

Appliancizer: Transforming web pages into electronic gadgets

Jorge Garza

Devon J. Merrill

Steven Swanson

Department of Computer Science and Engineering
University of California San Diego, CA, USA
{jgarzagu, djmerril, swanson}@eng.ucsd.edu

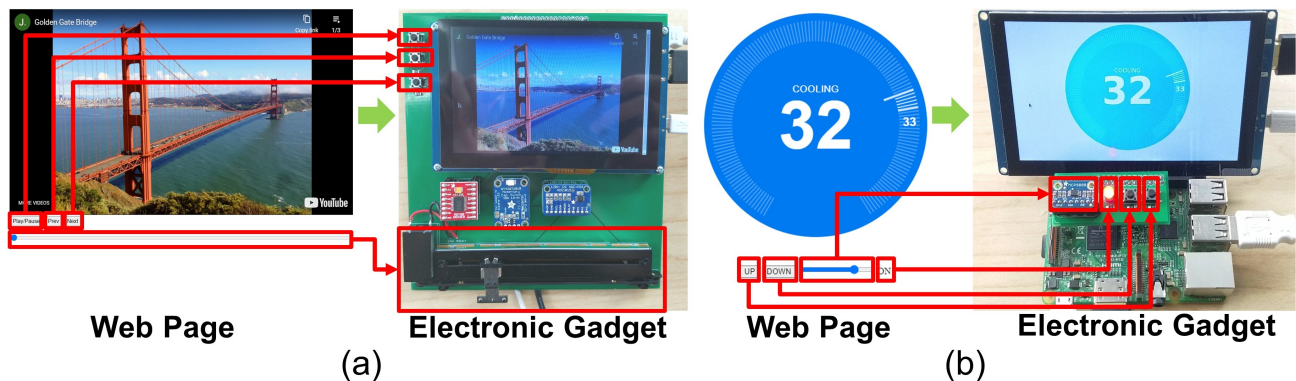


Figure 1. Appliancizer allows web pages to be transformed into electronic gadgets. At the left of a), we show a video player web page that provides a familiar on-screen interface for playing videos. At the left of b), a thermostat simulated on a web page with basic controls for adjusting the temperature. At the right of both web pages, the automatically generated electronic gadget running the same web application but with a full-tactile interface.

ABSTRACT

Making electronic gadgets that meet today's consumer standards has become a difficult task. Electronic gadgets are expected to have displays with visually appealing interfaces and, at the same time, be physically and screen-interactive, making the development process of these devices time-consuming and challenging. To address this problem, we have created Appliancizer, an online synthesis tool that can automatically generate sophisticated electronic devices from web pages. Appliancizer takes advantage of the similarities between software and physical interfaces to reduce development steps and allow the rapid development of electronic devices. By matching the interface of hardware components with the interface of graphical HTML elements found on web pages, our tool allows a designer to transform HTML elements from a digital to a tangible interface without changing the application source code. Finally, a modular design enables our tool to auto-

matically combine the circuit design and low-level hardware code of selected hardware components into a complete design. Attendees can interact with our online tool and produce manufacturable PCBs from web pages.

Author Keywords

Web applications, electronics design automation, digital fabrication, computer-aided design, HTML, PCB layout

CCS Concepts

•Human-centered computing → Human computer interaction (HCI); •Applied computing → Computer-aided design;

INTRODUCTION

Electronic devices play an essential role in our modern society. From the automation of many daily tasks, such as thermostats that automatically maintain a pleasant temperature in our house, to multimedia devices that are part of our everyday entertainment. The design of intuitive interfaces for these devices is important for their correct use. With the arrival of devices such as smartphones, the interacting expectation with devices has changed. It is now expected that devices can be interfaced with not only physically, but with a mix of both physical and a highly interactive digital display. At the same time, these devices are expected to be connected to the internet

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

UIST '20 Adjunct, October 20–23, 2020, Virtual Event, USA

© 2020 Copyright is held by the author/owner(s).

ACM ISBN 978-1-4503-7515-3/20/10.

<http://dx.doi.org/10.1145/3379350.3416158>

and integrated into ecosystems where they can communicate with other devices [4].

Web technologies such as HTML, CSS, and JavaScript have proven to be flexible for creating interactive interfaces and effective in communicating with the internet. With the myriad of available libraries, creating web pages that have visually compelling animations, reproduce multimedia videos and music, or transmit and receive data from the internet is possible with a few lines of code. These capabilities make web technologies ideal for their use in electronic devices. Tizen [10] and Firefox OS [8] are examples of architectures based on web technologies for smart electronic gadgets. However, the growing demand for smart devices also requires them to be fast and easy to build.

The conventional process for developing electronic gadgets at the board-level circuit consists of different tasks that require a high amount of time and expertise [6]. Each task requires different skills and knowledge in various fields: software, hardware, and mechanical. Designers, developers, and engineers at industries with the required expertise, have no problems with the usual process cycle. However, novices looking to create their own electronic gadgets find it challenging to meet the time and expertise required in the conventional process [7].

Previous attempts [3, 9, 2] to automate the conventional development process of embedded devices have just added high-level hardware specifications into existing programming languages requiring developers to learn new methodologies, add extra libraries or constrain, therefore increasing the time and skills necessary to build an electronic device.

APPLIANCIZER

Appliancizer is an online¹ synthesis tool for circuits that can automatically generate Printed Circuit Board (PCB) layout designs and hardware code from web pages. Compared to previous tools, Appliancizer allows synthesis without requiring adding any hardware specifications into the programming language. Instead, we propose a technique we call *binary interface mapping*, allowing a user to transform web based software applications into complete functional electronic devices without changing the application source code.

TRANSFORMING HTML ELEMENTS FROM A DIGITAL TO A TANGIBLE INTERFACE.

Appliancizer can transform a digital user interface into a mixed or fully tangible user interface. HTML elements can be moved off-screen to interact with the non-digital real world. Figure 2 shows an HTML button and a span element transformed into physical components with which a user can tangibly interact. The process of converting an interface is as follows. HTML elements are first dragged into a virtual PCB board area. A binary interface mapping allows matching the input and output similarities that exist between dragged HTML and similar hardware component. An HTML button, for example, matches tactile push buttons perfectly as they both produce an ON or OFF binary input state. The user then selects the desired hardware part that fits its design.

¹Appliancizer is available at <https://appliancizer.com>

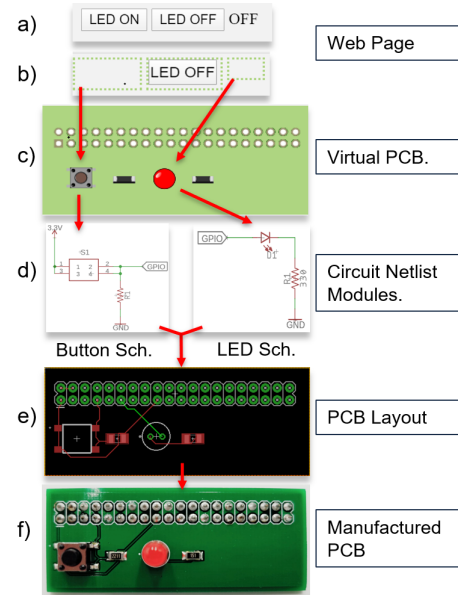


Figure 2. Digital to Tangible interface process: After HTML elements in a web page a), b) are dragged and dropped into the virtual PCB c). A complete circuit netlist is then generated from each separate hardware component circuit netlist module d). A PCB layout is then generated from the complete circuit netlist. f) Shows the final PCB after manufacture.

When ready for manufacturing, our tool automatically converts the virtual PCB to a fabricable PCB. For each project, a web application is also generated with the required low-level hardware code allowing its interaction with hardware components. A custom operating system image, provided by our tool, allows users to navigate and run the web application. Finally, at runtime, the selected HTML elements are replaced by *hardened* [5] HTML web components [1] that interface with the selected hardware components. Figure 1 shows two examples of more sophisticated electronic gadgets, a video player and a thermostat. Its PCBs were automatically created from a web page, and their hardened HTML elements allow physical interaction with the device's web application.

RAPID DEVELOPMENT OF ELECTRONIC GADGETS

By automating circuit design and low-level hardware code development, developing a new electronic gadget comes down to selecting the HTML elements to transform. Users can choose to leave some items inside the web page as "soft" elements and harden others. Figure 1 shows the electronic gadget (b) without a screen display component on the board and left as a "soft" element, allowing the PCB to be placed at the back or at a different angle from the screen.

CONCLUSION

In this work, we presented Appliancizer, an online synthesis tool that allows transforming web pages into functional electronic gadgets. Appliancizer significantly speeds development by matching interfaces at the binary level, allowing users to transform HTML elements from digital to tangible, and automate the development process, thereby reducing the skills required to build these devices. In the future, we hope to conduct an in-depth user study of our tool.

REFERENCES

- [1] 2020. Specifications - webcomponents.org. (2020).
<https://www.webcomponents.org/specs>.
- [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*. ACM, 331–342.
- [3] Jonathan Bachrach, David Biancolin, Austin Buchan, Duncan W Haldane, and Richard Lin. 2016. Jitpcb. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2230–2236.
- [4] Flavia C Delicato, Paulo F Pires, Thais Batista, Everton Cavalcante, Bruno Costa, and Thomaz Barros. 2013. Towards an IoT ecosystem. In *Proceedings of the First International Workshop on Software Engineering for Systems-of-Systems*. 25–28.
- [5] Jorge Garza, Devon J Merrill, and Steven Swanson. 2019. Amalgam: Hardware Hacking for Web Developers with Style (Sheets). In *International Conference on Web Engineering*. Springer, 315–330.
- [6] Richard Lin, Rohit Ramesh, Antonio Iannopollo, Alberto Sangiovanni Vincentelli, Prabal Dutta, Elad Alon, and Björn Hartmann. 2019. Beyond Schematic Capture: Meaningful Abstractions for Better Electronics Design Tools. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [7] David A Mellis, Leah Buechley, Mitchel Resnick, and Björn Hartmann. 2016. Engaging amateurs in the design, fabrication, and assembly of electronic devices. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. ACM, 1270–1281.
- [8] Mozilla Developer Network. 2013. Firefox OS, 2013. (2013).
- [9] Rohit Ramesh, Richard Lin, Antonio Iannopollo, Alberto Sangiovanni-Vincentelli, Björn Hartmann, and Prabal Dutta. 2017. Turning coders into makers: the promise of embedded design generation. In *Proceedings of the 1st Annual ACM Symposium on Computational Fabrication*. ACM, 4.
- [10] Sunil Saxena and HV Kwon. 2012. Tizen architecture. In *Tizen Developer Conference, San Francisco, California*.