MicroCode: live portable programming for children with tangible robotics

Lorraine Underwood
Lancaster University
Lancaster, UK

Peli de Halleux
Microsoft
USA

Joe Finney
Lancaster University
Lancaster, UK

Thomas Ball
Microsoft
USA

Elisa Rubegni
Lancaster University
Lancaster, UK

Steve Hodges
Lancaster University
Lancaster, UK

ABSTRACT

The recently introduced MicroCode portable programmer enables young children and educators to program the BBC micro:bit without the need for a computer or the Internet. It includes a tile-based programming language for the BBC micro:bit and allows users to code the micro:bit "on to go" when slotted into a battery-powered shield that provides a color display and extra input mechanisms. MicroCode utilises the radio functionality of the micro:bit to send commands to and receive events from various micro:bit robot cars, extending its reach to allow for more interactive learning opportunities.

CCS CONCEPTS

• **Computer systems organization** → *Sensors and actuators*; • **Human-centered computing** → **User interface programming**; **Graphical user interfaces**.

KEYWORDS

micro:bit, MicroCode, programming, live, portable, robotics

ACM Reference Format:
Lorraine Underwood, Peli de Halleux, Joe Finney, Thomas Ball, Elisa Rubegni, and Steve Hodges. 2024. MicroCode: live portable programming for children with tangible robotics . In *Proceedings of The 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24 Adjunct) (UIST '24).* ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION AND RELATED WORKS

Physical computing devices are re-programmable computing systems that cut across hardware and software that can interact with their physical environment [12]. The BBC micro:bit is one such device that has grown in popularity in recent years [1]. The micro:bit has many advantages - it is small, portable, has embedded sensors, can be battery-powered and coded in multiple programming languages including block and text-based languages.

UIST '24, October 13 – 16, 2024, Pittsburgh, USA

ACM ISBN XXX-X-XXXX-XXXX-X/XX/XX…$XX.XX
https://doi.org/XXXXXXX.XXXXXXX

One disadvantage of many physical computing devices, including the micro:bit, is they need a host computer to program them. The list of these "computers" has expanded: a tablet or even a mobile phone can be used to program many of these devices.

The need for a host computer makes many physical computing devices and robotics inaccessible to multiple communities around the world. Notably those without reliable Internet, electricity or resources such as computers, tablets or mobile phones.Using two devices, a computer and a physical computing device can be too much for very young children to manage, as well as an educator with large class sizes.

When physical computing devices do not require a connection to a host computer, we call this portable programming. The Bee-Bot mobile robot is an example of this [6], a standalone device with buttons on its back. Action Box is another similar robot with buttons [11]. In these cases, there is no visual reminder of the code entered: what buttons were pressed and in what order. While using physical computing with pre-reading children could successfully introduce them to computational thinking [10] [9] just a few of them are available. Bee-Bot is also an example of a physical computing device that is accessible to much younger children as there are no words on it; thus children can code it without knowing how to read. According to Vee [16] learning to code is comparable to and just as essential as learning to read and write. The concept of "Coding as literacy" [2] recognizes the importance of digital literacy as a lifelong learning skill and an inclusion factor for a wealthy digital societies. [13]

Lego Mindstorms are an example of portable programming with a screen. Its user interface allows the creation of very simple programs that contain a sequence of actions, including looping.

2 MICROCODE

We introduce MicroCode, a new portable programmer that enables coding of the micro:bit and coding it as a remote to a variety of micro:bit robots.

The MicroCode software and hardware provide a live, portable and visual programming environment. The micro:bit can be plugged into a portable battery powered arcade shield. These shields include a D-pad and extra buttons to program the micro:bit using MicroCode, see Figure 1.

At first glance there are no words in the MicroCode coding screen, until the user moves over the tiles using the D-pad. The icons which children have referred to as "emojis" are fun and easy to recognise for children who are young and/or have low literacy

Figure 1: The micro:bit inside a portable shield running MicroCode

skills. The tiles are structured in a manner many children are used to when learning to read: Left to Right, Top to Bottom.

The program runs on the micro:bit in the shield. The children can see the program on the screen, and experience it running on the micro:bit at the same time. Any change made on the screen is immediately applied to micro:bit. This is known as live programming and is known to be beneficial for novices [7]. Students feel in control, they can directly see the effect of their changes. This can, in turn, improve the self-direction of students [3], increase their engagement [14] and show them how to apply programming concepts [15]. Like physical computing, live programming is based on the constructivist framework [8] supporting discovery [5] and active learning [17].

3 MICROCODE AND ROBOTS OVERVIEW

A new recent addition to MicroCode is the ability to code a variety of robots using the arcade shield and a second micro:bit in a robot, see Figure 2.

There is no standard for how companies control their robots using the micro:bit. For example company A may use pin 16 for the left motor. Company B may use pin 20. To ensure both robots act in the same way, the user must select what robot they have and download a hex file to their robot.

The micro:bit in the arcade shield sends commands over the radio to the micro:bit in the robot. Figure 3 shows some basic commands a child can create to turn the micro:bit into a remote for the robot car.

MicroCode includes line following and distance sensing tiles. The sensors on the micro:bit robot send data back to the shield.

4 CONCLUSION AND FUTURE WORK

MicroCode allows children of all ages and educators to program the micro:bit and micro:bit robots in a live and portable way. We hope

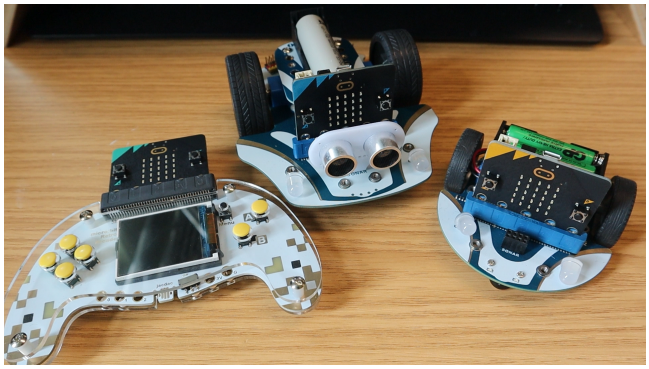

Figure 2: Different robots that can be remote controlled using MicroCode

Figure 3: MicroCode blocks to remote control robots, their distance sensors and line following sensors

to evaluate MicroCode with the communities it aims to support. In the future we will run a study in which pre-literacy children will be using MicroCode to program a robot in order to understand the opportunities and limitations of this approach to coding literacy in this age group. Future developments of the MicroCode project are ongoing. An exciting development is using the jacdac [4] port on many of the arcade shields to add extra accessories for the students to be creative, for example more buttons, multi-coloured LEDs and more sensors.

REFERENCES

[1] Jonny Austin, Howard Baker, Thomas Ball, James Devine, Joe Finney, Peli De Halleux, Steve Hodges, Michał Moskal, and Gareth Stockdale. 2020. The BBC micro:bit- from the UK to the world. *Commun. ACM* 63, 3 (2 2020), 62–69. https://doi.org/10.1145/3368856

[2] Marina Umaschi Bers. 2019. Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education* 6, 4 (2019), 499–528.

[3] Naomi R Boyer, Sara Langevin, and Alessio Gaspar. 2008. Self direction & constructivism in programming education. In *Proceedings of the 9th ACM SIGITE Conference on Information Technology Education (SIGITE '08)*. Association for Computing Machinery, New York, NY, USA, 89–94. https://doi.org/10.1145/1414558.1414585

[4] James Devine, Michal Moskal, Peli de Halleux, Thomas Ball, Steve Hodges, Gabriele D'Amone, David Gakure, Joe Finney, Lorraine Underwood, Kobi Hartley, Paul Kos, and Matt Oppenheim. 2022. Plug-and-Play Physical Computing with Jacdac. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 3 (9 2022). https://doi.org/10.1145/3550317

[5] Alessio Gaspar and Sarah Langevin-Gaspar. 2007. Active learning in introductory programming courses through Student-led "live coding" and test-driven pair programming. 1 (6 2007).

[6] Yen Air Caballero González and Ana García-Valcárcel Muñoz-Repiso. 2018. A robotics-based approach to foster programming skills and computational thinking. In *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*. ACM, New York, NY, USA, 41–45. https://doi.org/10.1145/3284179.3284188

[7] Christopher D Hundhausen and Jonathan L Brown. 2007. What You See Is What You Code: A "live" algorithm development and visualization environment for novice learners. *J. Vis. Lang. Comput.* 18, 1 (2 2007), 22–47. https://doi.org/10.1016/j.jvlc.2006.03.002

[8] Ilya Levin, Dina Tsybulsky, Ilya Levin, and Dina Tsybulsky. 2017. The Constructionist Learning Approach in the Digital Age. *Creative Education* 08, 15 (12 2017), 2463–2475. https://doi.org/10.4236/ce.2017.815169

[9] Larissa Meyer-Baron, Ramona Stieger, Bettina Waldvogel, Nicole Schumann, Cornelia Ruedisueli, and Franziska Mayr. 2022. Observing computational thinking skills of kindergarten children. In *Proceedings of the 17th Workshop in Primary and Secondary Computing Education*. 1–4.

[10] Ana Francisca Monteiro, Maribel Miranda-Pinto, and António José Osório. 2021. Coding as literacy in preschool: A case study. *Education Sciences* 11, 5 (2021), 198.

[11] Radia Perlman. 2004. Using Computer Technology to Provide a Creative Learning Environment for Preschool Children. (6 2004).

[12] Mareen Przybylla and Ralf Romeike. 2018. Empowering learners with tools in CS education: Physical computing in secondary schools. *it - Information Technology* 60, 2 (2018), 91–101. https://doi.org/10.1515/itit-2017-0032

[13] Luis González Rodero. 2023. THE EUROPEAN DIGITAL EDUCATION PLAN FOR THE DEVELOPMENT OF LIFELONG LEARNING. *The value of the difference and lifelong learning in the contemporary pedagogy* (2023), 113.

[14] Madhav Sharma, David Biros, Surya Ayyalasomayajula, and Nikunj Dalal. 2020. Teaching Programming to the Post-Millennial Generation: Pedagogic Considerations for an IS Course. *Journal of Information Systems Education* 31 (6 2020), 96–105.

[15] Sheng-Rong Tan, Yu-Tzu Lin, and Jia-Sin Liou. 2016. Teaching by demonstration: programming instruction by using live-coding videos. In *EdMedia+ Innovate Learning*. 1294–1298.

[16] Annette Vee. 2017. *Coding literacy: How computer programming is changing writing*. Mit Press.

[17] Nesra Yannier, Scott E Hudson, and Kenneth R Koedinger. 2020. Active learning is about more than hands-on: A mixed-reality AI system to support STEM education. *International Journal of Artificial Intelligence in Education* 30 (2020), 74–96.