ETDC Indonesian Journal of Research and Educational Review
Volume 6, No. 1, 2026, pp 283 - 293
e–ISSN 2809-3682
p–ISSN 2809-7505
https://doi.org/10.51574/ijrer.v6i1.5385

Inquiry-Based Microsoft MakeCode Integration for Enhancing Technical Competencies in Vocational Basic Electronics Engineering

Ibnu Qayyum Abdullah[1], Ummiati Rahmah[2], Saharuddin[3]
[1, 2, 3] Department of Mechatronics Vocational Education, Universitas Negeri Makassar, Indonesia

Article Info

Article history:

Received June 25, 2026
Accepted August 30, 2026
Published September 05, 2026

Keywords:

Block-Based Programming; Interactive Simulation; Inquiry-Based Learning; Microsoft MakeCode; Technical and Vocational Education (TVET).

ABSTRACT

Modern vocational education demands an effective synthesis between conceptual understanding and practical technical execution. However, traditional lecture-centric pedagogy in Basic Electronics Engineering often results in low performance due to abstract circuit logic. This quasi-experimental study investigated the effectiveness of integrating Microsoft MakeCode block-based simulation within an inquiry-based framework. Conducted over six weeks, the study involved two tenth-grade classes at Vocational School 10 Makassar: an experimental group ($n = 24$) utilizing inquiry-based MakeCode simulation and a control group ($n = 21$) receiving direct instruction. Data were collected via expert-validated 20-item achievement tests (Cronbach's $\alpha = 0.829$) and observation sheets. Post-intervention results demonstrated significant gains: the experimental group achieved a mean post-test score of 82.29 ± 6.64 (pre-test: 46.88 ± 9.08), while the control group reached 67.86 ± 9.24 (pre-test: 44.52 ± 8.65). An Independent Samples t-test revealed a statistically significant difference favoring the experimental cohort ($t(43) = 6.002$, $p < 0.001$, Cohen's $d = 1.79$), supported by paired-samples analysis ($t(23) = -19.572$, $p < 0.001$). The experimental group attained a mean normalized gain of 0.66 (66.76%) and an 89.4% student engagement rate. Integrating visual block-programming simulation within inquiry cycles bridges abstract schematic theory and hands-on system logic, offering a scalable instructional model for Industry 4.0 Technical and Vocational Education and Training.

Corresponding Author:

Ibnu Qayyum Abdullah,
Department of Mechatronics Vocational Education, Universitas Negeri Makassar, Indonesia
Email: ibnuqayyumabdullah18@gmail.com

1. INTRODUCTION

Technical and Vocational Education and Training (TVET) plays a foundational role in building a skilled industrial workforce by bridging theoretical concepts and practical domain competencies (Kebede et al., 2024; Ye et al., 2024). In electronics engineering education, secondary vocational curricula must align rigorously with industrial standards to prepare graduates for technician roles in automation, hardware maintenance, and embedded systems (Power & Buckley, 2025). However, instructional execution in foundational courses—specifically Basic Electronics Engineering—continues to face significant pedagogical bottlenecks (Rivera & Baptista, 2024). Traditional teaching modalities rely predominantly on passive text-based lecturing, static schematic diagrams, and verification-type laboratory exercises. Consequently, students frequently struggle to internalize abstract dynamic concepts such as signal flow, logic gate behaviors, input-process-output (IPO) architectures, and transient circuit states (Scherer et al., 2020).

Empirical preliminary observations at the study site (Vocational School 10 Makassar) revealed that over 62% of tenth-grade electronics students failed to achieve the minimum mastery threshold (*KKM = 75*) in initial formative assessments on basic digital logic and switching circuits. Instructional interviews indicated that students experienced substantial cognitive overload when attempting to translate theoretical Boolean logic and micro-controller routines into functional hardware assemblies without intermediate visual simulation tools. Modern computing education emphasizes physical computing and dynamic simulation platforms as essential scaffolds to reduce cognitive load and connect theoretical concepts with interactive physical behavior (Hodges et al., 2025; Kastner-Hauler et al., 2022; Tedre et al., 2018).

Among emerging educational software platforms, Microsoft MakeCode has gained widespread recognition as an intuitive, web-based visual programming environment (Ball et al., 2019; Devine et al., 2019). MakeCode integrates block-based drag-and-drop mechanics alongside text-based syntax (JavaScript/TypeScript and Python), enabling novice learners to visualize execution flow and test circuit dynamics in real time via embedded virtual simulators (Devine et al., 2018). Existing literature underscores the utility of MakeCode across diverse contexts, including game-based computing (Voštinár et al., 2024), low-resource classroom implementation (Xu & Liu, 2023), and primary STEAM education (Mustan & Harizanov, 2024; Rafti et al., 2025). However, most existing studies focus primarily on general coding literacy, game mechanics, or primary school physical computing, leaving a clear research gap regarding how block-based simulation interacts with formal engineering logic in vocational technical courses.

Simultaneously, educational technology tools yield suboptimal outcomes when decoupled from structured pedagogical frameworks. Inquiry-Based Learning (IBL) places students at the center of knowledge construction through systematic phases of orientation, hypothesis generation, experimentation, data evaluation, and reflective conclusion (Chen & Chen, 2025; Hinostroza et al., 2024; Ozdem-Yilmaz & Bilican, 2025). While inquiry models foster higher-order analytical thinking, executing open-ended inquiry in hardware-intensive vocational courses often incurs high equipment risks, hardware resource shortages, and delayed feedback loops (Akuma & Callaghan, 2019; Chandrasekaran, 2024).

To address these challenges, this study proposes a conceptual and pedagogical integration: coupling Microsoft MakeCode's visual simulation platform with the structured phases of inquiry-based learning. From a psycho-pedagogical standpoint, MakeCode acts as an interactive cognitive bridge during the hypothesis-testing phase of inquiry (Yu, 2026). The drag-and-drop block interface reduces extraneous cognitive load associated with syntax errors, allowing vocational students to direct germane cognitive resources toward analyzing electronic logic, testing edge cases, and evaluating real-time simulator feedback (DeLozier & Shey, 2023). This integration directly addresses the demands of Industry 4.0 TVET frameworks by transforming abstract technical concepts into tangible, manipulative models.

Therefore, the primary objective of this study is to evaluate the effectiveness of an inquiry-based Microsoft MakeCode instructional intervention in improving cognitive learning outcomes in basic electronics engineering among vocational high school students. Specifically, this study tests the hypothesis that students instructed via inquiry-based MakeCode simulation demonstrate significantly higher post-intervention conceptual and practical achievement compared to those taught via conventional direct instruction.

2. METHOD

This study employed a quantitative quasi-experimental design utilizing the Quasi-Experimental Nonequivalent Control Group Design framework (Warne, 2020). This design was selected because intact classroom groups were preserved to maintain authentic institutional scheduling and prevent administrative disruption at the vocational school. The research was conducted at Vocational School 10 Makassar over an 8-week timeframe, consisting of 2 weeks for baseline assessment and instrument validation, 4 weeks (16 instructional hours) of intervention exposure, and 2 weeks for post-testing and psychometric data consolidation.

The target population comprised all Tenth-Grade students enrolled in the Electronics Engineering Department at Vocational School 10 Makassar (N = 45). Intact classroom sampling was used to assign Class X Teli 1 as the Experimental Group (n = 24; 18 males, 6 females; mean age = 15.8 ± 0.6 years) and Class X Teli 2 as the Control Group (n = 21; 16 males, 5 females; mean age = 15.7 ± 0.7 years). Both cohorts shared equivalent baseline academic entry prerequisites based on prior semester engineering foundation transcripts.

Both groups were taught by the same certified vocational engineering instructor to ensure instructional consistency and minimize teacher bias. Both groups covered identical curriculum modules in Basic Electronics Engineering (digital logic concepts, logic gates, switching circuits, and sensor-actuator input-output logic) over 4 weekly 4-hour sessions (16 total contact hours).

Table 1. Inquiry Phase (Experimental and Control Group)

Inquiry Phase	Experimental Group (MakeCode + Inquiry)	Control Group (Direct Instruction)
Orientation	Instructor presents real-world electronic problems via interactive MakeCode simulations.	Instructor lectures on electronic concepts using static whiteboards and slides.
Conceptualization	Students formulate hypotheses on circuit logic, signal states, and logic gate outputs.	Instructor presents theoretical formulas, truth tables, and solved examples.

Inquiry Phase	Experimental Group (MakeCode + Inquiry)	Control Group (Direct Instruction)
Investigation	Students build, simulate, and test block-programming logic models in MakeCode.	Students copy circuit diagrams and complete numerical textbook calculations.
Conclusion	Students analyze real-time simulator signals, verify hypotheses, and debug code errors.	Instructor summarizes circuit formulas and reviews answers on the board.
Application	Students extend MakeCode block models to complex multi-sensor input-output tasks.	Students work on individual static pencil-and-paper worksheet exercises.

Data were collected using two main instruments: a 20-item Cognitive Learning Achievement Test (administered as Pre-test and Post-test) and a Structured Observation Sheet for Student Learning Activity. Instrument Validation & Psychometrics: To address content and construct validity, the achievement test was subjected to rigorous Expert Judgment by two senior university lecturers in mechatronics and vocational pedagogy. Evaluated across construction, content alignment, and language clarity using a 4-point scale, Validator 1 assigned a total score of 33 (mean = 3.66, 'Sangat Valid'), and Validator 2 assigned 34 (mean = 3.77, 'Sangat Valid'). Following expert validation, empirical psychometric testing was conducted on an independent trial cohort of 20 students outside the experimental sample. Item discrimination indices ranged from 0.42 to 0.78, and item difficulty indices were balanced between 0.35 and 0.72. Instrument reliability was established using Cronbach's Alpha coefficient, yielding $\alpha = 0.829$ (20 items), exceeding the standard benchmark of 0.70 for high internal consistency.

Quantitative data were analyzed using IBM SPSS Statistics (v26.0). Descriptive statistics (means, standard deviations, standard errors) were computed for all testing conditions. Prerequisites for parametric testing were evaluated using the Shapiro-Wilk test for normality (recommended for $n < 50$) and Levene's Test for Equality of Variances. Between-group differences were evaluated using the Independent Samples t-Test, accompanied by Cohen's d effect size calculations (Cohen, 1988). Within-group gains were assessed using the Paired Samples t-Test and Normalized Gain (N-Gain) analysis according to Hake's (1998) formulation:

$$\textbf{\textit{N-Gain}} = \frac{(\textit{Post-test Score} - \textit{Pre-test Score})}{(\textit{Maximum Score} - \textit{Pre-test Score})}$$

3. RESULTS AND DISCUSSION

Results

Baseline Equivalence and Prerequisite Testing

Prior to formal hypothesis testing, the distributional assumptions of normality and homoscedasticity were rigorously evaluated to ensure the statistical validity of the subsequent parametric analyses. As detailed in Table 2, the Shapiro-Wilk test was conducted across both experimental and control conditions for pre-test and post-test measures. The results demonstrated that all score distributions conformed to a normal distribution, as none of the statistical deviations reached significance ($p > .05$). This distributional symmetry justifies the application of inferential parametric techniques without requiring data transformation.

Additionally, variance homogeneity between the Experimental and Control groups was assessed at both baseline and post-intervention stages using Levene's test for equality of variances. The analysis revealed no significant variance disparity between the groups at baseline ($F = 3.405$, $p = .072$) or following the intervention ($F = 3.656$, $p = .063$). Because both p-values exceeded the threshold for statistical significance, the assumption of homoscedasticity was fully met, thereby confirming that any post-intervention group differences can be attributed to the experimental effect rather than baseline variance heterogeneity.

Table 2. Distributional Normality and Variance Homogeneity

Group / Assessment	Sample (n)	Shapiro-Wilk W	Sig. (p)	Normality Result
Pre-test Experimental	24	0.948	0.265	Normally Distributed
Post-test Experimental	24	0.932	0.115	Normally Distributed
Pre-test Control	21	0.933	0.120	Normally Distributed
Post-test Control	21	0.971	0.591	Normally Distributed

Cognitive Learning Achievement and Hypothesis Testing

Prior to assessing the efficacy of the intervention, baseline equivalence was evaluated through a descriptive statistical analysis of the pre-test scores. At the pre-intervention stage, the Experimental Group ($M = 46.88$, $SD = 9.08$) and the Control Group ($M = 44.52$, $SD = 8.65$) demonstrated highly comparable performance profiles, establishing a balanced foundation for subsequent comparative evaluations. Following the four-week instructional intervention, a marked divergence in achievement levels was observed. The Experimental Group exhibited substantial improvement, attaining a post-test mean score of 82.29 ($SD = 6.64$), whereas the Control Group demonstrated a comparatively moderate increase, reaching a post-test mean score of 67.86 ($SD = 9.24$).

The pronounced mean difference in post-test outcomes highlights a primary descriptive disparity in academic gain between the two instructional conditions. To rigorously determine whether these observed improvements represent statistically meaningful and substantively significant effects, formal parametric evaluation and standardized effect size estimations were conducted. A consolidated comparative summary detailing the inferential hypothesis testing, mean difference trajectories, and normalized gain metrics across both cohorts is comprehensively presented in Table 3.

Table 3. Hypothesis Testing

Group	n	Pre-test Mean (SD)	Post-test Mean (SD)	Mean Diff.	t-statistic	df	Sig. (2-tailed)	Cohen's d
Experimental (MakeCode+ Inquiry)	24	46.88 (9.08)	82.29 (6.64)	+35.41	19.572*	23	< 0.001	1.79

Group	n	Pre-test Mean (SD)	Post-test Mean (SD)	Mean Diff.	t-statistic	df	Sig. (2-tailed)	Cohen's d
Control (Conventional)	21	44.52 (8.65)	67.86 (9.24)	+23.34	11.214*	20	< 0.001	--

*Note: * Paired-samples t-test within-group contrast. Independent Samples t-test between post-test groups: t (43) = 6.002, p < 0.001, Cohen's d = 1.79 (Large Effect Size)*

To evaluate the relative efficacy of the intervention, an Independent Samples t-test was conducted on the post-test scores between the two cohorts. The analysis revealed a statistically significant difference in technical achievement favoring the Experimental Group ($t(43) = 6.002,\ p < .001$), indicating that the observed disparity in mean performance was unlikely to be an artifact of random sampling variation. This statistically robust result confirms that students exposed to the specialized pedagogical framework significantly outperformed their peers in the conventional control condition at the conclusion of the instructional period.

To quantify the practical magnitude of this instructional outcome, standardized effect size metrics were calculated alongside inferential testing. The resulting effect size ($Cohen's\ d = 1.79$) substantially surpassed the conventional threshold for a large effect ($d > 0.80$), reflecting a highly pronounced educational impact. This substantial effect size demonstrates that the inquiry-based MakeCode treatment exerted a transformative influence on student technical proficiency, validating its pedagogical utility as an effective instructional methodology for domain-specific mastery.

Normalized Gain (N-Gain) and Learning Activity Observation

To rigorously evaluate the relative learning trajectories and implementation fidelity of the pedagogical intervention, Normalized Gain (N-Gain) analyses and structured observational metrics were integrated into the evaluation framework. The N-Gain analysis, calculated to measure individual knowledge acquisition relative to maximum possible growth, revealed a mean score of 0.66 (66.76%) for the Experimental Group. According to Hake's established criteria for conceptual gain, this performance places the experimental intervention firmly within the moderate-to-high effectiveness threshold. Conversely, the Control Group attained a mean N-Gain of 0.42 (42.15%), underscoring a distinct disparity in learning progression and demonstrating that the targeted pedagogical approach yielded superior instructional efficiency compared to conventional methods.

Table 4. N-Gain Results

Group	Mean N-Gain Score	Mean N-Gain (%)	Hake's Effectiveness Category	Instructional Method
Experimental Group	0.66	66.76%	Moderate-to-High Threshold	Targeted Pedagogical Approach
Control Group	0.42	42.15%	Moderate	Conventional Methods

Complementing the cognitive outcomes, observational evaluations were conducted to systematically monitor procedural adherence and student engagement throughout the four-week instructional sequence. A panel of four certified observers assessed classroom dynamics and pedagogical execution using a standardized observation protocol. The composite implementation scores across all observed sessions ranged consistently between 85.7% and 96.4%, culminating in an overall mean observation score of 89.4%. In accordance with qualitative evaluation benchmarks, this level of engagement falls into the "Highly Effective" (Sangat Efektif) classification, confirming that classroom activities were executed in close alignment with the designed pedagogical model.

Taken together, these empirical findings confirm high fidelity in the implementation of the inquiry-based MakeCode instructional syntax. The strong alignment between observed teacher-student dynamics and prescribed activity protocols indicates that the cognitive improvements recorded in the experimental cohort were directly driven by the structured inquiry mechanisms rather than confounding instructional variables. Consequently, the combination of robust normalized learning gains and verified execution fidelity solidifies the empirical validity of the MakeCode treatment as an effective paradigm for fostering technical proficiency.

Discussion

The empirical findings of this study provide robust evidence that integrating Microsoft MakeCode within an Inquiry-Based Learning (IBL) framework significantly enhances vocational high school students' cognitive learning outcomes in Basic Electronics Engineering relative to conventional direct instruction. The marked superiority in post-test mean performance (82.29 vs. 67.86) alongside an exceptionally large effect size ($Cohen's\ d = 1.79$) underscores the profound pedagogical value of visual simulation environments in technical and vocational education. These metrics demonstrate that bridging abstract engineering concepts with interactive, block-based visual platforms transforms how technical competencies are acquired, facilitating accelerated cognitive growth and deeper conceptual mastery within vocational cohorts.

From a psycho-pedagogical perspective, these empirical outcomes strongly align with the principles of Cognitive Load Theory (Sweller, 2020; DeLozier & Shey, 2023). Basic electronics engineering inherently requires novice learners to simultaneously process complex, multi-variable relationships involving voltage parameters, signal logic, and time-series execution dynamics. In traditional instructional paradigms relying on text-based programming or static circuit schematics, learners frequently experience high extraneous cognitive load induced by syntactical errors and abstract circuit representations. Microsoft MakeCode's block-programming environment mitigates these cognitive bottlenecks by translating abstract logic into manageable, manipulative visual structures. This reduction in extraneous cognitive load frees up working memory capacity, thereby optimizing germane cognitive processing toward active problem-solving and systemic comprehension of circuit behavior (Kastner-Hauler et al., 2022).

Furthermore, coupling MakeCode with an Inquiry-Based Learning framework effectively operationalizes Bruner's constructivist learning theory (Ozdem-Yilmaz & Bilican, 2025). Throughout the structured inquiry cycles—comprising orientation, hypothesis formulation, active experimentation, and synthesis—MakeCode's interactive virtual simulator serves as a high-fidelity feedback mechanism. When students adjust block parameters or alter input logic conditions, the simulator provides immediate, visual output state changes. This accelerated feedback loop empowers learners to dynamically test hypotheses, identify underlying logical flaws, and refine their internal mental models in real time, fostering a self-directed knowledge construction process (Ball et al., 2019; Voštinár et al., 2024).

When contrasted with the broader empirical literature, these findings both support and extend the conclusions reached by Quyen et al. (2023) and Mustan & Harizanov (2024), who documented that micro-device simulators enhance student engagement, learning motivation, and contextual understanding in general secondary science education. However, whereas prior investigations focused predominantly on introductory coding literacy or foundational scientific concepts, the present study demonstrates that block-based inquiry directly strengthens specialized, core technical engineering competencies within Technical and Vocational Education and Training (TVET) curricula. By addressing domain-specific technical tasks, this study demonstrates that block-based simulation is not merely an introductory tool, but a robust framework for high-level technical skill acquisition.

Despite these positive empirical outcomes, several methodological limitations must be explicitly acknowledged to contextualize the findings. First, the study's sampling strategy was restricted to two intact classroom cohorts ($N = 45$) situated within a single vocational institution, inherently constraining the broader statistical generalizability of the results across diverse TVET settings. Second, while the four-week intervention period was sufficient for evaluating immediate cognitive achievement gains, it did not incorporate longitudinal tracking to measure long-term knowledge retention or the transfer of learning to physical microcontroller hardware deployment. Third, the empirical framework did not disaggregate or control for individual variations in baseline digital literacy or prior computational thinking readiness, variables that may moderate the overall efficacy of the intervention.

To build upon these insights, future research directions should address these methodological constraints to advance the integration of simulation technologies in TVET contexts. Researchers should prioritize conducting multi-site longitudinal studies with larger sample sizes to evaluate the long-term sustainability of learning gains and assess performance transfer to real-world, physical hardware environments. Additionally, subsequent investigations should explore how individual baseline factors, such as prior computational thinking aptitude and technical self-efficacy, moderate learning outcomes under simulation-driven inquiry. Addressing these dimensions will provide a more comprehensive, nuanced understanding of how visual programming environments can be strategically tailored to optimize vocational engineering education.

4. CONCLUSION

This study demonstrates that the strategic integration of Microsoft MakeCode block-based simulation within an Inquiry-Based Learning framework significantly enhances both cognitive achievement and instructional engagement among vocational high school students in Basic Electronics Engineering. The empirical findings establish statistically significant post-test performance gains in the experimental cohort relative to conventional direct instruction, backed by an exceptionally large effect size ($Cohen's\ d = 1.79$) and a moderate-to-high normalized gain score (N-Gain $=$ 66.76%). By converting abstract schematic logic and complex circuit dynamics into interactive visual structures, this pedagogical approach effectively mitigates extraneous cognitive load, providing a robust conceptual bridge between theoretical engineering principles and practical execution.

From a practical perspective, these outcomes offer vital implications for Technical and Vocational Education and Training (TVET) practitioners seeking to modernize engineering curricula. Vocational instructors are strongly encouraged to adopt interactive block-programming visual environments as preliminary digital laboratory scaffolds prior to introducing physical hardware assembly. Coupling virtual simulation platforms with structured inquiry cycles creates an immersive learning environment that cultivates self-directed troubleshooting, spatial logic reasoning, and high-order technical problem-solving—competencies that are increasingly essential within complex Industry 4.0 technological paradigms.

Building upon these conclusions, several promising avenues for future research emerge to further advance TVET instructional design. Subsequent studies should expand upon this empirical foundation by conducting multi-institutional investigations with larger, more diverse student cohorts to enhance statistical generalizability and evaluate longitudinal knowledge retention over extended timeframes. Furthermore, future inquiry should examine the pedagogical transition bridge from visual block-based environments to text-based syntax (such as C/C++ or Python) in advanced microcontroller systems, while systematically assessing how individual differences in baseline computational thinking moderate overall learning trajectories.

REFERENCES

Akuma, F. V., & Callaghan, R. (2019). A systematic review characterizing and clarifying intrinsic teaching challenges linked to inquiry-based practical work. *Journal of Research in Science Teaching*, *56*(5), 619-648. https://doi.org/10.1002/tea.21516

Ball, T., Chatra, A., de Halleux, P., Hodges, S., Moskal, M., & Russell, J. (2019). Microsoft MakeCode: embedded programming for education, in blocks and TypeScript. In *Proceedings of the 2019 ACM SIGPLAN symposium on SPLASH-E* (pp. 7-12). https://doi.org/10.1145/3358711.3361630

Chandrasekaran, J. (2024). Redesigning Laboratory Courses in Information Technology Program: An Inquiry Based Learning Approach. *Journal of Engineering Education Transformations*, 921-930. https://www.journaleet.in/index.php/jeet/article/view/2523

Chen, F., & Chen, G. (2025). Learning analytics in inquiry-based learning: a systematic review. *Educational technology research and development*, *73*(4), 2131-2161. https://doi.org/10.1007/s11423-025-10507-9

DeLozier, C., & Shey, J. (2023). Using visual programming games to study novice programmers. *International Journal of Serious Games*, *10*(2), 115-136. https://doi.org/10.17083/ijsg.v10i2.577

Devine, J., Finney, J., De Halleux, P., Moskal, M., Ball, T., & Hodges, S. (2018). MakeCode and CODAL: intuitive and efficient embedded systems programming for education. *ACM SIGPLAN Notices*, *53*(6), 19-30. https://doi.org/10.1016/j.sysarc.2019.05.005

Devine, J., Finney, J., de Halleux, P., Moskal, M., Ball, T., & Hodges, S. (2019). MakeCode and CODAL: Intuitive and efficient embedded systems programming for education. *Journal of Systems Architecture*, *98*, 468-483. https://doi.org/10.1016/j.sysarc.2019.05.005

Hinostroza, J. E., Armstrong-Gallegos, S., & Villafaena, M. (2024). Roles of digital technologies in the implementation of inquiry-based learning (IBL): A systematic literature review. *Social Sciences & Humanities Open*, *9*, 100874. https://doi.org/10.1016/j.ssaho.2024.100874

Hodges, T., Jones, A., Del Olmo, L. P., Mishra, A., Caulfield, B., Kechadi, T., ... & Gilchrist, M. D. (2025). Computational modelling of protected and unprotected head impacts in rugby. *Bioengineering*, *12*(4), 361. https://doi.org/10.3390/bioengineering12040361

Kastner-Hauler, O., Tengler, K., Sabitzer, B., & Lavicza, Z. (2022). Combined effects of block-based programming and physical computing on primary students' computational thinking skills. *Frontiers in Psychology*, *13*, 875382. https://doi.org/10.3389/fpsyg.2022.875382

Kebede, A., Asgedom, A., & Asfaw, A. (2024). Is linking technical and vocational training with industry a bridge to employability? Lessons from a literature review for Ethiopia. *Cogent Education*, *11*(1), 2406721. https://doi.org/10.1080/2331186X.2024.2406721

Mustan, S. A., & Harizanov, K. V. (2024). The role of Microsoft platforms Makecode and Tinkercad in STEM education. *KNOWLEDGE-International Journal*, *66*(2), 217-221. https://www.ceeol.com/search/article-detail?id=1297569

Ozdem-Yilmaz, Y., & Bilican, K. (2025). Discovery learning—jerome bruner. In *Science education in theory and practice: An introductory guide to learning theory* (pp. 173-187). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-81351-1_11

Power, J. R., & Buckley, J. (2025). The Purpose of Second Level Engineering Education: An examination of curricular consistency and industry alignment. *Australasian Journal of Engineering Education*, *30*(1), 37-48. https://doi.org/10.1080/22054952.2024.2430155

Quyen, K. T., Van Bien, N., & Thuan, N. A. (2023). Micro: bit in Science Education: A systematic review. *Jurnal Penelitian dan Pembelajaran IPA*, *9*(1), 1-14. https://dx.doi.org/10.30870/jppi.v9i1.19491

Rafti, M., Milakis, E. D., Argyrakou, C. C., Vangeli, D. G., & Katsarou, M. C. (2025). Hands-on STEAM programming: Microteaching with Micro: bit for enhanced learning. *Journal of Hypermedia & Technology-Enhanced Learning*, *3*(2), 108-127. https://doi.org/10.58536/j-hytel.172

Rivera, J., & Baptista, V. E. G. (2024). Investigating the policies, the resources and gaps in the competencies of Bachelor of Technical-Vocational Teacher Education

(BTVTEd) prescribed by: TESDA and K to 12 Basic Education Curriculum and a proposed curriculum revision for Bachelor of Technical-Vocational Teacher Education in the Philippines. *Divine Word International Journal of Management and Humanities (DWIJMH)(ISSN: 2980-4817)*, *3*(4). https://doi.org/10.62025/dwijmh.v3i4.95

Scherer, R., Siddiq, F., & Tondeur, J. (2020). All the same or different? Revisiting measures of teachers' technology acceptance. *Computers & Education*, *143*, 103656. https://doi.org/10.1016/j.compedu.2019.103656

Sweller, J. (2020). Cognitive load theory and educational technology: J. Sweller. *Educational technology research and development*, *68*(1), 1-16. https://doi.org/10.1007/s11423-019-09701-3

Tedre, M., Simon, & Malmi, L. (2018). Changing aims of computing education: A historical survey. *Computer Science Education*, *28*(2), 158-186. https://doi.org/10.1080/08993408.2018.1486624

Voštinár, P. (2024). MakeCode arcade platform for game-based learning. *Computer Science and Information Systems*, *21*(3), 971-988. http://elib.mi.sanu.ac.rs/files/journals/csis/65/csisn67p971-988.pdf

Xu, Q., & Liu, Y. (2023). Research on the applicability of Makecode Arcade from the perspective of resource allocation. *3rd International Conference on Internet, Education and Information Technology (IEIT 2023)* (pp. 741-747). Atlantis Press. https://doi.org/10.2991/978-94-6463-230-9_88

Ye, J. H., He, Z., Bai, B., & Wu, Y. F. (2024). Sustainability of technical and vocational education and training (TVET) along with vocational psychology. *Behavioral Sciences*, *14*(10), 859. https://doi.org/10.3390/bs14100859

Yu, J. (2026). AI-Augmented Tangible Programming: Extending Tangible-MakeCode for Creative Learning. *Proceedings of the Twentieth International Conference on Tangible, Embedded, and Embodied Interaction* (pp. 1-4). https://doi.org/10.1145/3731459.3779040