RHL-Butterfly: A Scalable IoT-Based Breadboard Prototype for Embedded Systems Laboratories

Matthew Guo
Department of Electrical & Computer Engineering
University of Washington
Seattle, USA
mguo5@uw.edu

Rania Hussein
Department of Electrical & Computer Engineering
University of Washington
Seattle, USA
rhussein@uw.edu

Pablo Orduña
LabsLand
Saint Louis, USA
pablo@labsland.com

***Abstract*—This Research to Practice Work-In-Progress paper presents a virtualized breadboard solution for FPGAs and ARM microcontrollers in remote laboratories. The circumstances that rose amidst the COVID-19 pandemic demonstrated the vulnerability of current engineering education practices, particularly in dealing with hardware resources. Pivoting to the emergency online instruction presented challenges to the traditional practices in delivering hands-on engineering labs, which necessitated a solution that handles hardware prototyping without compromising creativity and instruction. One vital aspect of the embedded systems learning experience is ensuring students and faculty members alike have opportunities to learn and build custom prototyping circuits that interact with microprocessors on breadboards. In this paper, we build on the prior work that our group implemented on using virtualization to interface a virtual breadboard with physical hardware through web applications. Our previous work was limited to interfacing with one particular kind of hardware, designed to explore the capabilities of fundamental transducers and actuators that interface with hardware I/O pins. In hardware engineering practice, however, designers are not constrained by a single microprocessor selection to control their system and designs and are not limited by the type of transducers and actuators that provide the external circuit functionality. This paper presents a solution by scaling the existing virtual breadboard research to support FPGAs and ARM microcontrollers and intermediate logic gate integrated circuits for practical use in engineering curriculums. Providing this increased selection of supporting hardware helps facilitate student learning and simulates hardware development in an industrial setting. Due to the rising popularity of FPGAs and ARM microcontrollers in industry and in education, we expect that our solution will serve a larger audience through this broader selection of supported hardware. Our solution virtualizes the breadboard prototyping experience without sacrificing the nature of real-time embedded systems by taking the user prototyped inputs and outputs and directly programming the functionality of the surrounding system to physical hardware. This balance between a virtualized interface and physical hardware implementation preserves a hardware curriculum embedded systems engineering education and brings a promising solution to expand the scalability and accessibility of engineering labs.**

Keywords—virtualization, remote laboratories, engineering education, educational technology, scalable IoT solutions.

I. Introduction

The COVID-19 pandemic necessitates a rethinking of educational technologies, popularizing virtual solutions for remote and simulated hardware laboratory implementations [1][2]. The Remote Hub Lab (RHLab) [3] is one such remote laboratory solution created amid the pandemic. With support for remote access to FPGAs, ARM microcontrollers, Arduino robots, and more, the RHLab creates an opportunity to redefine embedded systems and hardware engineering education: using cloud computing and the Internet of Things (IoT), users have opportunities to continue prototyping hardware and experimenting with designs via remote access, providing continued support for guided or detailed instruction that is accessible anywhere with internet connectivity. The current virtualized breadboard implementation by the RHLab [4] provides an innovative way to allow students to create external hardware for their microprocessors, but is limited rudimentary actuators and transducers. Limitations on breadboard solutions can deter potential creativity for original embedded system designs. As such, the RHL-Butterfly aims to scale the existing RHLab implementation by supporting larger functionality for the Intel DE1-SoC with the integration of additional logic gate ICs, and by adding support to common ARM microcontrollers, potentially allowing students to enhance their skills in circuit design and digital logic.

Recent research on virtualizing hardware practices has favored simulated and remote laboratory instruction, due to benefits in financial savings, increased safety for potential dangerous electrical experimentations, and improved laboratory hardware accessibility to handicapped learners [5]. This shows that remote laboratories are here to stay, even in a post-pandemic era. A recent study conducted by Hussein and Wilson [6] showed promising results that favor a remotely accessible laboratory learning environment. Using Bloom's taxonomy, the study compared students' performance in a Design of Digital Circuits course for two different academic quarters. In one academic quarter, the course used traditional FPGA hardware, while in the other academic quarter, the course used a remote FPGA laboratory. The study showed comparable results for both course options in every category except *Analyze*, to which the remote instruction using the remote FPGA laboratory produced favorable results over the traditional offering. With a virtual

The RHL-Butterfly project is funded by Intel Corporation.

breadboard prototyping environment, students can experiment with their designs without the added distraction of wondering whether a breadboard component has a silicon level imperfection, or contains an unforeseen internal short circuit that electrically damages the rest of their circuitry. This way, students can spend more time understanding the practical and theoretical concepts of course curriculums. As such, these studies showing the advantages of remote laboratories over using physical hardware equipment provide justification and motivation for the continued expansion of hardware virtualization. We expect that with even greater microcontroller support and microchips, virtualized laboratories can provide students with more options for their designs, increasing their performance to learning objectives.

One aspect of research for this IoT Virtual Breadboard for Remote Embedded Laboratories is its novelty in existing literature. In many aspects, this project mirrors the Java Digital Breadboard Simulator, developed by Glass [7]. Featuring an open-source library, this simulator is a desktop program designed to run on a Windows operating system. Where this project differs, however, from the Java Digital Breadboard Simulator, is in the level of simulation. The Digital Breadboard Simulator is designed for circuit-level simulation, logic level simulation, and functional level simulation, and acknowledges that one key disadvantage in a computer simulation is its extensive computation to behave as a real system would, thus slowing down the real-time process. To combat this, the breadboard by Glass simplified assumptions in the simulation program. Our IoT breadboard solution, while may not have the ubiquity in the models that the simulator possesses, contains one distinct advantage: by utilizing real-world hardware as the output to the user-defined breadboard input, this project preserves the real-world aspect, thus making this project less of a digital simulation as virtualizing real hardware. When users "submit" their virtual breadboard configuration, the web application client scans through the design and sends it to a JSON package, containing information regarding each component, their component logic states, and intended logic level output. The information is then sent to the physical microcontroller, which parses through the breadboard package information to determine the appropriate hardware input configuration and necessary output hardware response. That way, the idiosyncrasies of real-world embedded system microprocessors, microprocessor computation, and their peripherals are preserved, even when virtual breadboards are used to interface with the hardware.

In this paper, we present our solution for a scalable virtualized remote breadboard, aimed to address the shortcomings analyzed in previous remote solutions, and how this solution contributes to the practical use of embedded systems engineering curriculum. Using a virtual breadboard that communicates with physical hardware, we preserve the real-time functionality of embedded systems, allowing students to design around engineering requirements and constraints focusing on real hardware. Incorporating our solution into a platform accessible for everyone, we also aim to focus on the open-source nature of our solution to truly bring equitable remote laboratory access to all users around the world.

II. Methods

The RHL-Butterfly brings communication between a virtualized breadboard with physical hardware, taking into account the physical breadboard's electrical characteristics. Our virtual breadboard solution needs to be both electrically accurate in a physical breadboard, and have fast communication to the microcontroller hardware to justify the product design. These metrics define the user-experience for remote laboratories; noticeable differences with real-world hardware can provide distractions to the user experimenting their creations, and may inhibit the effectiveness of remote laboratories and their abilities to learn.

A. Core Components

This project is divided into three distinct components, categorized as subtasks, to mimic a sprint development cycle. The microprocessor categorizes the first component. The project uses an STM32F04 DISCOVERY Board, chosen for its wide range of 32-bit applications, a fully integrated STMicroelectronics development environment, and its ubiquity in usage for embedded systems capstone level courses. While this proof-of-concept design specifically uses the STM32F04, the project is not limited to this particular hardware, with hardware support reaching many ARM microcontrollers and Intel FPGAs, and as such, the microprocessor will be generically referred to as "microcontroller" for the remainder of the paper. For the second component, a Raspberry Pi 4 WiFi Edition is used to interface between the microcontroller and a client front-end website, using UART communication between the microcontroller and the Raspberry Pi, and HTTP between the Raspberry Pi and the client front-end. The third component is the client front-end itself, constructed from the python Flask framework, containing the user interactive, virtualized breadboard GUI, along with a live camera feed of the microcontroller to demonstrate real-time hardware response.

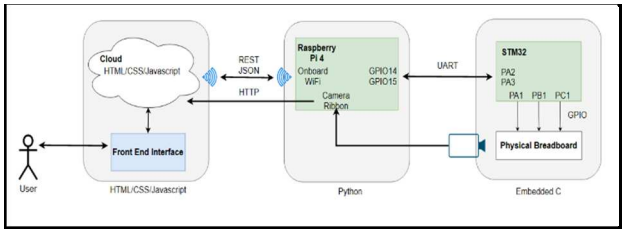

Fig. 1. Top-level block diagram of the system.

B. Microcontroller and Raspberry Pi

The Raspberry Pi 4 acts as a translation between the frontend GUI and the serial communication needed for the microcontroller to understand the virtualized breadboard. The microcontroller recognizes incoming data through its UART interrupt. Inside the microcontroller UART interrupt handler, the microcontroller parses the received data by scanning which digital GPIO needs to change its state and adjusts the GPIO logic level accordingly. Communication between the Raspberry Pi 4 and the microcontroller uses a handshake protocol for error-checking. In a rare instance in which the microcontroller receives misaligned data, the firmware error checking method will then send back to the Raspberry Pi 4 "error", informing the Raspberry Pi 4 to send the message again. Upon successful

completion of the microcontroller GPIO logic level change, the microcontroller will send back a "success" message, indicating that the user breadboard action has been successfully interpreted and completed.

C. Client Front-end

RHL-Butterfly's breadboard design stems from the embedded system's abstraction that digital inputs to microprocessors and microcontrollers focus on two main elements: the change in logic level, and the time at which the logic level has been changed. Both of these can be simulated through asynchronous virtual inputs on the breadboard GUI. This GUI is implemented using the breadboard visir open-source library [8][9], allowing users to freely draw different wires around the breadboard. A 50-pin connector is implemented on this GUI to simulate an input onto the 50-pin connector for the microcontroller. This way, users can place one end of a wire on their custom-created breadboard design, and the other end of the wire onto a GPIO pin of their choosing.

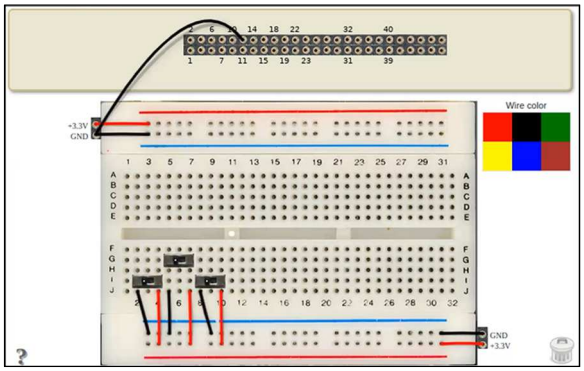

Fig. 2. The virtualized breadboard configurator for Anonymous-Butterfly.

Taking advantage of the mastery of the embedded system for asynchronous real-world detection, the breadboard GUI continuously updates its user-input state after every mouse click. When prototyping on hardware breadboards, a common workflow would be that the users create their circuit and verify it before initializing power through their system for electrical testing. To preserve this prototyping workflow, users of this breadboard GUI would submit their designs to the physical hardware by pressing the "submit" button on the webpage, indicating that step of "powering" their design. This updates the system one last time before scanning through the user-specified breadboard design and packaging the system in terms of the change in detected logic level for the output. Due to the numerous amounts of different permutations the users have control over with their designs, the frontend client looks for changes in previously submitted designs, rather than reconstructing the circuit each time, before abstracting their designs to a JSON protocol. This JSON packet is then passed to the Raspberry Pi 4 via an HTTP POST method. As such, by the time the Raspberry Pi 4 has the data, the python processing does not know, nor needs to know, what the user has implemented on their breadboard.

III. Preliminary Results

The existing RHLab breadboard GUI solution brings virtualization to breadboard experimentation for DE1-SoC FPGAs. In a previous study by Li *et al.* [4], this virtualized breadboard GUI was deployed to all students taking the Design of Digital Circuits class at the University of Washington. Students were asked to participate in an anonymous online survey indicating the usability and practicality of the virtualized breadboard based on a 5-point Likert scale. The study scored results favoring the virtualized breadboard, leading to that previous breadboard iteration as a basis for this improved version. With this previous research supporting virtual breadboard solutions for engineering curriculum, our breadboard GUI provides confidence that this scaled virtual solution brings similar effectiveness to laboratory learning, and to engineering curriculum beyond Design of Digital Circuits.

A. Breadboard Features

RHL-Butterfly's virtual breadboard not only scales the previous edition through added support to ARM microcontrollers, but also expands its functionality for Intel DE1-SoC FPGAs to include component support for 3-pin, 2-stage SPDT toggle switches and rudimentary 14-pin logic gates including NOT gates, AND gates, and OR gates, and breadboard error checking over breadboard electrical node characteristics. Users can draw wires anywhere from the virtual breadboard to the 50-pin connector mirroring pinout connections to the microcontroller.

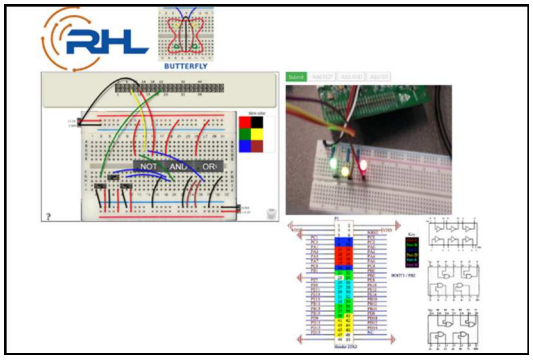

Fig. 3. Front-end virtualized breadboard graphic user interface.

RHL-Butterfly allows real-time communication between a virtualized breadboard interface to real-world microcontroller hardware and provides a camera visualization of the hardware response. With support for common logic gates, users can construct more external digital circuits than the project's predecessor, currently deployed at the RHLab and used for the Design of Digital Circuits course at the University of Washington.

B. Breadboard Usability

Our implementation of the breadboard virtualization delivers a promising proof-of-concept design highlighting scalability for additional virtualized integrated circuit components and scalability for support with additional microprocessor architectures. Despite its promise, it is still

considered a prototype and is not yet deployed at the RHLab, thereby providing limited consumer testing on its usability and practicality for virtualized implementation of embedded systems curriculum. Instead, we draw insight into the system's potential effectiveness based on studies with previous versions and similar designs in literature, rather than the use cases of this particular project. As we continue to expand on the functionality of this current proof-of-concept, we can gain a better understanding of the effectiveness of this design rather than relying on conclusions drawn from previous studies.

IV. Future Work

To provide users with visual feedback, the response from the physical hardware based on user input through the breadboard is captured through a live camera feed and displayed virtually on a web application alongside the breadboard solution, creating a fully remote laboratory experience. Because the virtual breadboard interacts directly with physical hardware, delays in the live stream must be minimized to mimic that real-time experience. There was a noticeable delay with the camera feed during the development phase of this project. This delay may cause frustrations with breadboard prototyping, especially when synchronous microcontroller peripherals are used. Research by Rodríguez-Gil [10] presents a solution to the limitations of video live streaming while keeping the software open-source, called the WILSP platform, becoming a backbone in remote laboratories and in the RHLab. This solution, using the Redis data structure store, targets applications for interactive live streaming rather than backseat live streaming, which fits the intended goal of the IoT-based virtual breadboard solution. Integrating this technology will minimize the camera video feed latency of the hardware response.

The next iteration will also feature analog circuit devices, allowing users to take advantage of ADC and DAC peripherals on the microcontroller. By doing so, we can expand on the supported integrated circuits for the virtual breadboard with simulated support for operational amplifiers, BJTs, and other passive circuit elements. In addition, we aim to provide support for virtual oscilloscopes and other virtual laboratory debugging tools to help students enhance their design creativity. This provides opportunities for students to explore all applications that embedded system design has to offer.

V. Conclusion

With remote laboratories becoming increasingly popular and research supporting its implementation for student curriculums, RHL-Butterfly delivers one practical solution to breadboard experimentation for embedded systems and engineering education. Using remote laboratories in education has shown to be as effective, if not more so, as laboratories that use traditional hardware. This project brings the potential to further enhance the engineering curriculum that uses remote laboratories. In addition, this project continues to use physical hardware over a purely digital and simulated experience to allow educational awareness of real-world constraints to circuit design. Using a breadboard GUI, users can prototype and customize their embedded system solutions by integrating external electronics and circuitry into an microcontroller using IoT. The advantages of remote laboratories and the scalability of our virtualized breadboard provide potential viability and effectiveness for engineering education, even in a post-pandemic era.

References

[1] S. Ray and S. Srivastava, *"Virtualization of science education: a lesson from the COVID-19 pandemic."* J Proteins Proteom, vol. 11, 2020, pp. 77-80. https://doi.org/10.1007/s42485-020-00038-7.

[2] R. Mulya, K. Krismadinata, N. Jalinus, and H. Effendi, "*Practical Work of Digital Systems Course Based on Virtual Laboratory.*" International Journal of Online and Biomedical Engineering (iJOE), vol. 17, 2021, pp. 22-37. https://doi.org/10.3991/ijoe.v17i08.23359.

[3] "*The Remote Hub Lab.*" 2022. Available: https://rhlab.ece.uw.edu/

[4] S. Li, H. Wang, L. Rodriguez-Gil, P. Orduña, and R. Hussein, "*FPGA Meets Breadboard: Integrating a Virtual Breadboard with Real FPGA Boards for Remote Access in Digital Design Courses.*" in *Online Engineering and Society 4.0*, 2020, pp. 144-151. https://doi.org/10.1007/978-3-030-82529-4_15

[5] R. Heradio, L. de la Torre, D. Galan, F. J. Cabrerizo, E. Herrera-Viedma, and S. Dormido, "*Virtual and remote labs in education: a bibliometric analysis*." Computers & Education, vol. 98, 2016, pp. 14–38, doi: 10.1016/j.compedu.2016.03.010.

[6] R. Hussein and D. Wilson, "*Remote versus In-hand hardware laboratory in digital circuits courses."* American Society for Engineering Education ASEE conference, Electrical and Computer Engineering Division. 2021.

[7] N. Glass, "*Java Digital Breadboard Simulator: A simulator for educational electronics environment,*" 2002. Available: https://muchlas.ee.uad.ac.id/downloads/nicholas%20glass-jbreadboard.pdf

[8] I. Gustavsson, J. Zackrisson, L. Håkansson, I. Claesson, and T. Lagö, "*The VISIR project - an Open Source Software Initiative for Distributed Online Laboratories.*" REV Conference 2007. 2007.

[9] D. May, B. Reeves, M. Trudgen, A. Alweshah *"The remote laboratory VISIR – Introducing online laboratory equipment in electrical engineering classes".* 2020 IEEE Frontiers in Education Conference (FIE). IEEE, 2020.

[10] L. Rodríguez-Gil, J. García-Zubia, P. Orduña, and D. Lopez-de-Ipiña, "*An Open and Scalable Web-Based Interactive Live-Streaming architecture: The WILSP Platform."* in *IEEE Access*, vol. 5, 2017, pp. 9842-9856, doi: 10.1109/ACCESS.2017.2710328.