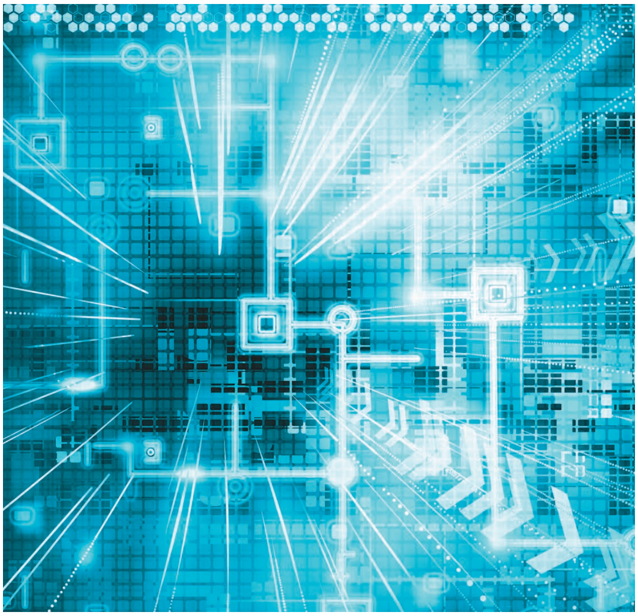

Designing Bespoke Interactive Devices

Björn Hartmann and Paul K. Wright
University of California, Berkeley

Advances in both software tools and digital fabrication technology have brought down the cost, time, and expertise required to design and prototype interconnected interactive devices.

The Invisible Computing column explores computing technologies that are becoming part of our everyday environment. To the end user these novel devices aren't computers—they're sophisticated bathroom scales, advanced irrigation systems, or multifeatured alarms. Nevertheless, creating them involves inventing and building a networked computer with specific input and output capabilities.

This article looks at the challenge of preparing students with skills to design and build this new breed of computers from the perspective of an interdisciplinary interactive design class. As computers are disguised as smart devices, the field of computer science becomes more prominent and provides opportunities for new endeavors.

Albrecht Schmidt, column editor

Walk down the aisles of a consumer electronics store, and it quickly becomes apparent that a new class of devices is pushing embedded sensing ever deeper into our everyday lives. Activity trackers count the steps we walk and floors we climb and let us compete against others online. Smart scales track our weight and graph trends against set goals on the Web. Food thermometers notify our smartphone when the turkey is safely cooked.

Many of these products are interactive devices: they don't just passively monitor or automate processes, but present information and offer control to a user through hardware and software user interfaces. While smartphones already offer many integrated sensors for detecting location, orientation, temperature, and other properties, this new product category demonstrates that software running on a commodity platform isn't always enough: designing the most appropriate solution for specific applications often requires dedicated hardware.

NETWORKED INTERACTIVE DEVICES

In contrast to earlier generations of stand-alone consumer electronics, the latest interactive devices are connected to an array of supporting computing platforms and services (M. Kuniavsky, *Smart Things: Ubiquitous Computing User Experience Design*, Morgan Kaufmann, 2010).

As Figure 1 shows, a new three-tier device and application architecture is emerging:

- custom embedded hardware devices for application-specific sensing or actuation;

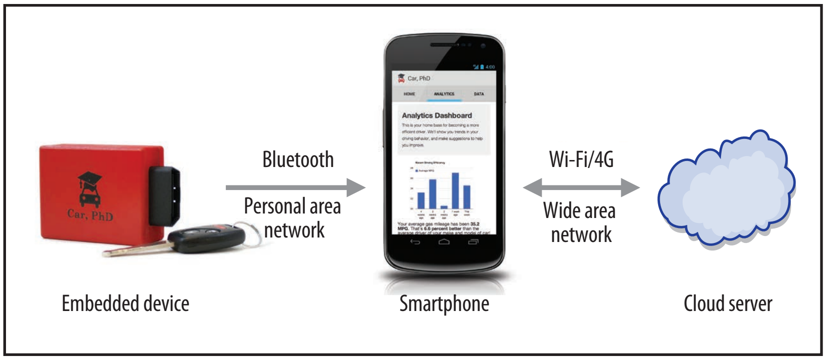

Figure 1. Emerging three-tiered architecture of modern interactive devices: applications span multiple system platforms and networking technologies.

- the user's mobile device for delivering graphical user interfaces; and
- server infrastructure in the cloud for storing, aggregating, and processing data.

The hardware devices have application-specific form factors and electronics. Sometimes these devices connect directly to the Internet, but power constraints often lead to designs in which the device uses low-energy, short-range wireless technology such as Bluetooth to communicate with a smartphone, tablet computer, or other local hub. That local device in turn exchanges data with the Internet through Wi-Fi or 3G/4G radios.

TRENDS ENABLING RAPID PROTOTYPING

Advances in both software tools and digital fabrication technology have reduced the cost, time, and expertise required to design and prototype such interconnected interactive devices.

Open source design and engineering software such as compilers for embedded processors and CAD packages are now available when, previously, expensive proprietary solutions were a gating factor. Tools such as Arduino that explicitly focus on supporting amateur and novice users have also lowered the expertise threshold and made experimentation more accessible.

At the same time, the physical fabrication of prototypes has become much easier as additive manufacturing processes such as fused-deposition modeling (FDM) and desktop circuit board milling are now widely available and dropping in price. The rise of companies that offer manufacturing as a service at single quantities also makes it possible to affordably outsource aspects of fabrication.

Together, these shifts have led to a sea change in product design: projects that before required teams of specialists or researchers are now within reach of students, hobbyists, and budding entrepreneurs. The increasing number of hardware startups has led some Silicon Valley observers to declare—maybe prematurely—that "hardware is the new software."

TEACHING INTERACTIVE DEVICE DESIGN

To enable rapid device design and prototyping, we've created a new teaching space at the Center for Information Technology Research in the Interest of Society (CITRIS). The Invention Lab at UC Berkeley (http://invent.citris-uc.org) is built around a "learn-build-launch" model that combines instruction, access to tools, and support for entrepreneurship through an incubator program. The lab has a broad range of electronic and digital fabrication tools, a store of frequently used parts, and expert guidance from lab managers.

Building devices in 14 weeks

In our interactive device design class, small interdisciplinary student teams create fully working interactive product prototypes in a single, 14-week semester. The class relies heavily on experiential learning: short lectures on conceptual topics—for example, how sensors work—alternate with hands-on exercises that ask students to design new sensing approaches, text entry devices, and game controllers, among other things.

As Figure 2 shows, students learn about embedded programming, sensing and actuation circuits, circuit board design and soldering, 3D modeling and fabrication, wireless communication, and interaction design. While this list might sound daunting, we've found that both graduate and advanced undergraduate students from various engineering and information science disciplines can successfully pick up these skills over the course of a single semester, provided they're already proficient programmers.

Sample student projects

Some of the novel devices envisioned and produced by students at UC Berkeley address the CITRIS core themes of personal health, sustainability, and smart transportation. For each, students produced and demonstrated a working prototype.

Drinke and SwigSmart. Designed by Amy Pavel, Steve Rubin, Sean Chen, and Elliot Nahman, Drinke, shown in Figure 3, is a smart beverage cup that tracks liquid consumption over time. A force-sensitive resistor monitors liquid level inside the cup, while a low-cost spectroscope consisting of a photocell and tricolor LED, coupled with a temperature sensor, infer the type

Figure 2. Students at UC Berkeley's CITRIS Invention Lab learn skills such as embedded programming and 3D modeling and printing to design both the software and hardware of their projects. (Photo on left by Cheryl Martinez; photo on right by Björn Hartmann.)

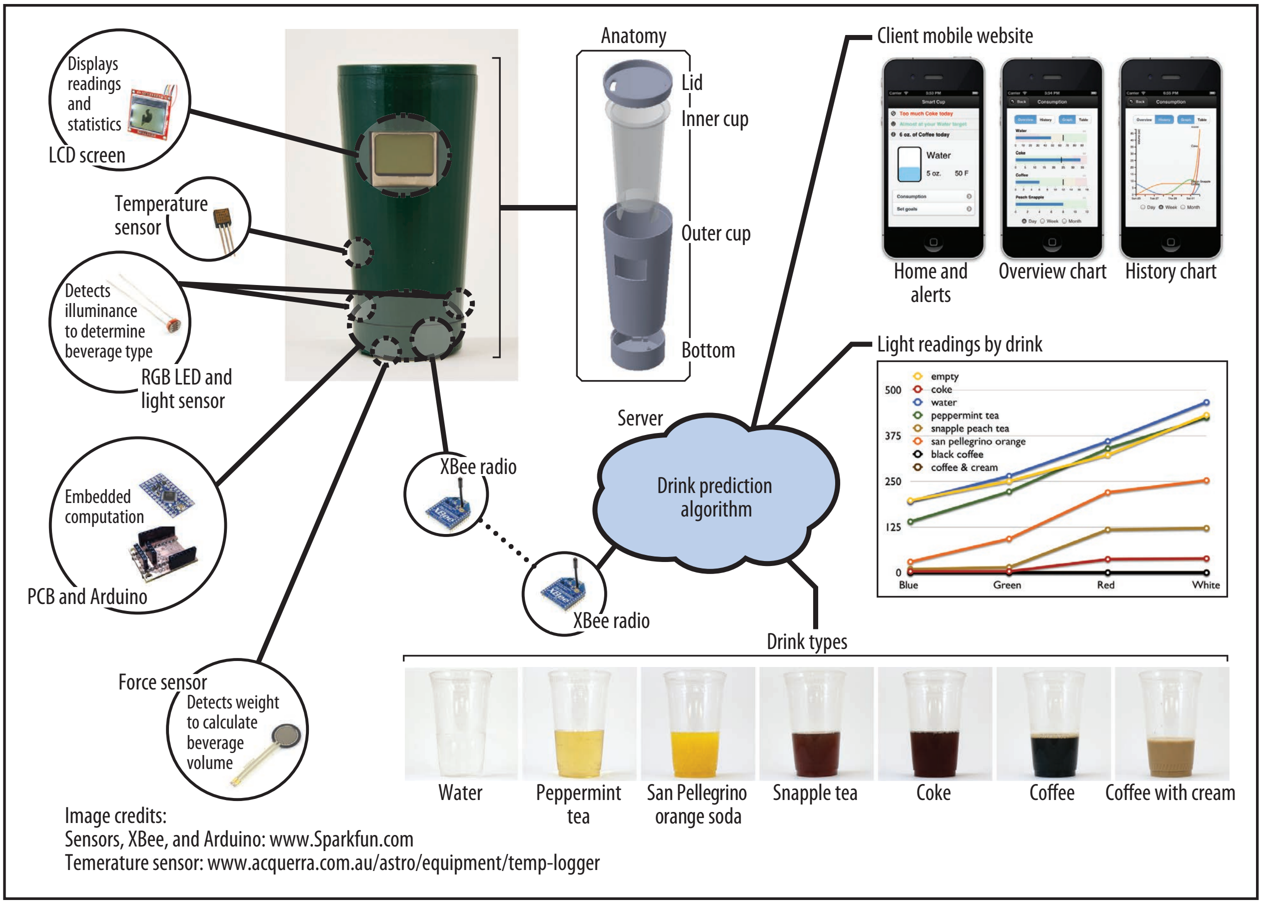

Figure 3. Drinke hydration-tracking cup.

of liquid. The students determined that beverages have unique light transmission profiles at different wavelengths. Drinke users can set hydration goals and limits—for example, for daily caffeine intake—on their smartphone and are alerted when they exceed those limits.

A related device, the SwigSmart cup, designed by Mozziyar Etemadi and Alex Heller, targets patient-physician communication for dialysis patients and others who have to carefully monitor their fluid intake. The device enables

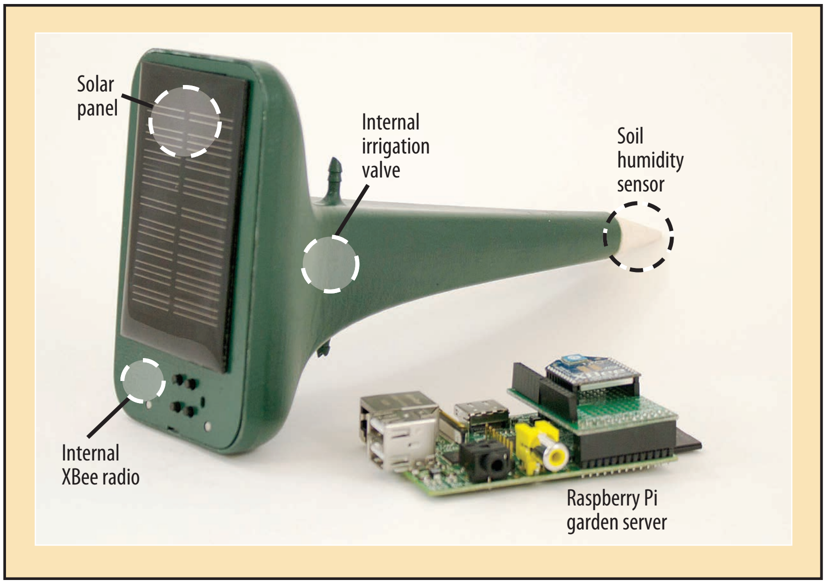

Figure 4. H2O IQ smart drip irrigation controller.

physicians to review charts of their patients' fluid intake online. The students are currently pursuing venture capital seed funding for this device.

H2O IQ. Designed by Valkyrie Savage, Shiry Ginosar, and Mark Fuge, H2O IQ, shown in Figure 4, is a tablet-controlled, solar-powered drip irrigation system. A humidity sensor at the tip of each "spike" records soil moisture; an internal servo in the 3D-printed enclosure opens and closes a drip irrigation line valve. Individual devices in a garden communicate with a central garden server, which also acts as a webserver that hosts the HTML-based user interface. Gardeners can review graphs of humidity readings over time and adjust watering plans through this Web application.

Car PhD. Designed by Daniel Haas, Daniel Bruckner, and Christopher Thompson, Car PhD, shown in Figure 1, tracks driving habits through car telemetry data and offers driving suggestions on a smartphone when the driver is not operating his or her vehicle. A small dongle plugs into the car's diagnostic OBD-II port and streams real-time driving data over Bluetooth to the driver's smartphone, which relays the data to a cloud server for driving pattern classification. The UC Berkeley-affiliated startup Automatic (www.automatic.com) is commercializing similar, but independently developed, technology.

Kinectograph. Designed by Derrick Cheng, Peggy Chi, and Taeil Kwak, Kinectograph is a video recording device that automatically pans and tilts to follow specific body parts, such as the hands, of a user in a video. It uses a Microsoft Kinect depth sensor to track users and adjusts the camera angle through a pan-tilt gimbal mount. Users control and configure Kinectograph wirelessly through a Web-based tablet application with real-time video preview.

OPEN RESEARCH QUESTIONS

Although all the teams in our class have successfully created working prototypes, some steps remain hard in interactive device design. Our experience suggests that even though tools for individual design stages are quite sophisticated, integrating the different components of a connected, interactive device at both the software and hardware levels is a significant challenge.

Software design

Many interconnected devices are essentially distributed systems that require developers to write code in different programming languages and paradigms:

- embedded code for sensing and actuation,
- native mobile applications or Web applications for displaying a user interface, and
- server code for storing and aggregating data across users.

Software in one layer must be able to communicate with software in other layers, often through multiple networking technologies, in a robust way. In their projects, students often implement brittle one-off solutions to these challenges. In response, researchers at UC Berkeley's Ubiquitous Swarm Lab (http://swarmlab.eecs.berkeley.edu) are investigating new toolkits that make it easier to rapidly write software that spans multiple heterogeneous layers.

Hardware design

In designing the hardware, students stumble when integrating electrical and mechanical designs: mounting bosses for circuit boards are misaligned, tactile switches don't have enough clearance to be pushed, and different parts frequently interfere with one another in early prototypes. The reason for this is that the tools used to design physically interdependent enclosures and electronics are completely separate today. Researchers at the Berkeley Manufacturing Institute (http://bmi.berkeley.edu) are studying lightweight tools that enable users to specify integration constraints to address this problem.

From prototype to product

Another open issue is how to move beyond a single working prototype toward a marketable product. The devices our students create almost always make some important tradeoffs—in size, power requirements, network configurations, or mechanical and software robustness. Designers currently lack tools to modify a prototype device so that it's suitable for a first small production run of dozens, hundreds, or thousands of units.

Many crowdfunded hardware startups on platforms like Kickstarter are currently learning about the complexities of design for manufacturing the hard way. Even though many had a working prototype when they solicited funding, stories of production delays or even project cancellations aren't uncommon and underscore the need for future research in this area.

These difficulties notwithstanding, we're clearly entering an era of bespoke interactive devices that can be designed and produced to exactly fit user needs and context. We invite readers to join in this exciting effort: what device will you design?

Björn Hartmann *is an assistant professor of computer science and Qualcomm Faculty Fellow at the University of California, Berkeley, where he also codirects the Berkeley Institute of Design and the Swarm Lab. Contact him at bjoern@eecs.berkeley.edu.*

Paul K. Wright *is a professor of mechanical engineering and holds the A. Martin Berlin Chair at the University of California, Berkeley, where he also directs the Center for Information Technology Research in the Interest of Society (CITRIS) and codirects the Berkeley Manufacturing Institute (BMI). Contact him at paulwright@berkeley.edu.*

Editor: Albrecht Schmidt, University of Stuttgart, Germany; Albrecht@computer.org

Selected CS articles and columns are available for free at http://ComputingNow.computer.org.