Where's the bug?

Helping students find errors in physical computing

Michael Schneider
Michael.J.Schneider@colorado.edu
University of Colorado, Boulder
USA, Boulder, Colorado

ABSTRACT

Popular platforms for teaching physical computing like the LilyPad Arduino and Adafruit Circuit Playground have simplified programming and wiring, enabling students to quickly engineer physical computing projects. But enabling students to rapidly design and build is a double-edged sword: Students can create functioning prototypes without fully understanding the underlying principles. With limited knowledge and experience, students struggle to locate and fix bugs, or errors, in their projects. Absent appropriate debugging tools, students rely on their instructor for locating errors, or worse, turn toward destructive tactics such as tearing apart and rebuilding their project, hoping the bug fixes itself. Students need tools targeted to their ability that scaffold debugging and help them locate bugs in the mixed hardware/software environment of physical computing. I developed Circuit Check to scaffold the debugging process for students. It enables students to observe real-time sensor data and test hardware components through a novel adaptation of the traditional breakpoint for physical computing.

CCS CONCEPTS

• **Applied computing** → **Education**.

KEYWORDS

debugging, debugging tool, physical computing, embedded systems, microcontroller, Arduino, computer science education

ACM Reference Format:
Michael Schneider. 2022. Where's the bug?: Helping students find errors in physical computing. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2022), March 3–5, 2022, Providence, RI, USA.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3478432.3499059

1 INTRODUCTION

In physical computing, which combines sensors and actuators with a programmable microcontroller, locating bugs can be quite difficult because they can occur in the code, in the circuit, or within the interactions of the two [1]. Unfortunately, existing debugging tools for physical computing fail to provide sufficient support for students [1], by providing system information at a level appropriate to the student's knowledge and skill level.

SIGCSE 2022, March 3–5, 2022, Providence, RI, USA

ACM ISBN 978-1-4503-9071-2/22/03.
https://doi.org/10.1145/3478432.3499059

2 RELATED WORKS

Current debugging tools for physical computing lie at two extremes of a spectrum [2]. At one end are print statements, which support some level of debugging but students struggle with interpreting sensor readings via the Serial Monitor [1], or introduce new bugs as they modify their code to include debugging print statements. At the other extreme are advanced debugging tools such as hardware debugging probes, which are paired with professional Integrated Development Environments, such as Microchip Studio IDE. In interviews with STEM instructors, I found not only are these tools inappropriate for middle school students, the teachers feared that the level of detail provided by the IDE's could overwhelm their students. Our teachers wanted a debugging tool that would enable students to observe their system's behavior, yet limited in complexity to ensure a manageable learning curve for adoption.

3 OVERVIEW / RESULTS

Through Circuit Check's web interface, students can isolate and test their hardware and evaluate live sensor data to better comprehend the behavior of their system, enabling them to test both hardware- and software- focused hypotheses. I evaluated Circuit Check's usability through two one-week long summer camp programs with middle school students. Students designed and crafted their own e-textile projects and used Circuit Check to observe live sensor data and locate software/hardware bugs.

4 CONTRIBUTIONS

Initial observations found Circuit Check facilitated active discussion between the summer camp instructor, students, and myself around their observations of system behavior. We plan to conduct additional studies to observe Circuit Check's impact on classroom discourse around debugging.

ACKNOWLEDGMENTS

This work was supported by grant #1742081 from the National Science Foundation.

REFERENCES

[1] 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.* 3485–3497.
[2] Jan Dolinay, Petr Dostálek, and Vladimír Vašek. 2021. Advanced debugger for Arduino. *International Journal of Advanced Computer Science and Applications* (2021).