MakeDevice: from prototype to manufacture of mobile devices

Aron Eggens, James Hahn, Kobi Hartley, Steve Hodges, and Joe Finney

Lancaster University, UK

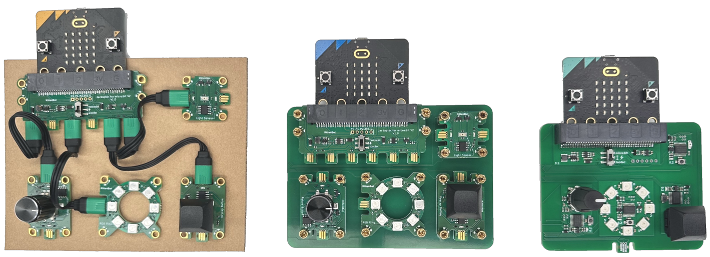

Figure 1: A smart office system at various stages of the device *flattening* process. On the left is a cardboard prototype with off-the-shelf Jacdac modules. In the center, modules are screwed down to an auto-generated carrier board. On the right, all Jacdac modules are *flattened*, and part of a single auto-generated circuit board design.

A proliferation of modular hardware ecosystems and accessible coding platforms have simplified prototyping of mobile devices. However, a gap persists in the journey from prototype to deployable product [3]. Moving beyond a functional prototype typically demands significant technical expertise and time investment in circuit design, component selection, assembly, programming and testing – barriers that prevent many promising innovations from reaching even modest-scale deployment.

Our work addresses this prototype-to-product transition through MakeDevice, a browser-based tool leveraging the micro:bit platform and Jacdac modular ecosystem to empower novice creators. By *flattening* modular hardware designs created with the micro:bit and Jacdac into more integrated solutions, introducing *virtual modules* with standardized representations, and automating device commissioning (firmware flashing and testing), we significantly lower barriers to producing deployable mobile devices.

While packages like KiCad, Altium, and Eagle serve experienced engineers, and services like Upverter and SparkFun à la Carte simplify layout and routing, all still require significant domain experience. Even recent innovations [2, 4] only begin to bridge the gap between prototyping and deployable device creation for non-experts.

Device *flattening* is the process of transforming modular prototypes into increasingly integrated hardware solutions, representing a spectrum from initial prototype to refined, product-ready devices. Our tooling enables users to easily navigate along the continuum based on their specific needs for cost, functionality, production volume, time investment, and other factors.

Our tool leverages the micro:bit, a widely adopted educational platform, alongside Jacdac [1], a modular prototyping ecosystem for sensors and input devices that integrates with both micro:bit and MakeCode. By maintaining concepts and workflows from these tools throughout, our tool should remain familiar to users of these tools at any stage of device *flattening*.

Off-the-shelf Jacdac modules are physically mounted to PCBs using screws, while *virtual modules* are integrated directly into the circuit board design itself. These *virtual modules* serve as the key enablers of the *flattening* process by functioning as digital equivalents of their commercially available counterparts. With this, they can be seamlessly substituted, allowing users to move along the continuum. Our new *GerberSockets* are an essential component of *virtual modules* as they encode additional metadata for autorouting electrical connections. By leveraging the industry standard Gerber files, we make our platform agnostic to the source of the *virtual module* designs, allowing creators to use their preferred CAD package, while still benefiting from our *flattening* technology.

Firmware programming and hardware testing form two stages of commissioning supported by MakeDevice. Our system automatically generates the appropriate firmware based on the specific device configuration. Flashing can be done through the micro:bit itself, or though an alternate RP2040 "brain" module. Both firmware flashing and subsequent device validation are done seamlessly via USB connection, eliminating the need for any additional tools or technical expertise.

The MakeDevice web application features an interactive 3D canvas for arranging off-the-shelf Jacdac modules, *virtual modules*, connectors and mounting holes. Our library contains a range of modules, from environmental sensors, to user interface elements like buttons and rotary encoders. Once a board is manufactured, the hardware can be connected to the tool to receive visual feedback on the commissioning process.

To demonstrate the capabilities of MakeDevice, we developed a smart office system (see Figure 1). The office system consists of a light sensor for desk presence detection, an RGB ring and rotary control which act as the user interface, and a button for changing status or pinging coworkers. The users can program and customise their device using MakeCode's accessible block-based editor and transition between all levels of *flattening*, while maintaining the same code throughout their development.

Our tool remains uniquely approachable for users without electrical engineering experience, while producing complete artifacts which can be turned into working products by any major manufacturer. We envision tools like MakeDevice allowing an entirely new range of designers to approach hardware production, unlocking many niche deployments of mobile and ubiquitous hardware.

References

[1] James Devine et al. "Plug-and-play Physical Computing with Jacdac". In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6.3 (Sept. 2022). DOI: 10.1145/3550317. URL: https://doi.org/10.1145/3550317.

[2] Jorge Garza, Devon J. Merrill, and Steven Swanson. "Appliancizer: Transforming Web Pages into Electronic Devices". In: CHI '21. Yokohama, Japan, 2021. DOI: 10.1145/3411764.3445732.

[3] Rushil Khurana and Steve Hodges. "Beyond the Prototype: Understanding the Challenge of Scaling Hardware Device Production". In: CHI '20. Honolulu, HI, USA, 2020, 1–11. DOI: 10.1145/3313831.3376761.

[4] Mannu Lambrichts et al. "A Survey and Taxonomy of Electronics Toolkits for Interactive and Ubiquitous Device Prototyping". In: 5.2 (June 2021). DOI: 10.1145/3463523.