BitWear: A Platform for Small, Connected, Interactive Devices

Kent Lyons, David H. Nguyen, Shigeyuki Seko, Sean White
Nokia
{kent.lyons, david.h.nguyen, shigeyuki.seko, sean.white}@nokia.com

Daniel Ashbrook
Samsung
d.ashbrook@samsung.com

Halley Profita
University of Colorado at Boulder
halley.profita@colorado.edu

ABSTRACT
We describe BitWear, a platform for prototyping small, wireless, interactive devices. BitWear incorporates hardware, wireless connectivity, and a cloud component to enable collections of connected devices. We are using this platform to create, explore, and experiment with a multitude of wearable and deployable physical forms and interactions.

Author Keywords
Interaction, UI Toolkit, Sensors, IoT, Web

ACM Classification Keywords
H.5.2 Information Interfaces & Presentation: User Interface

INTRODUCTION
The continuing miniaturization and integration of computation and radio technology enables new forms in the space of wearable and ubiquitous user interfaces [3,4]. System-on-Chip (SoC) solutions that combine RF and microcontrollers now measure less than 4mm on each side, making working wireless prototypes at small scales possible. In this work, we investigate small, interactive, functional devices that can be worn or deployed. We also describe our event-based cloud infrastructure designed to support these devices.

One of our goals was to create a device that pushed towards the limits of small size yet retained enough functionality to be open and useful for a many goals (e.g., provide generic user I/O with supporting computation and communication). We wanted devices that could be powered for at least a day at a time. We started with some of the simplest modes of interaction – pressing a button (input) and seeing an LED (output) – in a variety of physical forms. As a research community, we often turn to simulated or wired prototypes; we are now shifting into the realm where these devices can be made using standard electronics capabilities. By building, deploying, and experimenting with such devices, we intend to inform future iterations of research in this space.

UIST'13, Oct 08-11 2013, St Andrews, United Kingdom
ACM 978-1-4503-2406-9/13/10.
http://dx.doi.org/10.1145/2508468.2514716

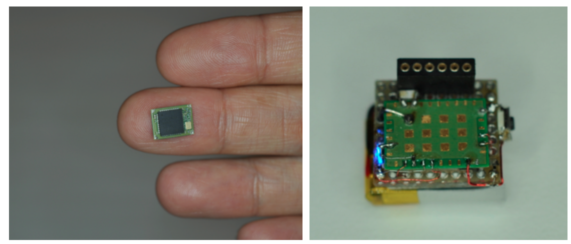

Figure 1. Murata BLE module with a TI CC2541.

CONNECTIVITY, POWER, AND SIZE
As part of the Bluetooth 4.0 specification, there are provisions for Bluetooth low energy (BLE) devices that take less power than other flavors of Bluetooth and are designed to handle low bit rate sensor data. Manufacturers are selling BLE SoCs that include small microcontrollers. For example, the Texas Instruments CC254X has a small 8051 processor, while Nordic Semiconductor's nRF51822 has an ARM Cortex M0 CPU. Nordic's chip-scale package measures only 3.5x3.8x0.5mm. Murata created a module using TI's solution that measures 10.4x7.7x1.8 mm (Figure 1a). This module serves as the base of our BitWear device (Figure 1b) where we have added an antenna, crystal and battery to be a fully functional embedded interactive system.

BITWEAR DEVICES, INTERACTION, AND FORM
We created a variety of BitWear-based devices incorporating a button and RGB LED. Our firmware is simple and only reports button presses and responds to requests to illuminate the LED via connections to other devices. This configuration gives us significant interactive expression flexibility while keeping the BitWear device simple. The button provides button down and up, press and hold, single press, double press, and n-press. The RGB LED gives us dimensional simplicity with color and temporal complexity [2]. Other sensors such as temperature or motion can be added on GPIO or the I2C bus.

We prototyped a variety of 3D printed forms using these modules. We explored both wearable forms including rings and pendants as well as deployable forms such as buttons with embedded magnets that attach to metal surfaces (Figure 2).

WORLDWIDE EVENT QUEUE

Although BitWear devices are simple, we connect them to other platforms in a general purpose way. Currently, we use BLE-enabled phones with WiFi and 3G/4G data connectivity to bridge to an open cloud infrastructure. Regardless of the device, all user input and other sensor data are relayed to the cloud as events.

Communication with the cloud is done via web standards (JSON over websockets for bi-directional communication). The cloud server puts all of the events (regardless of sensor type or where in the world they came) onto a ZMQ event bus. This event model enables a worldwide event queue that handles both user input and output. Thus, a button press on a BitWear device on one side of the planet can trigger an LED on the other side of the planet.

In many ways, our architecture is similar to the event mechanism of a desktop UI toolkit or of other physical prototyping toolkits such as Phidgets [1]. We have stateful processing of events in the cloud to transform raw input into events with semantic meaning. This is analogous to a UI system transforming raw up and down mouse button readings into click or double click. We have state machines for buttons (press, double press), accelerometer (shake, flip), light (lighter, darker), etc.

Connected wirelessly to the global event queue, BitWear devices are NOT physically tethered by cables, and thus can be worn on the user or deployed in the environment. Our intention is to shift the conceptual model away from a physically tethered peripheral to a stand-alone device with a logical role for the user. The web-standards based BitWear API enables this shift in perspective.

EXTENSIONS AND EXAMPLE USES

The simple BitWear platform enables many complex and compelling uses, scenarios, and extensions. For example, we have added other sensors and actuators (e.g., accelerometers and temperature sensors) to BitWear devices; these additional sensor readings are reported to the cloud as events. We have also connected the sensors on phones to the cloud infrastructure. In this configuration, the phone provides many additional sensors and actuators (e.g., play sounds or take photos) for the worldwide event queue. We have also connected WiFi based sensors and actuators such as the Electric Imp to our cloud infrastructure.

The cloud backend makes it straightforward to connect to other cloud services such as Twilio which provides a REST based API for sending and receiving SMS text messages. These capabilities are treated just like any other event in the system and put on the worldwide event queue.

Because of the openness of the BitWear platform, users can design compelling uses for their specific needs with these simple devices. For example, as an intimate form of communication, a user presses a button on their BitWear ring, which lights up an LED on a partner's ring. Or a user configures the system so when they press a button on a pendant, the system uses Twilio to send a preconfigured SMS to their spouse. Yet another example, BitWear rings and pendants can be configured to light up to alert users of important notifications such as phone calls.

Figure 2. (a) A variety of 3D printed physical forms including (b) rings, (c) buttons, and (d) pendants. (e) Overview map of over 184M worldwide input events.

DISCUSSION AND CONCLUSION

We have had a research prototype of this system in place for over 10 months, collecting 184,163,648 input events and generating 38,517 output events across 7 countries on 3 continents. We were initially concerned about latency of using a web API and remote server, but through our informal testing the system responds fast enough for many interactive scenarios. In the future, we plan to build on this work through deployment studies and use.

REFERENCES

1. Greenberg, S. and Fitchett, C. Phidgets: easy development of physical interfaces through physical widgets. In *Proc. UIST '01* (2001), 209-218.
2. Harrison, C., Horstman, J., Hsieh, G., & Hudson, S. Unlocking the expressivity of point lights. In *Proc. CHI '12* (2012), 1683-1692
3. Ni, T. and Baudisch, P. Disappearing mobile devices. In *Proc. UIST '09* (2009), 101-110.
4. Polastre, J., Szewczyk, R., & Culler, D. Telos: enabling ultra-low power wireless research. In *Proc. Information Processing in Sensor Networks, IPSN '05* (2005), 364-369.