AgentCube: AI-Assisted In-Situ Authoring for Physical Computing

Boyang Zhou
Paul G. Allen Center for Computer Science & Engineering
University of Washington
Seattle, Washington, USA
zby2003@cs.washington.edu

Jaewook Lee
Paul G. Allen School of Computer Science & Engineering
University of Washington
Seattle, Washington, USA
jaewook4@cs.washington.edu

Jon E. Froehlich
Paul G. Allen School of Computer Science & Engineering
University of Washington
Seattle, Washington, USA
jonf@cs.uw.edu

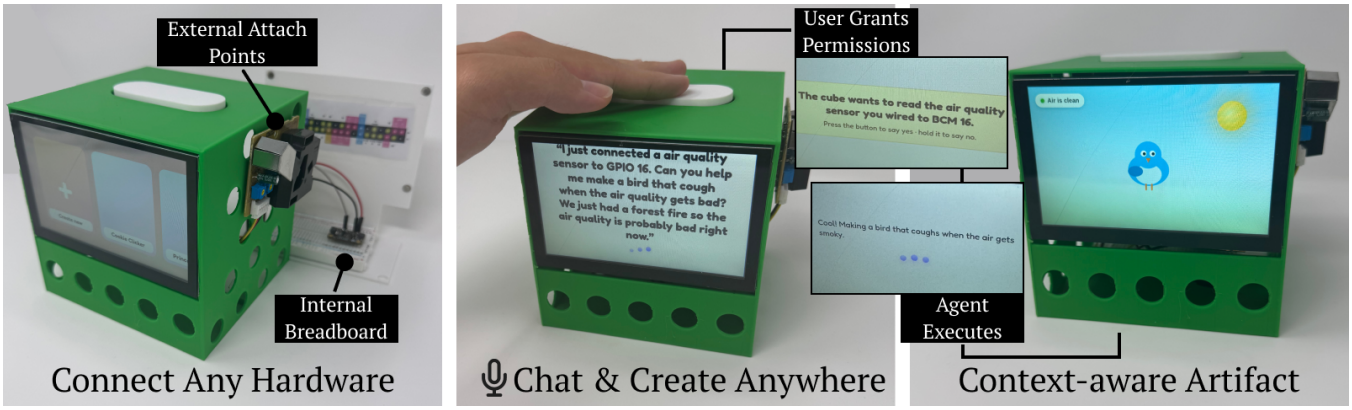

Figure 1: An example workflow with AgentCube: (1) The user connects components via external attach points (e.g., servos, buttons) or the internal breadboard (e.g., speakers, IMU). (2) The user describes their desired artifact aloud. The onboard agent clarifies intent, requests permissions, and narrates its progress. (3) The application is evaluated and refined in-situ. For example, the pictured bird artifact is programmed to cough when the air quality is poor. The user can verbally instruct the AgentCube to cough less frequently during wildfire season, without bringing the setup back to the workbench.

Abstract

Physical computing kits introduce novices to hardware programming, but the creation process is often separated from the environment where the resulting artifacts will be tested and used — such as building an interactive nightlight in a well-lit classroom. Every tweak typically requires a back-and-forth trip to the workbench. We present AgentCube, a portable, AI-agent-powered platform enabling novices to program interactive and context-aware hardware artifacts using voice, directly at the deployment site. AgentCube employs a human-agent workflow that translates a novice's spoken concepts into executable behaviors grounded in hardware states and, optionally, the environmental context. By eliminating the need for keyboards and screen-based programming interfaces, AgentCube allows users to author, test, and refine their creations in situ. In this demo, we present AgentCube's all-in-one architecture, its hardware-aware agent harness, and example artifacts enabled by AgentCube's in-situ authoring capability.

CCS Concepts

• **Human-centered computing** → **Ubiquitous and mobile computing**.

Keywords

large language model agents, physical computing, education technology, in-situ authoring, context-aware systems

ACM Reference Format:
Boyang Zhou, Jaewook Lee, and Jon E. Froehlich. 2026. AgentCube: AI-Assisted In-Situ Authoring for Physical Computing. In *The 39th Annual ACM Symposium on User Interface Software and Technology (UIST Adjunct '26), November 02–05, 2026, Detroit, MI, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3830397.3842474

1 Introduction and Related Work

Physical computing kits such as Arduino, the BBC micro:bit [18], and MakerWear [13] have shown success among novices who are learning to create hardware. However, the process of building and programming with these kits often occurs in front of a computer or inside a classroom, separate from where the artifacts are intended to be deployed (e.g., a sensor-driven smart hockey stick created in class can only be tested on the ice) [4, 13, 20]. **In-situ authoring** [11, 15, 16, 21, 22] has demonstrated that grounding authoring in the physical environment supports contextually relevant creation, yet existing systems target virtual content [16] or pre-built modules [21], not the open-ended programming of circuits. Closest to our

UIST Adjunct '26, Detroit, MI, USA

ACM ISBN 979-8-4007-2855-6/26/11
https://doi.org/10.1145/3830397.3842474

goal, Microcode [10] brings portable, computer-free programming to the micro:bit, but through a fixed vocabulary and no explicit incorporation of environmental context. Despite tools that aid circuit debugging and generation [2, 8, 17], none explicitly allow users to build or remix an artifact directly where it is deployed, nor do they integrate environmental context into the authoring process.

Voice-driven coding agents now let people create complex programs via natural language [1, 7]. Yet, these software-only workflows lack a direct way to interface with hardware [9], leaving them unable to inspect wiring [5], sensor readouts, or device state.

We introduce **AgentCube**, a physical computing platform designed for in-situ authoring capabilities, powered by a hardware-aware coding agent. By replacing traditional keyboards and programming environments with voice-based input and AI-agent-powered coding, AgentCube is designed to encourage novices to create and iterate upon their projects in varied physical environments, while incorporating context into their final design. With AgentCube, users describe their idea aloud [14] and physically connect the desired sensor/actuator components, while the agent generates pin-level wiring guidance and verifies each connection against live hardware scans. The onboard coding agent then dynamically handles low-level interfacing by jointly reasoning over the software, the hardware (AgentCube), and the readout of the sensors (environmental context). Ultimately, the user can tailor the software-hardware system to both their imagination and the context of the deployment site. The created artifacts are then captured in a reusable software format, allowing easy rewiring or remixing. A single AgentCube lets users create and store multiple distinct artifacts and replay them later.

Our contributions are: (1) The demonstration of AgentCube, a physical computing platform whose authoring loop – voice input, agentic coding, live preview, hardware verification – runs on the deployed device itself; (2) a hardware-aware agent harness and context manifest that ground LLM code generation in live pin state under user-permissioned access; and (3) example artifacts created that highlight the practical utility of this approach.

2 System Implementation

All-in-One Hardware. AgentCube does not require external hardware to function (except network connectivity). Built on a Raspberry Pi, AgentCube avoids firmware recompilation cycles. The fully portable, battery-powered system is equipped with a touchscreen, a microphone, and exposed physical pins.

Hardware-Aware Agent Harness. The sandboxed environment of AgentCube allows vanilla coding agents (we utilize Claude Code running Opus 4.7, with OpenAI Whisper for transcription) to securely interact with both onboard and external hardware. The harness constrains and observes the agent, granting access to physical pins and system root privileges strictly upon user permission.

Context-Aware Hardware Manifest. To provide the agent with accurate hardware context, we supply a local manifest divided into a static Raspberry Pi specification and a writable memory bank tracking connected components. To ground the system, the agent dynamically generates small testing scripts to poll the hardware, utilizing passive telemetry (e.g., GPIO polling, I2C bus enumeration) triggered by environmental states or user actions. By cross-referencing this live hardware telemetry against the user's intended design, the agent can detect physical mismatches and guide the user through wiring corrections. Once validated, the agent can selectively reuse these configurations across different artifacts. Unlike prior debugging aids that gain hardware observability by instrumenting the workbench [8], or that script assembly steps for a fixed set of components [14], AgentCube generates and verifies wiring guidance for arbitrary user-declared components.

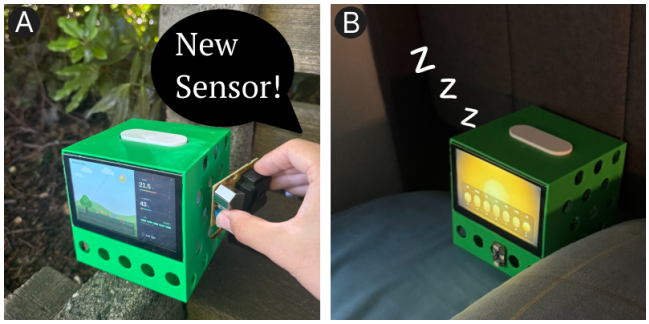

Figure 2: Hardware applications made by AgentCube in-situ (section 3). A: Weather Garden; B: Sleep Tracker.

Human-Agent Interface. AgentCube stores each artifact as a self-contained web application, executed through a local browser backend. To enable instant live previews without manual recompilation [6], the agent generates a lightweight local API that safely bridges the browser interface with the hardware pins. The authoring workflow operates as a turn-by-turn conversation between user and agent, following best practices of human-AI communication [3].

3 Example Artifacts

We present two artifacts demonstrating AgentCube's in-situ authoring capabilities (Figure 2):

Weather Garden. A user deploys a garden "digital twin" using a temperature and humidity sensor. Days later, they attach a dust sensor to the GPIO pins in the garden and say, "I added an air quality sensor..." The agent automatically updates the hardware manifest and integrates the data. However, upon viewing the live display, the user realizes raw air quality numbers are hard to interpret compared to temperature readings. They tell AgentCube, "Change the air quality reading to a good/bad scale." The agent updates the visualization schema accordingly, allowing the user to iterate on hardware and software directly in the deployment environment.

Sleep Tracker. AgentCube sleep tracker logs sleep (how long the room remains dark) using a spectrometer. However, the user forgot that their bedroom never gets completely dark. The user puts the AgentCube on the bed, turns off the lights and says, "Set the current lighting level as dark." The agent queries the live hardware readout from the spectrometer and edits the threshold logic.

4 Discussion and Future Work

AgentCube presents a new approach to physical computing kits by embedding open-ended authoring capabilities directly within a

portable computer. This shifts the creative process into the locations where artifacts are deployed, and allows the environment to directly inform the design. Future work will evaluate AgentCube in real-world settings to understand how novices collaborate with agents to author, deploy, and iterate on context-aware hardware.

There are several limitations of the current iteration of AgentCube. First, AgentCube cannot automatically retrieve and understand detailed specifications for complex electronic components, requiring users to maintain a foundational understanding of hardware pinouts. Because hardware verification relies on digital-side scans, purely analog faults cannot be caught either. Second, by allowing novices to "vibe code" hardware applications, there is a pedagogical risk of over-reliance [12]. AgentCube relocates the abstraction boundary [19] to the physical layer (how to connect the circuits) rather than remove it. While formal educational evaluation is outside the scope of this demonstration, whether hiding coding complexity supports or hinders learning is a research question that future classroom deployments will examine directly.

References

[1] [n. d.]. Continue local sessions from any device with Remote Control. https://code.claude.com/docs/en/remote-control

[2] Fraser Anderson, Tovi Grossman, and George Fitzmaurice. 2017. Trigger-Action-Circuits: Leveraging Generative Design to Enable Novices to Design and Build Circuitry. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. Association for Computing Machinery, New York, NY, USA, 331–342. doi:10.1145/3126594.3126637

[3] Gagan Bansal, Jennifer Wortman Vaughan, Saleema Amershi, Eric Horvitz, Adam Fourney, Hussein Mozannar, Victor Dibia, and Daniel S. Weld. 2024. Challenges in Human-Agent Communication. arXiv:2412.10380 [cs.HC] doi:10.48550/arXiv.2412.10380

[4] Paulo Blikstein. 2015. Computationally Enhanced Toolkits for Children: Historical Review and a Framework for Future Design. *Foundations and Trends® in Human–Computer Interaction* 9, 1 (Dec. 2015), 1–68. doi:10.1561/1100000057

[5] Tracey Booth, Simone Stumpf, Jon Bird, and Sara Jones. 2016. 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)*. Association for Computing Machinery, New York, NY, USA, 3485–3497. doi:10.1145/2858036.2858533

[6] Lautaro Cabrera, John H. Maloney, and David Weintrop. 2019. Programs in the Palm of Your Hand: How Live Programming Shapes Children's Interactions with Physical Computing Devices. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children (IDC '19)*. Association for Computing Machinery, New York, NY, USA, 227–236. doi:10.1145/3311927.3323138

[7] Valerie Chen, Ameet Talwalkar, Robert Brennan, and Graham Neubig. 2026. Code with Me or for Me? How Increasing AI Automation Transforms Developer Workflows. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26)*. Association for Computing Machinery, New York, NY, USA, 1–19. doi:10.1145/3772318.3790850

[8] Daniel Drew, Julie L. Newcomb, William McGrath, Filip Maksimovic, David Mellis, and Björn Hartmann. 2016. The Toastboard: Ubiquitous Instrumentation and Automated Checking of Breadboarded Circuits. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, Tokyo Japan, 677–686. doi:10.1145/2984511.2984566

[9] Zachary Englhardt, Richard Li, Dilini Nissanka, Zhihan Zhang, Girish Narayanswamy, Joseph Breda, Xin Liu, Shwetak Patel, and Vikram Iyer. 2024. Exploring and Characterizing Large Language Models for Embedded System Development and Debugging. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, 1–9. doi:10.1145/3613905.3650764

[10] Kobi Hartley, Elisa Rubegni, Lorraine Underwood, Joe Finney, Thomas Ball, Steve Hodges, Peli De Halleux, James Devine, Eric Anderson, and Michał Moskal. 2024. Meet MicroCode: a Live and Portable Programming Tool for the BBC micro:bit. In *Proceedings of the 23rd Annual ACM Interaction Design and Children Conference (IDC '24)*. Association for Computing Machinery, New York, NY, USA, 355–370. doi:10.1145/3628516.3656995

[11] Valentin Heun, James Hobin, and Pattie Maes. 2013. Reality editor: programming smarter objects. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, Zurich Switzerland, 307–310. doi:10.1145/2494091.2494185

[12] Majeed Kazemitabaar, Justin Chow, Carl Ka To Ma, Barbara J. Ericson, David Weintrop, and Tovi Grossman. 2023. Studying the effect of AI Code Generators on Supporting Novice Learners in Introductory Programming. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–23. arXiv:2302.07427 [cs.HC]. doi:10.1145/3544548.3580919

[13] Majeed Kazemitabaar, Jason McPeak, Alexander Jiao, Liang He, Thomas Outing, and Jon E. Froehlich. 2017. MakerWear: A Tangible Approach to Interactive Wearable Creation for Children. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 133–145. doi:10.1145/3025453.3025887

[14] Yoonji Kim, Youngkyung Choi, Daye Kang, Minkyeong Lee, Tek-Jin Nam, and Andrea Bianchi. 2020. HeyTeddy: Conversational Test-Driven Development for Physical Computing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 4 (Sept. 2020), 139:1–139:21. doi:10.1145/3369838

[15] Tobias Langlotz, Stefan Mooslechner, Stefanie Zollmann, Claus Degendorfer, Gerhard Reitmayr, and Dieter Schmalstieg. 2012. Sketching up the world: in situ authoring for mobile Augmented Reality. *Personal and Ubiquitous Computing* 16, 6 (Aug. 2012), 623–630. doi:10.1007/s00779-011-0430-0

[16] Jaewook Lee, Filippo Aleotti, Diego Mazala, Guillermo Garcia-Hernando, Sara Vicente, Oliver James Johnston, Isabel Kraus-Liang, Jakub Powierza, Donghoon Shin, Jon E. Froehlich, Gabriel Brostow, and Jessica Van Brummelen. 2025. ImaginateAR: AI-Assisted In-Situ Authoring in Augmented Reality. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. Association for Computing Machinery, New York, NY, USA, 1–21. doi:10.1145/3746059.3747635

[17] Will McGrath, Daniel Drew, Jeremy Warner, Majeed Kazemitabaar, Mitchell Karchemsky, David Mellis, and Björn Hartmann. 2017. Bifröst: Visualizing and Checking Behavior of Embedded Systems across Hardware and Software. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. Association for Computing Machinery, New York, NY, USA, 299–310. doi:10.1145/3126594.3126658

[18] Micro:bit Educational Foundation. 2016. BBC micro:bit. https://microbit.org/. Accessed: 2026-06-23.

[19] Mitchel Resnick, Robbie Berg, and Michael Eisenberg. 2000. Beyond Black Boxes: Bringing Transparency and Aesthetics Back to Scientific Investigation. *Journal of the Learning Sciences* 9, 1 (Jan. 2000), 7–30. doi:10.1207/s15327809jls0901_3

[20] Mitchel Resnick and Brian Silverman. 2005. Some Reflections on Designing Construction Kits for Kids. In *Proceedings of the 2005 Conference on Interaction Design and Children*. ACM, Boulder Colorado, 117–122. doi:10.1145/1109540.1109556

[21] Kristin Williams, Jessica Hammer, and Scott Hudson. 2023. The IoT Codex: A Book of Programmable Stickers for Authoring and Composing Embedded Computing Applications. In *2023 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, Washington, DC, USA, 1–11. doi:10.1109/VL-HCC57772.2023.00009

[22] Qian Zhu, Zhuo Wang, Wei Zeng, Wai Tong, Weiyue Lin, and Xiaojuan Ma. 2024. Make Interaction Situated: Designing User Acceptable Interaction for Situated Visualization in Public Environments. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–21. arXiv:2402.14251 [cs.HC]. doi:10.1145/3613904.3642049