptions of their programming self-efficacy after the training?
RQ2. Is there a significant improvement in PSTs' learning outcomes related to algorithmic thinking, basic electronics, Arduino, and mBlock after the training?
RQ3. What are PSTs' views regarding the pedagogical use of mBlock and Arduino in science teaching?

Methodology

Research Design

Using a mixed-methods framework, this study examines PSTs' self-efficacy perceptions regarding block-based coding and their views on its integration into science education. Mixed methods research integrates quantitative and qualitative approaches to provide a more comprehensive understanding of educational phenomena by drawing on the complementary strengths of each paradigm (Creswell, 2006).

An explanatory sequential mixed methods design was employed. In this design, quantitative data are collected and analyzed first, followed by qualitative data to explain and contextualize the quantitative findings (Creswell & Plano Clark, 2007). Accordingly, interview questions were structured to explore participants' experiences of the training process, their interpretations of learning gains, and the challenges or affordances encountered during the design of circuit-based instructional materials.

In the quantitative phase, a single group pre and posttest quasi-experimental design was used. No control group was included, which constitutes a limitation to internal validity. To mitigate potential threats to validity, such as maturation and testing effects, multiple achievement measures, a retention test, and complementary qualitative data were employed to triangulate. Quantitative findings were therefore interpreted alongside qualitative evidence to enhance the study's overall validity. This design was considered appropriate as the study was conducted within an existing instructional setting where random assignment and the formation of a control group were not feasible, and the primary aim was to examine changes in participants' outcomes following a structured pedagogical intervention.

Participants

The participants consisted of 37 PSTs enrolled in a science teacher education program at a state university in Türkiye. Convenience sampling was used, and participation was voluntary, as the study was conducted within the context of an elective course. The sample size was considered appropriate for an explanatory sequential mixed methods design, as it enabled meaningful quantitative analyses while supporting in-depth qualitative exploration.

Data Collection Tools

Quantitative Instruments

Several achievement tests developed by the researchers were used to assess PSTs' knowledge of algorithmic thinking, basic electronics, Arduino, and mBlock. These included the Basic Algorithm Achievement Test (BAAT), Arduino Achievement Test (AAT), mBlock Achievement Test (MBT), and Basic Electronics Achievement Test (BEAT). In addition, programming self-efficacy was measured

https://revistaeduweb.org/
ISSN: 1856-7576
Open Acces

with the Programming Self-Efficacy Perception Scale (PSEPS), translated into Turkish by Altun & Mazman (2012). All instruments were reviewed by experts in science education and educational measurement to ensure content relevance and item clarity.

BAAT initially consisted of 20 items (13 multiple-choice and 7 open-ended) designed to assess core algorithmic skills such as flowchart construction, symbol recognition, and basic algorithmic reasoning. Following a pilot study of 60 university students, five low-discrimination items were removed, resulting in a 15-item test. Item discrimination indices ranged from 0.34 to 0.62 the mean item difficulty was 0.51 the KR-20 reliability coefficient was 0.78.

AAT was developed to measure foundational knowledge of Arduino and its components. After pilot testing with 60 participants who had prior Arduino experience, three items with low discrimination were excluded. The final 10-item test yielded discrimination indices between 0.39 and 0.57, a mean difficulty of 0.54, and a KR-20 reliability coefficient of 0.64.

MBT comprised 10 multiple-choice items assessing participants' knowledge of mBlock and its symbols. The test was refined from an initial 12-item version following pilot testing with 60 students familiar with mBlock. Item discrimination indices ranged from 0.48 to 0.66, with a mean difficulty index of 0.51 and a KR-20 reliability coefficient of 0.75.

BEAT was designed to assess basic electronics knowledge required for Arduino applications. After pilot testing with 60 participants, four items with low discrimination were removed from the initial 14-item pool. The final 10-item test showed discrimination indices between 0.52 and 0.68, a mean difficulty of 0.62, and a KR-20 reliability coefficient of 0.71.

PSTs' programming self-efficacy was measured using the 9-item PSEPS developed by Altun & Mazman (2012). The scale consists of two sub-dimensions and employs a 7-point Likert scale ranging from 1 (I do not trust myself at all) to 7 (I completely trust myself). Altun & Mazman (2012) reported a Cronbach's alpha of .928 for the overall scale. For the present study, minor item revisions were made to align with the research context by replacing general programming references with block-based coding expressions consistent with the instructional approach. The revised version was reviewed by an expert before administration.

Qualitative Instrument

Semi-structured interviews were conducted during the qualitative phase of the study. Interview questions were adapted from Doğanay (2018) and revised in accordance with the objectives of the current research. Expert opinion from a faculty member specializing in science education was obtained to ensure the content validity and clarity of the interview questions.

Data Collection Process

Over an 11 week period, 37 PSTs participated in the study through a weekly two-hour course. Participants received theoretical instruction, completed achievement tests, and carried out practical applications and assignments throughout the process.

https://revistaeduweb.org/
ISSN: 1856-7576
Open Acces

Initially, participants were introduced to basic electronics, algorithmic thinking, and flowchart design, after which pre- and post-tests were administered. In the subsequent weeks, Arduino and mBlock training sessions were conducted, and participants engaged in hands-on coding and circuit design activities. As part of data gathering, PSTs were tasked with designing instructional materials for their future courses using mBlock and Arduino. Specifically, they were instructed to construct both series and parallel circuits on a breadboard using three different resistors (330 Ω, 1 kΩ, and 10 kΩ) to demonstrate variations in bulb brightness. For the task, PSTs had to assemble the circuits and program them to function correctly. Each resistor was configured in both series and parallel circuits, and all circuits were successfully implemented and tested. During the final weeks, retention tests and assignments were completed. Figure 1 below presents an instructional material constructed by PSTs that illustrates the LED brightness in a series circuit with a 1 kΩ resistor.

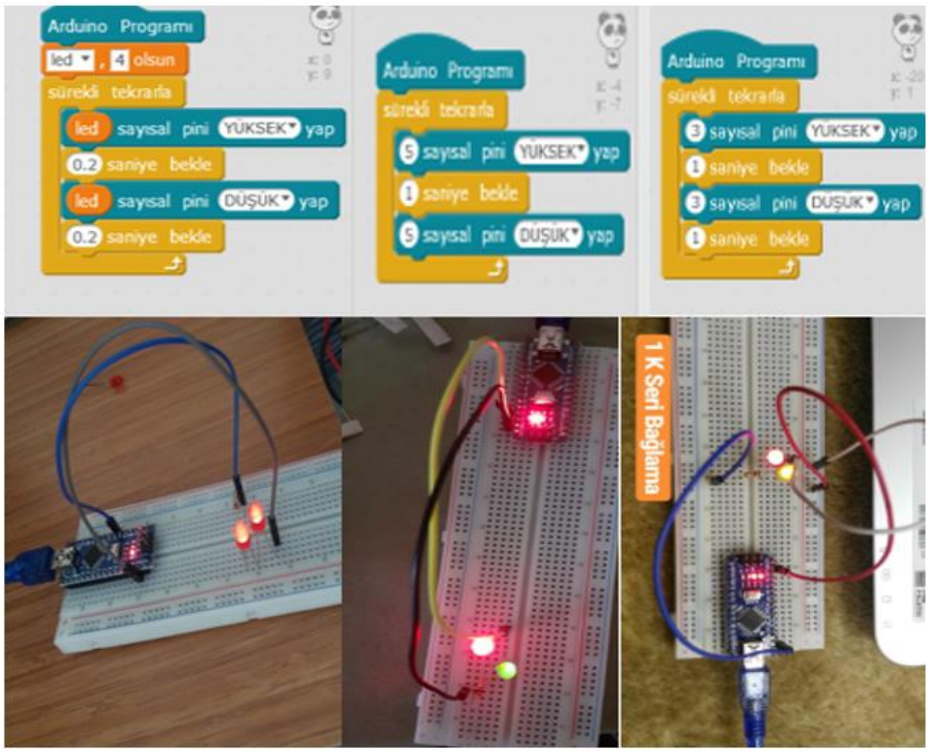

Figure 1. LED Brightness in a Series Circuit with a one kΩ Resistor

Implementation Process

The pedagogical intervention was implemented over an 11 week period within the scope of a practice based course. The process was structured progressively to support PSTs' development of both conceptual understanding and instructional design skills related to electrical circuits. The researcher conducted all lessons, applications, and subsequent activities in accordance with the week-by-week implementation plan (Table 1).

In the initial phase (Weeks 1–3), participants were introduced to the study and completed pre tests. This was followed by foundational instruction on basic electronics, including circuit components, breadboard usage, and connection methods. In the second phase (Weeks 4–5), participants developed algorithmic thinking skills through flowchart activities and simple algorithm design tasks.

Eduweb
https://revistaeduweb.org/
ISSN: 1856-7576
Open Acces

In the subsequent phase (Weeks 7–10), the focus shifted to technology integrated applications. PSTs were introduced to the Arduino board and the mBlock programming environment, where they engaged in hands-on activities involving coding, circuit construction, and transferring block-based code to physical systems. During this phase, participants worked collaboratively to design and implement circuit based instructional materials.

The process concluded in Week 11 with the administration of retention tests and the submission of participant assignments. Throughout the intervention, an inquiry based and collaborative learning approach was adopted to promote active engagement, problem solving, and pedagogical reflection.

Table 1.
Study Procedure

Week	Topic	Content
1	Introduction of the study and administration of tests	Introduction of the study topics; administration of pre-tests: BEAT, AAT, BAAT, MBT, and PSEPS
2-3	Basic electronics	Theoretical knowledge, use of breadboard, introduction of circuit components, connection methods of circuit components
4-5	Algorithm logic and examples, flowchart applications	Explanation of algorithmic logic, simple algorithm examples
6	Test administration	Administration of BEAT, BAAT, and PSEPS as post-tests
7-8	Arduino Board and mBlock Training and Applications	Introduction of Arduino board; introduction of mBlock program, mBlock applications, transferring codes from mBlock to Arduino board
9-10	Arduino Applications via mBlock	mBlock applications; transferring codes from mBlock to Arduino board; administration of MBT and AAT post-tests
11	Completion of the study and submission of assignments	Administration of retention tests, collection of assignments

Data Analysis

Quantitative data obtained from the BEAT, AAT, BAAT, MBT, and PSEPS were analyzed using SPSS Statistics 22. Descriptive statistics were calculated for all measures, and normality assumptions were examined using the Shapiro-Wilk test alongside skewness and kurtosis values. As the distributions met normality criteria, paired-samples t tests were conducted to examine differences between pre and post test scores (BEAT, BAAT, PSEPS) and across pre, post, and retention measurements (MBT, AAT).

Qualitative data collected through semi-structured interviews were analyzed descriptively. Interview transcripts were coded, categorized, and interpreted to identify participants' views on the use of block-based coding and Arduino in science teaching.

To ensure the validity and reliability of the instruments, a systematic development and validation process was followed. For all achievement tests (BAAT, AAT, MBT, and BEAT), pilot studies were conducted with 60 university students who had prior experience in the subject areas. Based on item analyses, items with discrimination indices below .30 were removed, and the final instruments demonstrated acceptable item discrimination and difficulty levels.

Specifically, the BAAT showed item discrimination values ranging from .34 to .62, with an average difficulty index of .51 and a KR-20 reliability coefficient of .78. The AAT had discrimination indices between .39 and .57, an average difficulty index of

https://revistaeduweb.org/
ISSN: 1856-7576
Open Acces

.54, and a KR-20 value of .64. The MBT demonstrated discrimination indices ranging from .48 to .66, with a mean difficulty index of .51 and a KR-20 coefficient of .75. Similarly the BEAT showed discrimination values between .52 and .68, with an average difficulty index of .62 and a KR-20 coefficient of .71. For the PSEPS, originally adapted into Turkish by Altun & Mazman (2012), the overall Cronbach's alpha coefficient was reported as .928. Minor revisions were made to align the scale with the study context, and expert opinions were obtained to ensure content validity. Overall, the combination of expert review, pilot testing, item analysis, and internal consistency coefficients provided evidence for the validity and reliability of the instruments used in this study.

The statistical procedures were determined based on the distribution characteristics of the data and the research design. Although some variables showed statistically significant results in normality tests, skewness and kurtosis values remained within the acceptable range (−1 to +1), supporting the assumption of normality. Therefore, parametric tests were considered appropriate for the analysis.

Findings

This section presents the study's findings in alignment with the research questions. Quantitative and qualitative results are reported in a complementary fashion; interpretive explanations are reserved for the Discussion section.

Findings Related to Research Question 1: Self-Efficacy

The results of the PSEPS indicate a statistically significant increase in PST's perceptions of self-efficacy following the training. As shown in Table 2, the mean post test score was substantially higher than the mean pre test score, and the paired samples t test indicated that this difference was statistically significant ($p<.05$). Beyond statistical significance, this finding suggests a considerable improvement in PST' confidence in using block-based coding and related technologies. The magnitude of change indicates that the traning played a meaningful role in enhancing self-efficacy beliefs.

Table 2.
PSEPS Paired t-Test

	N	M	SD	df	t	p
Pre-test	37	2.79	0.90	36	-6.062	0.000
Post-test	37	4.81	0.84			

Findings Related to Research Question 2: Learning Outcomes

Quantitative findings demonstrated significant improvements across all achievement measures. BAAT and BEAT results showed statistically significant gains from pre-test to post-test (Tables 3 and 4). These results indicate that participants' foundational knowledge in algorithmic thinking and electronics improved following the instructional process.

Table 3.
BAAT Paired t-Test

	N	M	SD	df	t	p
Pre-test	37	3.24	2.30	36	-14.86	0.000
Post-test	37	10.94	2.46			

Eduweb
https://revistaeduweb.org/
ISSN: 1856-7576
Open Acces

Table 4.
BEAT Paired t-Test

	N	M	SD	df	t	p
Pre-test	37	5.16	2.04	36	-7.15	0.000
Post-test	37	8.16	1.74			

The AAT was administered to PST at three stages of this study: as a pre-test before the instructional intervention, as a post-test following the completion of the lessons, and as a retention test during the final week of the study.

Table 5.
AAT Descriptive Statistics

	N	M	SD	p
Pre-test	37	3.59	1.73	0.353
Post-test	37	7.62	1.36	
Retention	37	8.05	1.59	

The results of Mauchly's test, presented in Table 5, demonstrated that the sphericity assumption was met ($p > .05$). The descriptive results reveal a clear upward trend, with the mean scores increasing steadily from the pre to post test and then to the retention test, suggesting continuous progress in participants' achievement in Arduino.

For Arduino (AAT) and mBlock (MBT), repeated measures analyses revealed statistically significant differences among pre-test, post-test, and retention test scores (Tables 5-8). Mean scores increased steadily across measurement points, with retention test results exceeding post test scores. This pattern suggests that learning gains were not only immediate but also sustained over time, particularly in applied and practice-oriented domains such as Arduino and mBlock.

Table 6.
Results of ANOVA for the AAT

	SS	df	MS	F	p
Sphericity Assumed	447.586	2	223.793	103.105	0.000

The ANOVA results indicated in Table 6 that the pre-test, post-test, and retention test scores differed significantly from one another ($p<.05$). These findings imply that the instructional activities had a meaningful and lasting impact on PST's Arduino knowledge and skills. The consistent increases across all three measures demonstrate immediate learning gains and sustained retention, confirming that the training achieved its intended objectives.

Table 7.
MBT Descriptive Statistics

	N	M	SD	p
Pre-test	37	5.72	1.78	0.342
Post-test	37	7.21	1.37	
Retention	37	9.13	1.00	

Mauchly's test showed no violation of sphericity ($p > .05$) (Table 7). The descriptive statistics show a steady increase in mean scores across all the tests, indicating progressive improvement in participants' performance on mBlock.

https://revistaeduweb.org/
ISSN: 1856-7576
Open Acces

Table 8.
Results of ANOVA for the MBT

	SS	df	MS	F	p
Sphericity Assumed	215.694	2	107.847	65.267	0.000

As presented in Table 8, test scores differed significantly across the three measurement periods, $F(2, 72) = 65.267$, $p < .05$. This finding suggests that the instructional activities had a meaningful and sustained impact on participants' mBlock achievement. Steady gains in score demonstrate effective knowledge acquisition and skill retention resulting from the training.

Findings Related to Research Question 3: Pedagogicall Views

Qualitative findings obtained from semi-structured interviews were organized to reflect how PSTs interpreted and experienced the instructional process in relation to the quantitative outcomes. To reduce thematic overlap and enhance analytical clarity, related themes were grouped under three broader categories: (a) learning engagement and participation, (b) instructional design and material development, and (c) perceived challenges and instructional needs. Representative quotations selected to illustrate each category are presented in Table 9.

Learning Engagement and Participation

PSTs frequently reported that mBlock and Arduino applications increased students' interest, attention, and active participation in science lessons. Statements highlighted that hands-on interactions and collaborative activities made learning experiences more engaging and motivating.

Instructional Design and Material Development

A large majority of PSTs emphasized the pedagogical value of designing instructional materials using mBlock and Arduino. PSTs indicated that material development facilitated concretization of abstract concepts, like electrical circuits, and supported more permanent learning. Several responses also reflected participants' intention to integrate such materials into future science lessons.

Perceived Challenges and Instructional Needs

Although many PSTs expressed confidence in using block-based coding environments, some reported anticipated difficulties, particularly regarding Arduino, due to limited prior knowledge or resource constraints. PSTs expressed a strong demand for allocating a separate course hour to coding supported science activities, associating these activities with increased effectiveness and student motivation.

Importantly, these qualitative findings help explain the quantitative results by highlighting the mechanisms underlying the observed improvements. Increased engagement, active participation, and material design experiences appear to have contributed to both enhanced self-efficacy and improved learning outcomes.

Eduweb
https://revistaeduweb.org/
ISSN: 1856-7576
Open Acces

Table 9.
Themes, Categories, Codes, and Sample Excerpts from the Semi Structured Interviews

Theme	Category	Code	Sample Quotation
Increased Interest and Engagement	Students' attention and motivation	Attracting interest and participation	"Because students have something to work on, it makes the lesson more fun and engaging." (PST10). "The applications are attention-grabbing." (PST20) "They would increase students' interest in science lessons." (PST21)
Necessity of Material Development	Teaching with self-designed materials	Enhancing retention through materials	"Developing and teaching with materials would attract students' interest more and lead to more permanent learning." (PST23) "Using mBlock and Arduino makes learning more lasting because students learn by doing." (PST2) "Material development helps students visualize what they learn and understand better." (PST14)
Challenges in Implementation	Knowledge and resource limitations	Lack of technical knowledge	"I think I wouldn't struggle because I received the basics of Arduino, but with more training, I could produce better content." (PST6) "Limited resources might make it difficult to ensure every student's participation." (PST17)
Perceived Necessity in Science Lessons	Improving learning outcomes	Concretization of abstract concepts	"Yes, it is necessary because it provides concrete learning and makes students love the subject more." (PST1) "It helps students visualize concepts and understand them more easily, turning theory into practice." (PST29)
Applications and Inventions	Purpose of design and innovation	Educational and practical uses	"Using a distance sensor to measure heights or distances in construction." (PST7) "Designing a lightning simulation to make science lessons more engaging." (PST32)
Collaboration and Motivation	Peer interaction and teamwork	Group learning and idea exchange	"Group work helps us complete our deficiencies and discuss ideas together to create a project." (PST5) "Shared decision-making and the reciprocal flow of ideas made the process much more productive." (PST12) "Working in groups made me more curious and motivated; we learned a lot from each other." (PST12)
Preferred Topics for Teaching	Subject preferences	Earth and the Universe as priority	"I would prefer to teach topics related to Earth and the Universe first." (PST9)
Social Reactions and Dissemination	Sharing experiences with peers	Positive community feedback	"I talked to my friends; they were curious, and I explained it to them in detail." (PST6) "They were very surprised, found it cool, and some even decided to take a course." (PST17)
Demand for a Separate Course Hour	Curriculum integration	Increasing interest and motivation	"It would attract attention by doing something different and make abstract concepts more concrete and memorable." (PST18) "Allocating a separate course hour for mBlock and Arduino would make lessons more effective and motivate students to participate actively." (PST24) "Such a course would help students grasp abstract science concepts more easily and develop their creativity." (PST36).

https://revistaeduweb.org/
ISSN: 1856-7576
Open Acces

Discussion

The present study employed an explanatory sequential mixed methods design to investigate changes in PSTs' programming self-efficacy perceptions and learning outcomes, and to explain them through their experiences. Accordingly, this section integrates quantitative and qualitative findings and situates them within the relevant literature to clarify how and why block-based coding and Arduino-supported activities influenced both cognitive and affective outcomes.

Integration of Quantitative and Qualitative Findings

The quantitative results demonstrated statistically significant improvements in both perceptions of programming self-efficacy and in achievement across all measured domains, including algorithmic thinking, basic electronics, Arduino, and mBlock. The qualitative findings provide explanatory insight into these results by highlighting the learning experiences that PSTs associated with their development. PSTs frequently emphasized active participation, hands-on experimentation, collaborative work, and material design as central elements of the instructional process. Similar patterns have been reported in previous studies, indicating that block-based coding environments foster engagement and confidence by allowing learners to interact directly with technological systems (Arslan & Tanel, 2021; Bower at al. 2017; Sanusi et al., 2025; Çoban et al., 2020).

From a theoretical perspective, the observed increase in programming self-efficacy can be interpreted through Bandura's self-efficacy framework. PSTs repeated successes in completing coding tasks, constructing circuits, and observing immediate outcomes represent mastery experiences, which are considered the most influential source of self-efficacy development. In addition, peer collaboration and group-based problem solving, frequently mentioned in the qualitative findings, align with vicarious experiences and social persuasion, further reinforcing confidence beliefs. Similar mechanisms have been reported in studies examining the relationship between coding activities, computational thinking, and self-efficacy among preservice teachers (Ateşkan & Hart, 2021; Ortega-Ruipérez & Lázaro Alcalde, 2023).

Learning Outcomes, Retention, and Instructional Design

The significant gains observed in achievement tests, particularly the sustained improvement in Arduino and mBlock retention scores, suggest that learning extended beyond short term performance. This finding is consistent with previous research showing that block-based coding and robotics applications, when combined with hands-on practices, enhance academic achievement and support durable learning (Coşkunserçe, 2023; Ga & Chang, 2025). Qualitative findings help explain this durability by revealing that participants perceived material design and experimentation as key contributors to meaningful learning.

Designing instructional materials using mBlock and Arduino required PSTs to integrate theoretical knowledge with practical implementation, which aligns with constructivist and experiential learning principles. Similar conclusions have been drawn in studies emphasizing that coding supported material development facilitates the concretization of abstract scientific concepts and promotes learning by doing (Ga & Chang, 2025). In this sense, the present study extends earlier findings by demonstrating that material design functions as a mediating process between coding activities and sustained learning outcomes in science teacher education.

https://revistaeduweb.org/
ISSN: 1856-7576
Open Acces

Programming Self-Efficacy and Pedagogical Readiness

The increased programming self-efficacy observed in this study is consistent with previous research indicating that block-based coding reduces anxiety and cognitive load while supporting the development of confidence among novice learners (Köksaloğlu, 2022; Unal & Topu, 2021; Şentürk & Sari, 2025). PSTs' qualitative responses further suggest that feeling competent in using coding tools enhanced their readiness to integrate these technologies into future science teaching. This finding aligns with studies reporting that preservice teachers with higher self-efficacy are more likely to adopt technology integrated instructional practices (Bower et al., 2017; Hur, 2021).

Moreover, PSTs' emphasis on active engagement, curiosity, and motivation supports the argument that affective gains play a crucial role in technology adoption. Previous studies have similarly highlighted that positive attitudes and interest are essential precursors to sustained engagement with coding and robotics in educational contexts (Şentürk & Sari, 2025).

Challenges, Cognitive Load, and Structural Constraints

Despite the overall positive outcomes, some PSTs anticipated difficulties in implementing Arduino based activities, particularly due to limited prior knowledge and resource constraints. These concerns are consistent with earlier studies reporting technical challenges, infrastructure limitations, and insufficient instructional time as barriers to effective coding integration (Bower et al. 2017; Rao & Bhagat, 2024; Ortega-Ruipérez & Lázaro Alcalde, 2023). Such challenges may increase cognitive load and reduce teachers' willingness to experiment with technology in real classroom settings.

In this context, PSTs' strong demand for a separate course hour dedicated to mBlock and Arduino activities should be interpreted as a structural need rather than a personal preference. Similar recommendations have been made in prior research emphasizing that sufficient instructional time and institutional support are critical for developing coding competencies and pedagogical confidence (Bower et al., 2017; Govender et al., 2025). Addressing these constraints may enhance the scalability and sustainability of coding supported science instruction.

Positioning the Study within the Literature

While many previous studies have reported positive effects of block-based coding and Arduino on attitudes, achievement, or computational thinking, they often focus on a single outcome or rely on descriptive analyses. The present study contributes to the literature by integrating programming self-efficacy, achievement, and retention within a single explanatory sequential mixed methods framework. In addition, by situating coding activities within a specific science content domain, electrical circuits, the study responds to calls for more context specific and pedagogically grounded research in science teacher education (Govender et al., 2025; Vasconcelos & Kim, 2022).

Conclusions

This study demonstrates that integrating block-based coding and Arduino into PST education can lead to significant improvements in programming self-efficacy and learning outcomes. The explanatory sequential mixed methods design revealed not

https://revistaeduweb.org/
ISSN: 1856-7576
Open Acces

only measurable gains in achievement and retention, but also the experiential factors such as active participation, material design, and collaboration that underpin these gains. These findings contribute to the literature by showing that technology-integrated instructional processes are effective not only in improving cognitive outcomes but also in strengthening affective dimensions such as self-efficacy, which plays a critical role in teachers' willingness to integrate technology into their future practices.

By positioning PSTs as instructional designers rather than passive technology users, the study highlights the pedagogical value of coding-supported material development in science education. From a pedagogical perspective, the study highlights the importance of positioning digital technologies as tools for instructional design rather than as ends in themselves. At the same time, identified challenges related to technical complexity and instructional time underscore the need for structured support within teacher education programs.

Overall, the findings suggest that block-based coding and Arduino, when systematically integrated into science teacher training, can strengthen both cognitive and affective dimensions of learning and contribute to the development of digitally competent and pedagogically confident science teachers.

Limitations and Recommendations for Future Studies

This study was limited to a small sample of participants from a single institution. Future studies necessitate larger and more diverse samples to improve generalizability.

Only block-based programming was used. Comparative studies of block-based and text-based programming (mBlock vs. Arduino interface) are recommended.

The research focused on a single content area. Expanding to multiple domains of science (Earth and the Universe, Physical Phenomena, Matter and Change, Living Things and Life) could provide broader insights.

Because Arduino materials are costly, replication studies using simulation platforms such as Proteus are recommended.

Further research may examine whether factors such as gender, age, or type of high school influence performance and perceptions in coding activities.

The findings have implications for teacher training. Future studies could develop in-service professional development programs to support the integration of mBlock and Arduino into science education.

Statements and Declarations

This study was approved by the Ethics Committee of University Ondokuz Mayis, Approval No: 2019-329, Date: 25.10.2019, and all procedures were carried out in accordance with institutional and international ethical standards. The authors declare that no financial support was received for the conduct or publication of this research. The study was conducted independently, without any external funding or sponsorship.

This study was conducted as part of the first author's master's thesis research, supervised by the second author.

https://revistaeduweb.org/
ISSN: 1856-7576
Open Acces

Bibliographic references

Altun, A., & Mazman, S. G. (2012). The reliability and validity study of the Turkish form of the Self-efficacy Perception Scale for Programming. *Journal of Measurement and Evaluation in Education and Psychology-EPOD*, *3*(2), 297-308. https://acortar.link/AI40km

Arslan, K., & Tanel, Z. (2021). Analyzing the effects of Arduino applications on students' opinions, attitude and self-efficacy in programming class. *Education and Information Technologies*, *26*(1), 1143-1163. https://doi.org/10.1007/s10639-020-10290-5

Ateşkan, A., & Hart, D. O. (2021). Demystifying computational thinking for teacher candidates: A case study on Turkish secondary school pre-service teachers. *Education and Information Technologies*, *26*(5), 6383-6399. https://doi.org/10.1007/s10639-021-10626-9

Bower, M., Wood, L. N., Lai, J. W., Howe, C., Lister, R., Mason, R., Highfield, K., & ve Veal, J. (2017). Improving the computational thinking pedagogical capabilities of school teachers. *Australian Journal of Teacher Education, 42*(3), 53-72. https://doi.org/10.14221/ajte.2017v42n3.4

Bulus Kirikkaya, E, & Basaran, B. (2019). Investigation of the effect of the integration of arduino to electrical experiments on students' attitudes towards technology and ICT by the mixed method. *European Journal of Educational Research, 8*(1), 31-48. https://izlik.org/JA73XL45GA

Çoban, E., Korkmaz, Ö., Çakır, R., & Uğur Erdoğmuş, F. (2020). Attitudes of IT teacher candidates towards computer programming and their self-efficacy and opinions regarding to block-based programming. *Education and Information Technologies*, *25*(5), 4097-4114. https://doi.org/10.1007/s10639-020-10164-w

Coşkunserçe, O. (2023). Comparing the use of block-based and robot programming in introductory programming education: Effects on perceptions of programming self-efficacy. *Computer Applications in Engineering Education*, *31*(5), 1234-1255. https://doi.org/10.1002/cae.22637

Creswell, J. W. (2006). *Qualitative Inquiry and Research Design: Choosing Among Five Approaches*. Londra: Sage Publications.

Creswell, J. W., & Plano-Clark, V. L. (2007). *Designing and conducting mixed methods research*. California: SAGE Publications.

Dat, L. C., Pham, H. T., & Nguyen, N. T. (2025). Effects of a block-based Arduino robotics course on computational thinking skills and STEM career interests of Vietnamese students. *Eurasia Journal of Mathematics, Science and Technology Education*, *21*(6), em2642. https://doi.org/10.29333/ejmste/16414

Doğanay, K. (2018). *The effect of science festivals upon with problem based stem activities on the student's science attitudes and academic achievements*. (Unpublished master's thesis). Kastamonu University.

Fjukstad, B., Angelvik, N., Hauglann, M. W., Knutsen, J. S., Grønnesby, M., Gunhildrud, H., & Bongo, L. A. (2018). Low-cost programmable air quality sensor kits in science education. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 227-232). https://doi.org/10.1145/3159450.3159569

Ga, S. H., & Chang, C. Y. (2025). Developing affordable and research-grade measurement devices with arduino for school science: a guide for non-coders. *Journal of Chemisrty Education, 102*, 404-409. https://doi.org/10.1021/acs.jchemed.4c00998

Govender, I., Govender, R. G., & Govender, D. W. (2025). Robotics and coding for teacher education: A constructionist approach. *Educational Process: International Journal*, *15*, e2025176. https://doi.org/10.22521/edupij.2025.15.176

Eduweb
https://revistaeduweb.org/
ISSN: 1856-7576
Open Acces

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education, 126*, 296-310. https://doi.org/10.1016/j.compedu.2018.07.004

Hur, J. W. (2021). Pre-service teachers' beliefs, confidence, and interest in computer science education. In C. Mouza, A. Yadav, & A. Ottenbreit-Leftwich (Eds.), *Preparing pre-service teachers to teach computer science: Models, practices, and policies* (pp. 29–49). Information Age Publishing.

Jeon, M., & Kwon, K. (2024). Parallel instruction of text-based and block-based programming: On novice programmers' computational thinking practices. *TechTrends*, *68*(6), 1033-1050. https://doi.org/10.1007/s11528-024-00993-8

Köksaloğlu, C. (2022). *Block-based coding in K-12 education: A systematic literature review* (Unpublished Master's thesis). Bilkent University, Turkey.

Kozcu Cakir, N., & Guven, G. (2026). Enhancing engineering design, scientific creativity, and decision-making skills in prospective science teachers through engineering design-based robotics coding applications. *Research in Science & Technological Education*, *44*(1), 157–182. https://doi.org/10.1080/02635143.2025.2456778

Mashishi, T. C., & Ramaila, S. (2024). Preservice teachers' perceptions, attitudes, and challenges of using scratch as a coding tool to foster active learning in life sciences classrooms. *International Journal of Learning, Teaching and Educational Research*, *23*(9), 472-497. https://doi.org/10.26803/ijlter.23.9.24

Muşlu Kaygısız, G, Üzümcü, Ö., & Ucar, F. M. (2020). The case of prospective teachers' integration of coding-robotics practices into science teaching with STEM approach. *Elementary Education Online*, *19*(3), 1200-1213. Doi: 10.17051/ilkonline.2020.728020

Nannim, F. A., Ibezim, N. E., Oguguo, B. C., & Nwangwu, E. C. (2024). Effect of project-based Arduino robot application on trainee teachers computational thinking in robotics programming course. *Education and Information Technologies, 29*(10), 13155-13170. https://doi.org/10.1007/s10639-023-12380-6

Ortega-Ruipérez, B., & Lázaro Alcalde, M. (2023). Teachers' perception about the difficulty and use of programming and robotics in the classroom. *Interactive Learning Environments*, *31*(10), 7074-7085. https://doi.org/10.1080/10494820.2022.2061007

Rachmatullah, A., & Wiebe, E. N. (2023). Changes and sources of changes of middle school teachers' self-efficacy for teaching science in a computationally rich environment: A mixed-methods study. *Journal of Science Teacher Education*, *34*(2), 132-156. https://doi.org/10.1080/1046560X.2022.2035990

Rao, T. S. S., & Bhagat, K. K. (2024). Computational thinking for the digital age: a systematic review of tools, pedagogical strategies, and assessment practices. *Educational Technology Research and Development*, *72*(4), 1893-1924. https://doi.org/10.1007/s11423-024-10364-y

Rodrigues, R. N., Costa, C., Abbasi, M., Martins, F., & Brito-Costa, S. (2025). Self-efficacy in computational thinking: Preliminary analysis of pre-service teachers' perception through a Portuguese tool. *Eurasia Journal of Mathematics, Science and Technology Education,* *21*(6), em2647. https://doi.org/10.29333/ejmste/16507

Sanusi, I. T., Cudjoe, E. S., Ayanwale, M. A., & Adepoju, B. (2025). Pre-service teachers' perception of programming education. *SAGE Open*, *15*(1), 21582440251327019. https://doi.org/10.1177/21582440251327019

Sarı, U., Pektaş, H. M., Şen, Ö. F., & Çelik, H. (2022). Algorithmic thinking development through physical computing activities with Arduino in STEM

https://revistaeduweb.org/
ISSN: 1856-7576
Open Acces

education. *Education and Information Technologies*, *27*(5), 6669-6689. https://doi.org/10.1007/s10639-022-10893-0
Şentürk, M. L., & Sari, U. (2025). Pre-service science teachers' computational thinking skills and STEM self-efficacy. *International Journal of Learning Technology, 20*(4), 471-506. https://doi.org/10.1504/IJLT.2025.151035
Sun, L., & Zhou, D. (2024). K-12 teachers' programming attitudes among different disciplines: Analysis of influential factors. *Journal of Computer Assisted Learning*, *40*(2), 538-556. https://doi.org/10.1111/jcal.12895
Tsai, F. H. (2023). Using a physical computing project to prepare preservice primary teachers for teaching programing. *Sage Open*, *13*(4), https://doi.org/10.1177/21582440231205409
Türkoğuz, S., & Sefer, F. (2019). Investigation of the effects on pre-service teachers' self-efficacy toward information technologies of analytical chemistry experiences supported by arduino. *Journal of Theoretical Educational Sciences, 12(*4), 1164-1192. https://doi.org/10.30831/akukeg.409775
Unal, A., & Topu, F. B. (2021). Effects of teaching a computer programming language via hybrid interface on anxiety, cognitive load level and achievement of high school students. *Education and Information Technologies*, *26*(5), 5291-5309. https://doi.org/10.1007/s10639-021-10536-w
Vasconcelos, L., & Kim, C. (2020). Preparing preservice teachers to use block-based coding in scientific modeling lessons. *Instructional science, 48*(6), 765-797. https://doi.org/10.1007/s11251-020-09527-0
Vasconcelos, L., & Kim, C. (2022). Preservice science teachers coding science simulations: Epistemological understanding, coding skills, and lesson design. *Educational Technology Research and Development*, *70*(4), 1517-1549. https://doi.org/10.1007/s11423-022-10119-7
Wawire, B. A., Nkunika, F., Robinette, J., Manyau, M., Koo, J. B., & Barnes-Story, A. E. (2025). Teacher educator knowledge, skills, and self-efficacy: systemic impacts on initial teacher education programming. *Trends in Higher Education*, *4*(3), 43. https://doi.org/10.3390/higheredu4030043
Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education (TOCE)*, *18*(1), 1-25. https://doi.org/10.1145/3089799
Woo, K., & Falloon, G. (2025). Integrating coding across the curriculum: A scoping review. *Computer Science Education*, *35*(2), 216-237. https://doi.org/10.1080/08993408.2024.2344402
Xu, E., Wang, W., & Wang, Q. (2023). A meta-analysis of the effectiveness of programming teaching in promoting K-12 students' computational thinking. *Education and Information Technologies*, *28*(6), 6619-6644. https://doi.org/10.1007/s10639-022-11445-2
Yuana, R. A., Sajidan, S., Wiranto, W., & Nizam, M. (2025). Strategies for integrating computational thinking and scientific approaches in programming education: a systematic literature review. *Discover Education,* *4*(1), 371. https://doi.org/10.1007/s44217-025-00834-7
Yucelyigit, S. (2023). Coding practices in early childhood classrooms: Aphenomenological study with ECE teachers in Turkey. *Issues in Educational Research,* *33*(3), 1250-1270. https://search.informit.org/doi/epdf/10.3316/informit.T2024050800009891006785635