Hands and Fingers: A mobile platform for a person-centric network of computational objects

John Kestner and Henry Holtzman
Information Ecology Group
MIT Media Lab
20 Ames St., Cambridge, MA 02139
{jkestner, holtzman} @media.mit.edu

ABSTRACT
We describe a practical vision of ubiquitous computing with tangible interfaces, that orbits around an individual and is mediated by his or her personal consumer electronic devices. We illustrate this with a software and hardware platform for creating a personal area network of information accessories, connecting people physically with network-accessible information.

The toolkit hardware consists of a Bluetooth-connected microcontroller to which input sensors and output actuators are connected. The software component consists of a networking library residing on the microcontroller, and a corresponding application on a handheld consumer electronics device that routes data between the Bluetooth modules and the Internet. This platform is used to create three examples of locally networked consumer objects that expose their inputs and outputs to a wide area network.

Author Keywords
Ubiquitous computing, mobile computing, interaction prototyping platform, wearable computing.

ACM Classification Keywords
H.5.2. Information interfaces and presentation: User Interfaces.

General Terms
Design, Human Factors.

INTRODUCTION
Cell phones and other handheld electronics devices are gaining abilities previously reserved for personal computers, in addition to their own unique attributes. With rich UI options, powerful processors, and multiple avenues of connectivity, they are primed to host the next major HCI experience. But this shift need not be confined to the screen. Our computing experiences are dominated by activities such as social networking and information retrieval, consuming and sharing bite-sized chunks of content with little thought. The exponentially expanding flood of information from the Internet and sensors embedded in our lives is too large and varied to be represented in traditional formats, especially as screens have only shrunk. Cell phones, already at the center of this lifestyle, are well-positioned to carry us closer to invisible, ubiquitous computing.

Weiser’s original vision of viewing and manipulating data using many ‘tabs’ and ‘pads’ was screen-centric. [3] He knew that this was not an optimal experience, stating, “A good tool is an invisible tool,” and giving the example of eyeglasses which disappear in everyday use. [4]

Permission to make digital or hard copies of all or part 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. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 2010 ACM 978-1-60558-841-4/10/01...$5.00.
Spimes reside both in the physical world and in the digital world, and so encourage a new class of interactions. Prototyping the systems behind these interactions has been possible for technically savvy researchers, but the designers who best understand the end user, and the bleeding-edge users themselves who tend to discover novel uses of their tools, are lacking a way to string together these invisible computational tools. This paper outlines a platform for a future where consumer electronics will be distributed and open to user modification, defined by Hand software and Finger microcontrollers, [Fig. 1] developed to aid those who are designing spimes and our interactions with them.

RELATED WORK
Pekka, et al. describe using a mobile device as a human interface for wearable technology. [10] Their work focuses on improving the durability of wearables by removing interface elements, and only speculates on the mobile device as a substitute interface for that removed from the wearables. The work in this paper implements routing software and a configuration interface, and considers applications beyond wearables.

Arduino is perhaps the best-known of existing microcontroller prototyping platforms. Finger hardware shares the AVR microprocessor heart and in fact can run Arduino code, just as Arduino hardware can be loaded with the Finger networking code. But the focus of this work is enabling a consumer electronics device to manage communications between the microcontrollers and information resources via the Internet (which may be other Fingers). The microcontroller code written is intended to support this focus.

Closer to the work presented here are Open I/O and Pinkies, [2] which we consider ancestors to Hands and Fingers. Pinkies are remote microcontrollers that use Ethernet ports to talk to each other over the Internet; Open I/O provides the glue: software running on a desktop on a Pinkie’s local network, and a central server that allows the microcontrollers to find each other. The Open I/O framework speaks to a larger need for standardized protocols by which devices can find and learn how to communicate with each other. Sterling suggests that successful products will have public APIs that allow them to be repurposed by the user in ways that the manufacturer or designer could not foresee.

SYSTEM DESCRIPTION
The platform consists of Fingers, small I/O boards that are embedded in personal objects, and Hand routing software running on a cell phone. These are connected by knuckles, software layers in the routing software that manage the communications hardware, such as Bluetooth or Zigbee.

A Finger contains an AVR microcontroller with several of its pins connected to high-current optocouplers to easily connect it to a variety of actuators, a UART Bluetooth module, and a lithium-polymer battery that’s charged via a mini USB connector. It is loaded with a networking library written for the occasion (and a bit of glue code) that waits for incoming data from the Bluetooth module and switches output pins in response, and also listens to input pins and relays any changes up the stream to be reported back to its associated URL. The pins and associated behaviors are configured by the developer using software on a personal computer, allowing each microcontroller to be flashed once with the library. Pins retain a state until told to change it, to remain predictable in case of loss of connection somewhere along the chain.

The Hand routing software runs as a background application on a personal computer or a cell phone (currently a Symbian S60 device) with persistent Internet access. When first paired with a Finger device, it receives a URL to poll for data, or a URL can be assigned from the routing software. It then manages connections to known and available Fingers, only connecting when new data is available via the related URL and disconnecting immediately, to save power and to get around limitations of multiple simultaneous connections. The software can parse XML structures received from remote servers, or be used as a framework that quickly enables custom applications.

While the Finger currently communicates upstream via Bluetooth, any radio, such as Zigbee, that understands the UART protocol can be used if a corresponding knuckle to manage the connection is written for the Hand router.

Figure 2. An original handmade Finger with LiPo battery.

DEVELOPMENT
The first Finger boards were custom made to meet each application’s space requirements, as well as to make initial improvements simple. [Fig. 2] However, they have all been flashed with the same code. While a smaller all-purpose Finger has been designed using a PCB and surface-mount transceiver, Arduino is now the preferred platform upon which to load the networking library in order to make it useful to a wider audience.
The routing software has been written in Python for Symbian S60 devices, and in Java for desktop computers. It can run in the background, and the phone version has a simple GUI for configuration options.

THE PLATFORM IN USE
The Finger platform has been used to implement three body-area networked peripherals that explore new forms of human-computer interaction: haptic wallets connected to personal financial information, thermal actuating earmuffs connected to personal social information, and an alarm clock connected to social productivity information. Descriptions of these follow.

Proverbial Wallets
The Proverbial Wallets connect people physically with their personal financial information accessed via the internet. They attempt to guide the user toward responsible decisions by delivering the appropriate message at the appropriate time. Haptic feedback is used to connect intangible virtual transactions and account balances to the user’s physical world, and at the same time provide a private ambient channel of communication for sensitive data.

The wallets use Finger modules to control actuators such as a vibrating motor, servo, and a geared motor that is shorted to produce rotational resistance. Field testing, which is in progress, is made much easier by the Finger platform. The hardware module’s compact size and rechargeable flat battery suit it for the stresses wallets typically undergo. Using test subjects’ existing cell phones to run the Hand software minimizes needed infrastructure and distractions for the users. Using one’s own phone also improves trust, as the software indirectly accesses the user’s personal financial information.

My Ears Are Burning
My Ears Are Burning explores the concept of ambient social awareness through a tangible user interface. By making a user aware of attention being paid to his online presence, or the overall mood of his social group, we hope to nurture a sort of social proprioception. Heating elements placed on the user’s ears are activated when, for example, his web page is viewed, he is tagged in a photograph on Facebook, or the overall emotional valence of friends’ messages are positive.

This takes the form of a pair of earmuffs and a woolen hat with a nichrome heating element sown into the ear flaps. A Finger module fits into either of these clothing items with little bulk and drives the heating element through an optocoupler when the Hand software polls a website for changes in the related data.

DoingDoing
Daydar is a framework for socially-motivated productivity. It uses to-do lists as the primary input, and a variety of physical and digital artifacts as output. These outputs may be abstract or focused, but they all use an awareness of the habits of a user’s social circle in order to influence her own habits. Within this framework we are experimenting with various systems, both physical and digital, that allow a user to monitor and reflect on her own and other’s productivity.

DoingDoing is a socially aware alarm clock that gets the user’s Daydar feed. When she hits snooze, a bell pings once every time someone else in her social group checks off a to-do item. Hearing the activity nudges her out of bed, or if it’s quiet, she can laze in bed knowing that she’s not falling behind in the rat race. When she turns off
the alarm and gets up, other DoingDoings that are in snooze mode will ping.

Unlike the first two peripherals described above, this is not an on-body device. DoingDoing was built using a Finger module that triggers solenoids to ring the bells when it receives commands from routing software that’s listening to a Daydar feed accessible via the Internet, and relays back user input (alarm turned off, the user is out of bed). This client may run on the user’s cell phone, perhaps lying on a nightstand next to the clock, or a nearby computer.

Figure 5. DoingDoing socially aware alarm clock.

FURTHER WORK AND CONCLUSION
The microcontroller networking library is being packaged into an Arduino library which will be released to the community. Currently, Fingers have a few built-in behaviors, which can be added to with knowledge of microprocessor programming. The next iteration will add the ability to program Fingers using the software running on a cell phone, making it possible for end users to create interactions not defined by the designer. Defining a more robust service discovery mechanism may also help end users link disparate objects and data feeds.

The focus of our work is to examine how the end user can use software on their consumer electronic device to manage Fingers embedded in objects, and to encourage user-designed interactions with an ecosystem of computational objects. We believe that cell phones or similar personal electronics are a practical way to enable this now, and may also lead to useful approaches to related ubiquitous computing issues such as privacy and notification management.

ACKNOWLEDGEMENTS
We would like to recognize the rest of the Information Ecology group, Matt Hirsch and ReeD Martin, for their input into this research direction. We also want to thank Danny Bankman for his hand in designing the Finger board.

REFERENCES
   http://hdl.handle.net/1721.1/36319
   http://www.wired.com/techbiz/media/magazine/15-07/st_thompson
8. Arduino physical computing platform,
   http://www.arduino.cc/