WireOn: Supporting Remote Collaboration for Embedded System Development

Yan Chen
University of Michigan, Ann Arbor
yanchenm@umich.edu

Jasmine Jones
Berea College, Berea
jonesj2@berea.edu

Yaxing Yao
Carnegie Mellon University
yaxingyao@cmu.edu

ABSTRACT

The rise of the Maker movement has led to a growing number of developers who prototype and program embedded systems. When programming, these developers often rely on support from various resources—including other developers. However, other developers may not always be available to provide support in person, and existing technologies for online help, such as voice chat or Q&A forms, face the fundamental limitation of inspecting and manipulating developers' circuit boards. As a result, remote helpers can only provide suggestions or guidance, rather than contributing via physical changes made to the devices. And only end-user developers have the ability to carry out the planned tasks. In this paper, we demonstrate WireOn, a programming support research prototype that allows remote helpers to directly perform tasks on end-user developers' circuit board by teleoperating a robot arm. The helpers can control the robot arm via a web user interface to perform simple tasks such as pick-and-place the electronic components, visually inspect the physical artifacts in real time, and also review the code that the end-user sent over to them. The new system has the potential to enable more efficient remote collaboration on embedded system development. (https://youtu.be/uggyxHAlLDQ)

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

on-demand support; programming collaboration; crowdsourcing

INTRODUCTION

The rise of the Maker movement has led to a growing number of developers who prototype and program embedded systems (e.g., Arduino). When programming, developers often rely on the support from others [1, 4]. They might ask questions related to software programming, such as "what's the code syntax to set the pin?" which can typically be answered using Integrated Development Environment (IDE) auto-complete or search engines. However, the types of questions developers choose to ask are influenced by the available support tools. For queries like "is this sensor on my breadboard reading the correct data from the environment?" is unlikely to produce useful results in a search engine.

However, many teams have shifted to online remote collaboration, from professional development teams to programming classes to computing-related workspaces (e.g., makerspaces). While many online collaboration tools for supporting program development has been introduced by the HCI community [3, 9, 10], most of them have designed with software development in mind. Embedded system developers must collaborate remotely to work on their projects, which can be problematic. Specifically, there are two challenges. **First**, capturing the context of physical devices can be daunting. Typically, development history such as prior wire connections and hardware states is not automatically captured. This makes communication inefficient, as helpers might ask requesters to try previously attempted solutions or request historical information (e.g., prior wire connections). **Second**, remote helpers lack access to inspect and manipulate developers' physical devices, which limits them to providing only suggestions or guidance rather than contributing via physical changes made to the devices. Thus, in spite of the collaboration, only end-user developers have the ability to carry out the planned tasks.

To address these challenges, existing tools such as Bifröst allow developers to more efficiently debug embedded systems [8]; likewise, Heimdall allows instructors to remotely inspect, measure, and rewire students' circuits [6]. However, both tools have drawbacks: the use of Bifröst is limited to specific tasks, and developers must rely on their own capacity to complete their tasks. Heimdall requires students to build their circuits on a specialized workbench, limiting the size of students' projects and prohibiting them from working in parallel when seeking help [6]. Therefore, this prior work leads to the following research question: **How can we facilitate effective remote collaboration for embedded system development?**

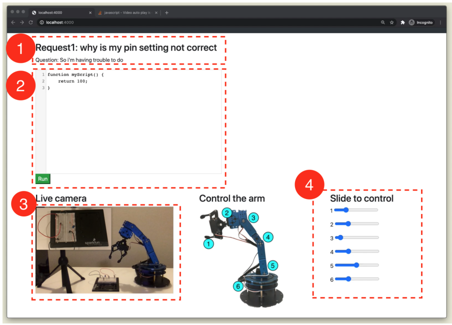

Figure 1: WireOn web UI. It includes four parts: 1) a natual language task description; 2) a code editor that includes the task relevant code; 3) two fix-position live stream camera views; 4) a 6-slider control panel that corresponds to the 6 servos of the robot arm

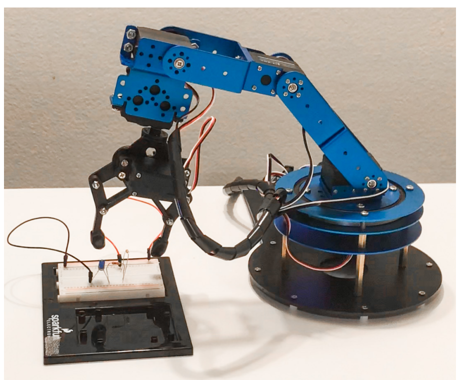

Figure 2: A photo of WireOn robot arm performing a wiring task.

WIREON

As robots increasingly enter our everyday life, we envision a future in which home robots are more affordable, and ubiquitous in our daily encountered environments. With this vision in mind, this work demonstrates our initial effort to address this question by designing WireOn, a web-based, tele-operated DIY robot arm that allows remote collaborators (helper) to perform inspecting, wiring, and even coding tasks upon end-user developer's (requester) requests. WireOn consists of two parts: a robot arm, and a web user interface.

Robot Arm

To enable physical manipulation, WireOn uses an off-the-shelf, with 6 degree-of-freedom (DoF), robot arm (USD $59.00) [5] that can perform simple pick-and-place tasks. The six servos are connected to an Arduino Uno microcontroller which is controlled by WireOn's web UI.

Web User Interface

To enable remote control, we design a web UI that consists of four parts (Fig. 1): a natual language task description that specifies end-user developer's request; a code editor that includes the task relevant code; two fix-position live stream camera views, including one top-down view and one side view; and a 6-slider control panel that corresponds to the 6 servos of the robot arm.

We used two Logitech C270 cameras which capture 1280x720 resolution in live-viewing. We adopted a fixed position camera view rather than a moving and controllable camera view, because prior studies have shown that users may have various privacy concerns of moving and controllable cameras, such as drones [11, 12] and teleoperated robot arm [2]. We think a fixed camera view can minimize users' privacy concerns while not sacrificing the utility of the arm.

DEMONSTRATION

When conference attendees arrive at our demonstration, they will be able to teleoperate the robot arm via the web UI to perform a predefined set of simple embedded system development task (e.g., wiring). Each user connection time will be capped to 5 minutes if there are others in line.

LIMITATIONS AND FUTURE WORK

Through our initial attempt of designing WireOn, we noticed a couple of limitations. In this section, we briefly describe them and propose a few future work along the lines. The biggest limitation of WireOn is its controllability. With fully manual and naive UI, we realized that non-expert users might find difficult to perform a tasks. Prior work on designing a teleoperated robot UI for non-expert has found that a point-and-click UI, which proposes grasp poses based on local 3D surface geometry, helped users complete more tasks than existing approaches [7]. However, this approach requires scene information which might raise users' privacy concerns. Future work could explore novel ways to guide helpers to learn how to control the arm to perform relevant tasks.

The potential tasks that the requesters would want to hand-off to remote helpers are also unexplored. The success of paid online crowdsourcing marketplaces (e.g., Upwork) and Q&A sites has made it desired and possible for software developers to request on-demand programming assistance from remote experts in nearly real time [4]. Future work could explore what types of requests that embedded system developers want to make if they have a WireOn available.

ACKNOWLEDGMENTS

We thank Steve Oney, Anhong Guo, and Liang He for their feedback on the motivation and limitation.

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 *CHI*.

[2] Daniel J. Butler, Justin Huang, Franziska Roesner, and Maya Cakmak. 2015. The Privacy-Utility Tradeoff for Remotely Teleoperated Robots. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction* (Portland, Oregon, USA) *(HRI '15)*. Association for Computing Machinery, New York, NY, USA, 27–34.

[3] Yan Chen, Sang Won Lee, Yin Xie, YiWei Yang, Walter S Lasecki, and Steve Oney. 2017. Codeon: On-demand software development assistance. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM.

[4] Yan Chen, Steve Oney, and Walter S Lasecki. 2016. Towards providing on-demand expert support for software developers. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. ACM.

[5] Hiwonder Inc. 2020. . https://www.hiwonder.hk/collections/robotic-arm Accessed: July, 2020.

[6] Mitchell Karchemsky, JD Zamfirescu-Pereira, Kuan-Ju Wu, François Guimbretière, and Bjoern Hartmann. 2019. Heimdall: A Remotely Controlled Inspection Workbench For Debugging Microcontroller Projects. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[7] David Kent, Carl Saldanha, and Sonia Chernova. 2020. Leveraging depth data in remote robot teleoperation interfaces for general object manipulation. *The International Journal of Robotics Research* 39, 1 (2020), 39–53.

[8] 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*. 299–310.

[9] Steve Oney, Christopher Brooks, and Paul Resnick. 2018. Creating guided code explanations with chat. codes. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–20.

[10] Soya Park, Amy X Zhang, and David R Karger. 2018. Post-literate Programming: Linking Discussion and Code in Software Development Teams. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings*.

[11] Yang Wang, Huichuan Xia, Yaxing Yao, and Yun Huang. 2016. Flying eyes and hidden controllers: A qualitative study of people's privacy perceptions of civilian drones in the US. *Proceedings on Privacy Enhancing Technologies* 2016, 3 (2016).

[12] Yaxing Yao, Huichuan Xia, Yun Huang, and Yang Wang. 2017. Privacy Mechanisms for Drones: Perceptions of Drone Controllers and Bystanders. 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, 6777–6788.