

THE ROLE OF GENERATIVE AI IN LOW-CODE ROBOTICS IN STEM EDUCATION

Mafor Penn

*Department of Mathematics, Science and Technology Education, Faculty of Education,
University of Johannesburg (South Africa)*

Abstract

The integration of generative artificial intelligence (GenAI) tools into low-code robotics platforms is sharply changing how educational robotics is integrated in science, technology, engineering, and mathematics (STEM) education. This paper presents a combination of an empirical study and a theoretical positioning to examine the integration of GenAI into low-code robotics within STEM education. The main aim of the study is to examine how GenAI facilitates progress in learn-to-code from basic syntax mastery, such as block-based programming, to advanced system architecture design that integrates multi-agent coordination, sensor fusion, and the ethical deployment of robots in STEM disciplines. Following PRISMA 2020 guidelines, 42 studies conducted between 2022 and 2026 were reviewed from databases including Scopus, Web of Science, IEEE Xplore, and ACM Digital to establish the theoretical positioning. In the empirical study, three low-code-generated STEM robot designs and workflows were analysed deductively to provide a thick description of what is feasible. Findings from both the empirical study and the review suggest that GenAI tools can make programming more approachable for beginners by helping them get started with simple code, providing debugging support, and offering Personalised scaffolding. At the same time, these tools can help learners gradually build the confidence and skills needed to take on more complex tasks, like designing modular systems and improving their work through ongoing refinement. From the analysis of developed robots, although generative AI offers many benefits, several challenges remain when Gen AI tools are used for programming. These challenges include learners still over-relying on AI-generated responses; equity issues related to language and cultural biases that continue to surface in Gen AI responses. This gap clearly shows a need for a well-structured teaching framework that thoughtfully blends human input with Gen AI collaboration. The Implications of this review suggest the need for a conceptual pathway for GenAI-augmented low-code robotics curricula and identify directions for educators and researchers to implement and validate the proposed framework.

Keywords: *Generative AI, low-code robotics, prompt engineering, STEM education, system architecture, syntax mastery.*

1. Introduction

The digital divide in STEM education has historically been defined by the "syntax barrier". The Syntax barrier refers to the initial steep learning curves that beginners face due to their lack of familiarity with coding rules (Syntax) and which prevents the execution of a computer program (Sarkar & Drosos, 2025). In the aspect of coding, the high cognitive load required to master programming languages before a student can engage with robotics hardware also creates a barrier to the actualisation of any product. So, looking at this objectively, the syntax barrier has discouraged students from pursuing robotics because mastering complex code is frustrating. The emergence of low-code platforms sought to lower this barrier, yet the transition from simple block-based logic to complex, real-world robotic applications remains a significant pedagogical hurdle when teaching young minds how to code for artefact generation. This study examines theoretical drivers and three cases of 11th graders versed in using the Arduino coding cloud to generate STEM robots.

The advent of Generative AI (GenAI) has introduced a transformative layer to this ecosystem. By allowing students to describe robotic functions in natural language, GenAI tools facilitate a smoother transition through the coding hierarchy (Creely, 2026). However, the mere presence of these tools does not guarantee the educational quality of the Robots developed; there is a pressing need for a structured framework or pathway that ensures GenAI remains a scaffold rather than a crutch. This paper addresses this gap by proposing a structure which will eventually lead to the generation of a model that integrates human agency with AI efficiency.

2. Theoretically speaking

Integrating GenAI into educational robotics, particularly with hardware architectures such as Arduino or ESP32 microcontrollers, requires a strong theoretical foundation that combines cognitive scaffolding with computational pedagogy. This study is underpinned by the Cognitive Load Theory (CLT) and the emerging GenAI-Enabled Computational Thinking (Creely, 2026; Ouazaki et al., 2026). The problem that drives this study is situated in a phenomenon called syntax overload, which is an extraneous cognitive load caused by bracket placements, semicolons, and low-level C++ pin configurations, when beginners start to learn to code. Syntax overload usually diverts working memory from focus and commitment when developing artefacts, especially targeted for STEM. The research question that arises from this challenge is: How can GenAI be used to effectively, in combination with human agency, to support robot generation in STEM education?

Introducing GenAI as an epistemic partner via conversational "vibe coding" (Sarkar & Drosos, 2025) automates lower-level syntax tasks, enabling students to focus on higher-level problem-solving, systemic abstraction, and algorithmic design. This human-AI co-creation changes the learning cycle, shifting the primary task from deterministic code writing to an iterative, meta-cognitive exercise in prompting, executing, and evaluating AI output against a robot's physical actions (Creely, 2026). Consequently, GenAI integration offers a personalised cognitive scaffold that connects abstract mathematical and biological logic with tangible hardware engineering, reducing barriers to digital physical innovation while enhancing domain-specific STEM mastery and problem solving (Hsu et al., 2022).

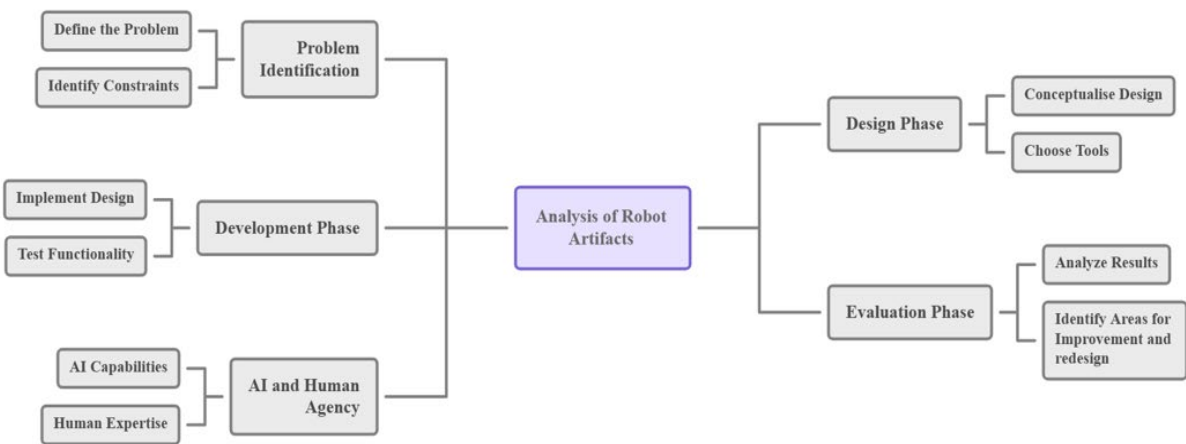
3. Methods and design

Two approaches were adopted for this study. First, a systematic review to understand the theoretical perspectives of low-code robotics, and second, a qualitative artefact analysis of the workflow and design of three students' robot creations.

For the theoretical perspective, the PRISMA 2020 guidelines were followed to select recent work from 2022 to 2026 to ensure a rigorous and transparent synthesis of current research. Databases interrogated during the search include Scopus, Web of Science, IEEE Xplore, and the ACM Digital Library. The inclusion criteria included articles and conference papers published between 2022 and 2026 that focused on GenAI in the context of robotics and STEM pedagogy. Book chapters and periodicals were excluded, as were web blogs. All due processes were followed to read and analyse these excerpts from all selected sources.

In the thematic analysis of the literature, 42 key studies were interrogated, and the findings were synthesised into three primary dimensions: learner scaffolding, system architecture design, and the ethical deployment of robots. For artefact analysis, all dimensions of the project planning and phases were analysed. Figure 1 below shows the dimensions that were considered from the design through to the analysis of artefacts.

Figure 1. Pathway from design to artefact analysis.



3.1. Phase I: Problem identification and ideational prompting

In this initial stage, the students' roles were shifted from passive recipients of tasks to "Prompt Engineers" of solutions. Students began by identifying a specific challenge within their community and the broader STEM field that could be addressed using crafted robots. After identification, students used GenAI to conduct preliminary research and feasibility studies to develop their proposed robot solution. Instead of a simple search, they engage in iterative chain-of-thought prompting to break a large problem into smaller robotic sub-systems. The analysis of student chats in this phase showed that GenAI acted as a brainstorming assistant, helping students refine their project's scope and identify necessary robotic components.

3.2. Phase II: Scaffolding the learn-to-code progression

This phase addresses the "syntax mastery" by using GenAI as a dynamic tutor. For two of the three students who did not have good Arduino coding knowledge, GenAI translated high-level logic into block-based structures that they could easily follow. While some students already knew quite well how to code with Arduino. As the beginner students progressed, the AI provided the corresponding text-based code, explaining the syntax line by line. All of this was accomplished using the Arduino AI Assistant within Arduino Cloud and occasionally ChatGPT and Claude via a split-screen browser window.

Analysing the prompt history and troubleshooting logs also showed that debugging loops offered students valuable learning opportunities. GenAI provided real-time, personalised scaffolding, helping students identify logic errors without the "blank page" anxiety that goes on when students are on their own.

3.3. Phase III: Advanced system architecture

The final phase of analysing the artefact moved beyond simple motor control to complex system design. Looking at how the participants integrated sensors in their robots and how they managed multi-agent coordination in their product. GenAI assists in writing complex filtering algorithms that enable multiple sensors to interact and also helps debug errors. Finally, students reported on how they used GenAI to simulate ethical dilemmas or environmental impacts of their robotic designs.

4. Results and discussions

The synthesis of studies and analysis of the actual artefact designs revealed that GenAI significantly boosts learner confidence by allowing them to take on more complex tasks through ongoing brainstorming and refinement loops with different GenAI tools. Table 1 below summarises some of the insights from the analysis.

Table 1. Core themes generated both from theory and cases.

Your Potential Finding	Core Theme	Supporting Academic Literature
Students who prompt iteratively for debugging show deeper mastery than those who ask for full code blocks.	Student Agency & Help-Seeking	(Creely 2026) highlights the risk of executive delegation vs. the benefits of instrumental exploration by students
Lowering the technical coding barrier allows students to focus on complex physics/biology logic.	Higher-Order Thinking (HOT)	Meta-analyses show Gen AI exerts a strong positive effect on problem-solving and critical thinking by acting as personalised scaffolding (Zhao et al., 2025).
Students develop robust prompt patterns (e.g., assigning the AI a "Persona") to produce better sensor calibration code.	Prompt Engineering Literacy	Mzwri & Turcsányi-Szabo (2025) note that explicit or experiential mastery of prompt patterns enhances autonomous learning and self-regulation.

From Table 1, the findings reveal the following themes as discussed below.

4.1. GenAI tools as a help-seeking tool

Findings reveal that higher-performing students typically engage in instrumental help-seeking. They use natural language to prompt the GenAI tool for conceptual guidance, error debugging, or modular code blocks, treating the AI as an adaptive tutor. Conversely, lower-performing or less confident students tend to seek executive help, delegating entire tasks to GenAI tools. For example, one of the logs from participants asked the Arduino assistant for "complete code for a titration robot" and passively copied and pasted the output without parsing its logic.

This finding aligns with Creely (2026), who found that, in an undergraduate setting, GenAI currently "mirrors student intent" rather than automatically optimising for learning. This means if a student approaches the AI passively, the tool complies passively, which can lead to accepting the first functional script without critical iteration or alternative exploration (Creely, 2026).

4.2. Cognitive scaffolding and the reduction of syntax overload

Building physical robots usually presents a steep learning curve because students must master hardware wiring and syntax-heavy coding (like C++ for Arduino) simultaneously for a viable product to be generated.

However, it was clear that a low-code AI assistant can act as a powerful cognitive bridge to help students achieve their development goals. Automating the syntax generation significantly reduced students' cognitive load. This allows 11th graders to focus their intellectual energy on the high-level logic of the robot (e.g., *How do we translate a change in light intensity into motor steps to track phototropism?*). This finding concurs with the literature on high school students' computational thinking, particularly in the dimensions of abstraction and algorithmic thinking when GenAI is integrated for assistance. Gen AI is able to eliminate initial syntax-related frustrations, which historically discourage novice learners, and maintains high learning enthusiasm, which supports exploration in a risk-tolerant environment (Hsu et al., 2022; Ouazaki et al., 2026).

4.3. The epistemological shift: From "doing code" to "evaluating output"

In traditional robotics tasks, success is measured by procedural correctness related to code compilation. However, the findings from this study reveal that an AI-mediated environment shifts toward meta-cognitive regulation and critical evaluation. Because the AI can generate code instantly, the students' primary challenge shifts from writing code to testing, validating, and debugging the code against physical reality. For example, if the GenAI generates code for the Snell's Law optics robot, the students must critically evaluate why the physical laser arm is overshooting its target and refine their prompts accordingly. These findings reflect what educational researchers describe as an epistemological shift in STEM education: transitioning from traditional knowledge transmission to knowledge co-construction mediated by intelligent systems (Pacheco-Olea & Villacís-Macías, 2026). The literature notes that AI shifts the focus away from purely deterministic problem-solving toward interpretability, validation, and the management of human-AI collaboration.

4.4. The challenges associated with GenAI integration remain

Several challenges, including the over-reliance trap, in which students may become overly dependent on AI-generated code without understanding its underlying logic, often arise during the process. In addition, equity and cultural biases inherent in GenAI training data continue to surface, potentially disadvantaging students from non-dominant cultural or linguistic backgrounds. The "Black Box" risk, in which successful implementation requires students to manually "annotate" AI-generated code to prove comprehension and ensure a "human-in-the-loop" approach, revealed gaps in some cases.

5. Conclusion and implications

Holistically, this study advocates that it is important to provide a roadmap for students to integrate GenAI when coding to create robots for STEM disciplines without sacrificing the requisite pedagogical integrity and the learning of core syntax through the learn-to-code progression. From all that was experienced and analysed, GenAI can become a powerful scaffold for robotics learning and not a replacement for the thinking process that must be applied in creation. Integrating GenAI into low-code environments also supports inquiry-based and project-based learning. In addition, GenAI was found to be useful in schools with limited specialist coding expertise, facilitating progression from block-based to text-based programming with adequate teacher guidance and a 'human-in-the-loop' pedagogical approach.

The best impact of low-code GenAI environments occurs when paired with learner agency, reflection, and contextual problem-solving. Future research should focus on developing frameworks that span diverse cultural contexts. For robotics educators, there must be an intentional shift from evaluating only code correctness to focusing more on students' ability to explain the thinking and logic behind their syntax, as well as their ethical use of GenAI to arrive at the end product.

References

- Creely, E. (2026). Generative AI to Foster Computational Thinking in Initial Teacher Education: A Thematic Literature Review and Model. *Behavioral Sciences*, 16(4), 575.
- Hsu, T.-C., Chang, C., Wu, L.-K., & Looi, C.-K. (2022). Effects of a pair programming educational robot-based approach on students' interdisciplinary learning of computational thinking and language learning. *Frontiers in Psychology*, 13, Article 888215. <https://doi.org/10.3389/fpsyg.2022.888215>
- Mzwri, K., & Turcsányi-Szabo, M. (2025). The impact of prompt engineering and a generative AI-driven tool on autonomous learning: A case study. *Education Sciences*, 15(2), 199. <https://doi.org/10.3390/educsci15020199>
- Ouaazki, A., Shibani, A., Knight, S., & Holzer, A. (2026). Generative AI-enhanced learning experiences for computational thinking: A systematic scoping review and design guidelines. *Computers and Education: Artificial Intelligence*, 100608.
- Pacheco-Olea, F. E., & Villacís-Macías, C. (2026). Artificial intelligence in STEM education: applications, learning outcomes, and methodological gaps (2016–2025). *Frontiers in Education*, 11, 1791361).
- Rinja, D., Oliveira, E. A., López-Pernas, S., Saqr, M., Specht, M., & Misiejuk, K. (2026). Unpacking Vibe Coding: Help-Seeking Processes in Student-AI Interactions While Programming. *arXiv preprint arXiv:2604.27134*.
- Sarkar, A., & Drosos, I. (2025). Vibe coding: programming through conversation with artificial intelligence. *arXiv preprint arXiv:2506.2325*
- Zhao, Y., Yue, Y., Sun, Z., Jiang, Q., & Li, G. (2025). Does Generative Artificial Intelligence Improve Students' Higher-Order Thinking? A Meta-Analysis Based on 29 Experiments and Quasi-Experiments. *Journal of Intelligence*, 13(12), 160.