Generative AI-Augmented End-User Development of Creative Physical Computing Projects

Monica Maceli[1*] and Haoxiang Zhang[1]

[1] Pratt Institute School of Information, 144 W 14th St. New York, NY, USA

Abstract
End-user development (EUD) aims to empower non-programmers to design and adapt computational systems to their emergent needs, a challenge recognized as more complex in physical computing contexts requiring both programming and hardware knowledge. Generative AI (GenAI) tools offer new opportunities to lower technical barriers by translating end users' natural language needs into code and assisting in debugging and trouble-shooting. This paper presents an in-progress study examining how students use GenAI tools during the ideation, prototyping, and debugging of creative physical computing projects using the BBC micro:bit. Through a structured design intervention, participants develop personally meaningful projects with optional use of GenAI, enabling comparison of AI-assisted and non-AI workflows. The study investigates students' choices around using GenAI, and its impact on design processes, problem-solving, and perceptions of creative ownership. Connections to the CoPDA workshop are proposed and potential contributions are contextualized within the EUD goal of empowering end users in shaping their designed environments.

Keywords
End-User Development, Generative AI, Physical Computing

1. Introduction

End-user development (EUD) describes an approach to technology design and development in which end users are empowered to craft and customize designed objects throughout their lifecycle. End-user development seeks to enable non-programmers to customize computational systems during use [1]. In shifting design and development power towards end users, numerous challenges emerge in how we, as information system designers, may support end users in evolving their systems in use. Approaches such as trigger-action programming, visual programming environments, and design critics [e.g. 2], among others, have been employed to create flexible environments in which end users can design their own system behaviors to meet their emergent needs.

With the arrival of maker platforms such as Arduino, Raspberry Pi, and the BBC micro:bit, EUD has expanded into physical computing contexts where users can prototype interactive systems using sensors and microcontrollers. In such physical computing contexts, end users must grapple not only with software, but also with hardware, wiring, and other aspects of design unique to physical computing. Not surprisingly, non-technical end-users are still struggling with physical prototyping with those technologies, as they still require foundational programming knowledge to implement and modify system behaviors [3, 4, 5].

Recent advances in Generative AI (GenAI) allow users to translate natural language intentions into software code, potentially lowering technical barriers for novice software programmers, and have attracted significant interest in EUD research. The potential for GenAI to serve as a personalized coding assistant, tutor, and/or design critic may more deeply assist users beyond just the coding

[1] *Proceedings of the 10th International Workshop on Cultures of Participation in the Digital Age (CoPDA 2026): Exploring the Relationship between EUD, AI-Assisted Development, and Meta-Design, June 2026, Venice, Italy.*
* Corresponding author.
mmaceli@pratt.edu (M. Maceli); hzhang68@pratt.edu (H. Zhang);
0000-0003-3132-6401 (M. Maceli); 0009-0001-0556-1152 (H. Zhang)

aspect of physical computing projects. Such GenAI tools, however, have been intensely critiqued for their environmental impact and algorithmic bias, among other concerns.

The purpose of this in-progress research work is to explore how students use generative AI tools to support ideation, problem-solving, and prototyping in physical computing projects. Using the BBC micro:bit, a low cost physical computing prototyping board, the study aims to understand how GenAI-assisted development influences students' design processes during hands-on prototyping activities. The research questions investigated shall include: *how and why do students choose to use generative AI tools to support ideation, problem-solving, writing code, and debugging code in physical computing projects? How does the use of such generative AI tools impact students' perception of creative ownership of the resulting physical computing project?*

The research questions, and any other emergent themes, will be explored through the lens of end-user development with the goal of more deeply understanding how, and if, such GenAI tools can assist end users in bridging the gap between digital and physical computing prototyping.

2. Background

Though GenAI tools are still relatively new technologies, numerous research studies have explored how (and if) such tools can lower technical barriers in assisting non-expert programmers in writing software code, and whether these very activities can be considered EUD behaviors.

While GenAI models have demonstrated promising performance in benchmark evaluations of code generation, their accuracy is often associated with the quality and structure of prompts provided to guide the model [6, 7, 8]. Technically-skilled users, such as programmers, achieve better results when interacting with AI models through iterative, conversational prompting strategies [9, 10]. Within EUD literature, survey research suggests that conversational, step-by-step interaction is the dominant approach to prompting employed in AI-EUD environments of research [11].

For non-technical end users, this conversational interaction paradigm may distribute agency between AI and users, shifting users' role from directly authoring code to specifying intentions [11, 12]. Users then are typically required to review, edit, and validate the generated code, which can be a significant burden on novice programmers [13, 14]. Such shifts also introduce new challenges, as users may overgeneralize the success or failure of their prompts and develop over-reliance on AI coding systems while reducing their verification practices [8, 13]. This dynamic can lead to the accumulation of unverified generated code, which may contain errors, inconsistencies, or fabricated details. When the executed outcomes diverge from their intent, non-technical users may struggle to interpret, modify, or debug the generated code [14]. Even if syntactically-correct code is generated, it may include logic errors; Jabrw and Sarhan [7] suggest that beginning programmers need a way to visualize this logical behavior in order to identify logic errors.

Physical computing environment, such as robotics, may further amplify the challenges as end-user developers must understand both programming and electronics concepts and reason about the relationship between virtual program logic and physical circuits in order to diagnose problems that arise [3]. GenAI tools can generate syntactically correct code, but produce unexpected behaviors when executed on physical devices, as was explored in the context of robotics by Chen and Huang [6]. Bimbatti et al. [15] addressed this problem by restricting the GenAI's output to a fixed series of visual blocks, to avoid logic errors in robot programming, in a preliminary study.

Debugging in such contexts often requires reasoning across both code and physical system behavior. Prior work shows that diagnosing whether errors originate from the program or the physical circuit can be difficult, and sometimes modifying the wrong component can introduce further issues [3]. Studies also suggest that, from a meta-design perspective, having a visualizer or simulator may help to mitigate the challenge of verifying and repairing execution errors (e.g. [11]).

Research has also explored whether such GenAI systems are fundamentally compatible and in alignment with the goals and priorities of EUD. Fischer [16] posits that existing AI systems are oriented towards automatically adapting towards perceived user needs, with a subsequent loss of transparency and end-user control, whereas the traditional focus of EUD has been in adaptable systems wherein users control and change the system consciously. In their survey of AI-related literature in EUD, Gargioni et al. [11] find that most of the papers focus on novice programmers as end users. Thus, these studies fundamentally differ from traditional EUD concerns, which aimed to empower domain experts to solve their emergent and novel problems programmatically. Studies suggest that GenAI tools may provide benefits more for experienced programmers and less so for novice programmers (e.g. [12]), again differing from traditional EUD concerns around supporting non-technical users.

3. Proposed Design Intervention

We plan a *hands-on design intervention* centered on AI-augmented end-user development (EUD) for physical computing using the BBC micro:bit, to be tested on student participants at Pratt Institute School of Information. This design intervention has been crafted to leverage both existing research findings within EUD, presented above, as well as to more deeply explore open (and problematic) questions in this realm, all of which are of high relevance to the CoPDA 2026 Workshop.

First, the long-standing motivation of "true" end-user developers, as discussed in the prior section, is the need for domain experts to solve personally-meaningful problems, in contexts of use that tend to emerge long after the formal design phase of the project has concluded. The design intervention presented here invites the participants to ideate around personally meaningful problems, as opposed to simply considering any novice programmer to be an EUD practitioner in the new era of GenAI. However, our design intervention takes place in an educational context, in which the end users cannot (yet) be considered to be domain experts. To resolve this mismatch with more traditional domain oriented design environments [17], we take the approach that the combination of novice end-user developers, augmented through GenAI systems, may then serve as de facto domain experts, and be studied in educational settings as in prior work [e.g. 18].

We also attempt to explore further the EUD research addressing: effective prompting in GenAI, the use of visual programming languages, as well as the need to simulate physical computing behavior and wiring, all as necessary in guiding novices in designing such complex physical systems. The design intervention will incorporate all these approaches such that they can be assessed for their effectiveness with the participants. Notably, the web-based educational code editor Microsoft MakeCode to be used includes a blocks-based visual programming language, as well as a panel to preview and simulate the behavior of the micro:bit code created, before downloading to the physical device (Figure 2, below).

And lastly, we also introduce the possibility for participants to decide not use GenAI in ideating, programming, and debugging their design solutions. While participants may individually choose to refrain from using GenAI, this may be difficult to uphold in our group-based design intervention described below, where design and coding work is shared. The hope is that it will create dialogue amongst participants as to the value and need for such tools. Such tools have been extremely polarizing in public and academic discourse with real and valid concerns including: environmental impact, computational bias, lack of algorithmic transparency, and surveillance applications. As we consider EUD's goal of empowering end users to shape their designed environments, it is in striking contrast to how GenAI tools may disempower large populations and reduce algorithmic transparency.

3.1. Overview of the Design Intervention Study Procedure

The session will begin with a brief introduction to the workshop and research goals, followed by the collection of informed consent from participants. Participants will then participate in short tutorials introducing micro:bit fundamentals and the use of conversational AI to support physical computing and prototyping (Figure 1, below).

Guided AI Prompt Activity

Starter Prompt:

> *"Help me write a microbit program with **MakeCode** that prints my name when you press button A."*

Notice that the AI will give steps to drag and drop blocks in MakeCode to create the program.

You can also ask AI to generate JavaScript or Python code. The code can be pasted directly into the MakeCode JavaScript or Python view if you want a faster workflow, e.g.:

> *"Help me write a microbit program with **Javascript** that prints my name when you press button A."*

Figure 1: Suggested prompts for design intervention participants to experiment with in developing their micro:bit creative projects using the web-based educational code editor Microsoft MakeCode (https://makecode.microbit.org/).

Participants will work in small groups (2–3 participants) to identify everyday problems, develop solution concepts using a structured brainstorming worksheet, and prototype a micro:bit project with the optional assistance of selected AI models. Figure 2 (below) demonstrates the web-based programming interface in which participants ask an AI assistant for guidance using open-ended prompts and then implement the suggested functionality in the micro:bit JavaScript environment provided in Microsoft MakeCode. This coding environment also includes the ability to toggle into a view of a simulated visual of the device running their code (not depicted). To the right in Figure 2, the participants then download and test their code on the physical micro:bit device. The entirety of the design workshop will take approximately two hours, with a total of 10 to 20 participants. At the end of the intervention, consenting participants will complete a short wrap-up questionnaire reflecting on their experiences and perspectives during and after the session.

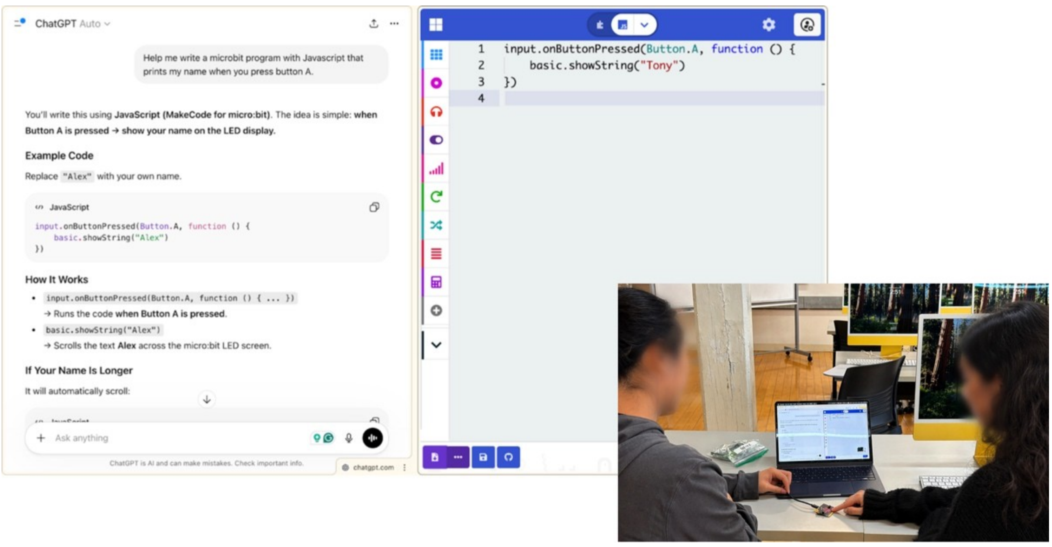

Figure 2: Example of the AI-assisted micro:bit prototyping tools used in the workshop, including a screenshot of the coding interface on Microsoft MakeCode (left) and a pair of participants testing the physical device (right).

The initial design intervention workshop, scheduled for early May 2026 will include 13 graduate student participants in total, forming six participant groups. Researchers will observe and take field notes on the use, or non-use, of GenAI during group design work, as well as the application (for ideation, code creation, debugging, or other emergent behaviors). Images of the resulting groups' design objects and transcripts from the GenAI chatbot interactions will be collected, as well as the results of the final individual survey which will contain both closed and open-ended questions exploring the research questions presented in Section 1 of this paper. Qualitative data will be analyzed using thematic coding to identify patterns and themes relevant to the research questions. Inductive coding will be conducted iteratively by three researchers on the GenAI chatbot transcripts.

4. Connections to CoPDA 2026 Workshop

We find our proposed work to be of high relevance to the CoPDA 2026 workshop and in close alignment to the AVI 2026 conference theme of "*Interactive Creativity: Agencies, Interfaces, and Ethics*". Specifically in the context of the CoPDA workshop, we directly address the motivating theme of how EUD, Meta-Design, and Cultures of Participation could and should evolve in the AI era. Notably our research questions explore not just the technical barriers to EUD in physical computing, but also the larger concerns in fostering a culture of participation, in assessing why users may or may not use GenAI tools and how their perceptions of creativity and ownership are impacted by such choices. We also invite discussion around shifts in the identity of the end-user developer, as well as more broadly if GenAI tools are philosophically compatible with the goals of EUD. Lastly, as our work is currently in progress, there is significant opportunity for the CoPDA participants to contribute their feedback to the project, as well as identify future research collaboration possibilities. We expect to have initial study results from the May 2026 design intervention completed for presentation at the CoPDA 2026 workshop.

Declaration on Generative AI

During the preparation of this work, the authors used OpenAI's ChatGPT (GPT-5.3) in order to: Grammar and spelling check, Abstract drafting. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] H. Lieberman, F. Paternò, M. Klann, V. Wulf, End-user development: an emerging paradigm, in: H. Lieberman, F. Paternò, V. Wulf (Eds.), End User Development, volume 9 of Human-Computer Interaction Series, Springer, Dordrecht, 2006. doi:10.1007/1-4020-5386-X_1.

[2] Fischer, G., Lemke, A.C., Mastaglio, T.W., & Morch, A.I. The role of critiquing in cooperative problem solving. ACM Transactions on Information Systems (TOIS), 9, 1991, pp. 123 - 151.

[3] T. Booth, S. Stumpf, J. Bird, S. Jones, Crossed wires: investigating the problems of end-user developers in a physical computing task, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, CHI '16, ACM, New York, NY, USA, 2016, pp. 3485–3497. doi:10.1145/2858036.2858533.

[4] M. Maceli, Low-cost physical computing platforms for end-user prototyping of smart home systems, Behaviour & Information Technology 40 (2021) 997–1007. doi:10.1080/0144929X.2021.1918248.

[5] M. Maceli, An internet-of-things end-user development approach to environmental monitoring of cultural heritage archives, in: A. Malizia, S. Valtolina, A. Morch, A. Serrano, A. Stratton (Eds.), End-User Development. IS-EUD 2019, volume 11553 of Lecture Notes in Computer Science, Springer, Cham, 2019, pp. 149–157. doi:10.1007/978-3-030-24781-2_10.

[6] J. T. Chen, C. M. Huang, Forgetful large language models: lessons learned from using LLMs in robot programming, in: Proceedings of the AAAI Symposium Series, 2023, pp. 508–513. doi: https://doi.org/10.1609/aaaiss.v2i1.27721.

[7] S. K. Jabrw, Q. I. Sarhan, Evaluating large language models for Arduino code generation, ARO–The Scientific Journal of Koya University 14(1) (2025) 75–85. doi:10.14500/aro.12344.

[8] J. D. Zamfirescu-Pereira, R. Y. Wong, B. Hartmann, Q. Yang, Why Johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts, in: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI '23, ACM, New York, NY, USA, 2023, pp. 1–21. doi:10.1145/3544548.3581388.

[9] J. Shin et al., Prompt engineering or fine-tuning: an empirical assessment of LLMs for code, in: 2025 IEEE/ACM 22nd International Conference on Mining Software Repositories, MSR '25, IEEE, Ottawa, ON, Canada, 2025, pp. 490–502. doi:10.1109/MSR66628.2025.00082.

[10] S. Zhong, Y. Zou, B. Adams, Developer-LLM conversations: an empirical study of interactions and generated code quality, 2025. URL: https://arxiv.org/pdf/2509.10402.

[11] L. Gargioni, B. R. Barricelli, D. Fogli, A. Locoro, Exploring the role of end users in performing EUD with large language models, in: Proceedings of the 8th International Workshop on Cultures of Participation in the Digital Age, CoPDA 2024, CEUR-WS.org, Vol. 3685, 2024. URL: https://ceur-ws.org/Vol-3685/short3.pdf.

[12] M. Q. Feldman, C. J. Anderson, Non-expert programmers in the generative AI future, in: Proceedings of the 3rd Annual Meeting of the Symposium on Human-Computer Interaction for Work, CHIWORK '24, ACM, New York, NY, USA, 2024, pp. 1–19. doi:10.1145/3663384.3663393.

[13] G. O'Brien, A. Parker, N. Eisty, J. Carver, More code, less validation: risk factors for over-reliance on AI coding tools among scientists, 2025. URL: https://arxiv.org/abs/2512.19644.

[14] P. Vaithilingam, T. Zhang, E. L. Glassman, Expectation vs. experience: evaluating the usability of code generation tools powered by large language models, in: CHI Conference on Human Factors in Computing Systems Extended Abstracts, CHI EA '22, ACM, 2022. doi:10.1145/3491101.3519665.

[15] G. Bimbatti, D. Fogli, L. Gargioni, Can ChatGPT support end-user development of robot programs, in: Proceedings of the 9th International Symposium on End-User Development, IS-EUD 2023, 2023.

[16] G. Fischer, Adaptive and adaptable systems: differentiating and integrating AI and EUD, in: L. D. Spano, A. Schmidt, C. Santoro, S. Stumpf (Eds.), End-User Development. IS-EUD 2023, volume 13917 of Lecture Notes in Computer Science, Springer, Cham, 2023. doi:10.1007/978-3-031-34433-6_1.

[17] Fischer, G. Domain-oriented design environments. Autom Software Eng 1, 177–203 (1994). https://doi.org/10.1007/BF00872289

[18] Andersen, R., Mørch, A. I., & Litherland, K. T. Collaborative learning with block-based programming: investigating human-centered artificial intelligence in education. Behaviour & Information Technology, 41, 9, 2022, pp. 1830–1847. https://doi.org/10.1080/0144929X.2022.2083981