A Personal-Area Network of Low-power Wireless Interfacing Devices for Handhelds: System & Hardware Design

Lin Zhong[*]
Dept. of ECE
Rice University
Houston, TX 77251
lzhong@princeton.edu

Mike Sinclair
Hardware Devices Group
Microsoft Research
Redmond, WA 98052
sinclair@microsoft.com

Niraj K. Jha[†]
Dept. of Electrical Engineering
Princeton University
Princeton, NJ 08544
jha@princeton.edu

ABSTRACT

Handhelds, such as smart-phones and Pocket PCs, have the potential to become the computing, storage, and connectivity hub, or *Digital Hub*, for pervasive computing. However, their current interfacing paradigms fall short of achieving this goal. To meet this challenge, we present the system and hardware design for a Bluetooth-based personal-area network (PAN) of low-power wireless interfacing devices. The network consists of a wrist-watch, a single-hand single-tap multi-finger keypad, and a smart speech portal, and a GPS receiver. These devices serve a handheld in a synergistic fashion, collectively providing the user with immediate and more natural access to the computing power and enabling more and better services.

Categories and Subject Descriptors: C.m [Computer systems organization]: Miscellaneous; D.2.2 [Software Engineering]: Design Tools and Techniques—*User interfaces*

General Terms: Design, Measurement, Human Factors.

Keywords: User interface, Handhelds, Low power design, Personal-area network, Bluetooth.

1. INTRODUCTION

We believe that handhelds are on their evolutionary track to become the digital hub for personal computing. To a significant extent, the challenge for such a role comes from developing interfacing technologies that enable users to access the computing power anytime and anywhere, which we call *Pervasive Interfacing*. This work represents our initial step towards meeting this challenge. In view of the limitations of popular interfacing paradigm, we follow a number of principles for developing new interfacing devices:

- Separating interfacing from computing;
- Separating information capturing from storage;
- Employing more interaction channels such as speech;
- Simplicity: low-power and inexpensive designs.

[*]Lin Zhong was an intern with Microsoft Research in the summer of 2004 and later supported by a Harold W. Dodds Honorific Fellowship from Princeton University.

[†]Niraj K. Jha was supported in part by NSF under Grant No. CCF-0428446

MobileHCI'05, September 19–22, 2005, Salzburg, Austria.

We follow the first two principles by employing Bluetooth to connect a handheld and its interfacing devices into a PAN. We design a keypad, called *violin-pad*, which is small enough to be attached to a keychain. Since it is physically detached from the handheld, the user can hold it in one hand and type in a fashion similar to playing violin. We also design a wrist-watch, called *cache-watch*, which exports a display service to the handheld so that the latter can displays information on the wrist. We realize that one of the largest challenges to speech-based interfaces lies in delivering a noise-resilient and high-quality voice stream to the handheld in a user-friendly fashion. We design a *smart speech portal* based on bone-conduction sensing to respond to this challenge. We have incorporated a Bluetooth GPS receiver, as an example of information capturing devices, into the system as well. We believe that new interfacing devices have to be inexpensive to initiate a market and train users, and they have to be low-power to obviate frequent recharging. Accordingly, we design the wireless interfacing devices as inexpensive add-ons to the handheld. We partition the interfacing task so that only the minimal set of functionalities are kept on the add-ons.

The paper is organized as follows. We present the design and implementation of the Bluetooth-based PAN in Section 2. In Section 3, we provide details of each of the interfacing devices: cache-watch, violin-pad, and smart speech portal. We address related works in Section 4 and conclude in Section 5. Although user studies are critical for such a system that is intended for interfacing and pervasive applications, we focus on the *system and hardware design issues* in this paper due to space limitations.

2. BLUETOOTH-BASED PAN

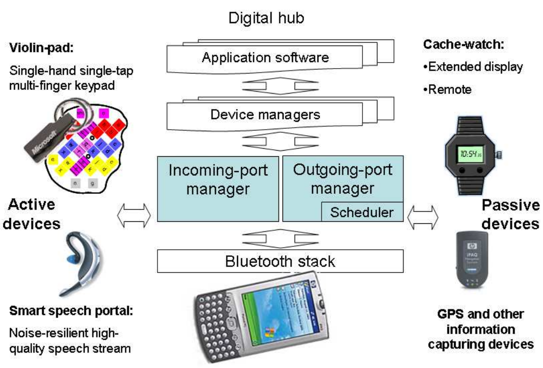


Figure 1: System overview of the PAN

The PAN is illustrated in Figure 1. The handheld, an HP iPAQ 4350, is the digital hub. The PAN manager (shaded part) and device managers are software installed on the handheld. The PAN manager coordinates communications between the handheld and different interfacing devices, while device managers interpret raw data received from interfacing devices, control them, and function like device drivers.

2.1 Bluetooth

Promi-ESD^{TM} [8], Bluetooth-RS232 adapters , are used for all add-on devices. Unfortunately, Bluetooth low-power modes are not supported for communication with the iPAQ Bluetooth. We have to rely on the supported APIs to configure the parameters for Page Scan/Paging and connect or disconnect for power management. To establish a connection between two Bluetooth devices, one of them has to initiate the connection by *Paging*. It is called an *active* device. The other device must do *Page Scan* to accept the initiation and establish the connection. It is called a *passive* device. Both Paging and Page Scan are carried out in sessions.

2.2 PAN manager

The PAN manager was developed using Microsoft Embedded Visual C++ and the BTAccess library [1]. It consists of two major components and interacts with the device managers and Bluetooth stack. The *incoming-port manager* controls the connections of the handheld with active devices such as the smart speech portal and violin-pad. It forces the handheld to enter a Page-Scan session periodically to catch possible Paging from active devices. Once a connection is established, the corresponding device manager is called.

The communication delay between passive devices, such as the cache-watch and information-capturing devices, and the handheld is usually tolerable. Therefore, the *outgoing-port manager* schedules the communication to share the port among all passive devices. The corresponding device managers run even when there is no connection. They interact with applications, and send requests for a connection to the scheduler. The port manager buffers these requests and schedules its outgoing connection accordingly. When a connection to a passive device is scheduled, the outgoing-port manager forces the handheld to enter a Paging session for a certain period of time or until the connection with that device is established.

2.3 System power optimization

Since Bluetooth power consumption is the most significant overhead we pay for following the first two principles, we focused on it for power optimization.

Scheduled communication for passive devices: For a passive device, Bluetooth consumes most of its energy in the Page-Scan sessions since data communication is usually very brief. Fortunately, the handheld knows when it needs to talk to a passive device from the outgoing-port scheduler. When the handheld is connected to a passive device, the outgoing-port manager predicts when the former will seek communication with the latter again based on history or a prefixed schedule. The manager then notifies the passive device just before disconnecting. The passive device will keep its Bluetooth module down until just before the next scheduled communication. The handheld manages its own Bluetooth accordingly. Such an arrangement minimizes the time devices spend in Paging/Page Scan. Only when they lose synchronization do they enter a Paging/Page-Scan session periodically to re-synchronize.

Tradeoffs for active devices: Active devices typically seek connection with the handheld when the user request it. The connection latency is thus important. The handheld Bluetooth would need to stay in a Page-Scan session all the time, which will drain the battery very quickly. So we trade connection latency for energy saving on the handheld. The handheld enters a Page-Scan session for 2 seconds every 4 seconds. With an extra latency of 2 seconds, we reduce the energy consumption by half. Moreover, users can manually start or stop the handheld Bluetooth Page-Scan session if more energy savings are desired.

Limitations: Despite the above optimizations, we still suffer from limitations imposed by Bluetooth itself and its implementations on iPAQ and Promi-ESD^{TM}. When we started our project, only two wireless PAN technologies were commercially available: Bluetooth and ZigBee. We chose Bluetooth over ZigBee for two reasons. First, a variety of Bluetooth modules were widely available and high-end handhelds had already been equipped with Bluetooth. Second, ZigBee does not provide a data rate high enough for media applications we envisioned for the future. We are now evaluating the possibility of a heterogeneous wireless PAN with ZigBee for low data-rate applications and Bluetooth for high data-rate ones.

3. WIRELESS INTERFACING DEVICES

In this section, we provide details on the cache-watch, violin-pad, and smart speech portal.

3.1 Cache-watch

We begin with the cache-watch, an extended and immediate display for the handheld. "Cache" is used in the name since the watch caches information from the handheld. It also resembles the memory cache in that it hosts simple and frequent interactive tasks outsourced from the handheld.

Hardware design: The prototype is shown in Figure 2. There are only three major components: Promi-ESD class II, Microchip PIC16LF88, and an 8-by-2 monochrome character LCD. There are also three buttons. Users can use them to browse and manage the content displayed, and make confirmations manually. The watch is powered by an 800mAh pack of three AAA batteries. As mentioned before, the Bluetooth module consumes most of the power. The whole design draws less than 7mW when the Bluetooth module is off, 76mW when the Bluetooth is in a Paging Session, 110mW when exchanging data with the handheld.

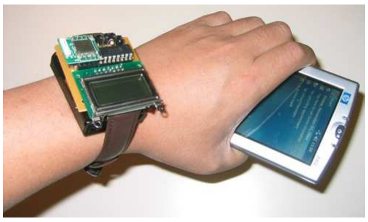


Figure 2: Prototype for the cache-watch

Software design: The PIC16LF88 storage is partitioned into chunks to cache messages from the handheld. Each chunk has meta-data to indicate whether it contains a valid message, how it is to be displayed and what its priority is. The software is extremely simple in that it just displays valid messages according to their meta-data. The storage is managed and updated according to commands from the user or handheld with an application-layer protocol. A command can specify both the text message and its meta-data. The scheduled communication technique discussed in Section 2.3 is implemented through commands.

The display has three modes: automatic, manual, and idle. In the automatic mode, cached messages are automatically scrolled on to the display line letter by letter. A message can be scrolled at different speeds, for different times, and blink based on its meta-data. In the manual mode, the user can use two buttons to browse messages, scroll a message, delete it, or make a confirmation. In the idle mode, no message is shown and the display can be turned off. The user can put the display into different modes with the buttons.

To ensure simplicity of design, the handheld is responsible for implementing applications and the device manager to utilize the watch as an extended display. This also facilitates application development.

3.2 Violin-pad

It is well-known that slow text entry is an obstacle to many services on handhelds. We next present our solution, a keypad inspired by the violin, called violin-pad.

Design: Figure 3 shows the violin-pad prototype. The electrical part simply consists of two modules: Promi-ESD^{TM} class II and Microchip PIC16LF873. However, the mechanical part is very challenging due to its size. 12 of the 20 mechanical buttons are two-way rocker buttons, i.e., they can be pressed toward either the top or the bottom end with different directions corresponding to different inputs.

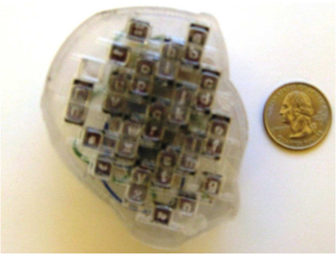


Figure 3: Prototype for the violin-pad

The alphabet and digit input relies on the 12 rocker and 4 regular buttons. They are organized into four columns, each akin to a violin string, and four rows, to each of which a finger is devoted. The buttons in the third column have a prominent texture so that the user can feel which column the fingers are on. For the current version, the keys are organized primarily according to the alphabet. Since the user's hand is in control when the violin-pad is used, there is a button for the user to switch it on/off. Upon being switched on, the violin-pad enters a Paging session to connect to the handheld. It draws about 33mW power during active usage.

Usage: The violin-pad is intended to be used in a fashion similar to how a violinist positions fingers on the violin neck, as illustrated in Figure 4. Nevertheless, the user can use it in any other preferred way. We are still conducting user studies of the violin-pad with initial results being very promising. In addition to providing fast text-entry, the violin-pad can be used together with the cache-watch for short-message inputs without the user having to take out the handheld.

3.3 Smart speech portal

The smart speech portal offers better audio quality (11K samples per sec. and 10 bits per sample) than the Bluetooth

Figure 4: Using the violin-pad

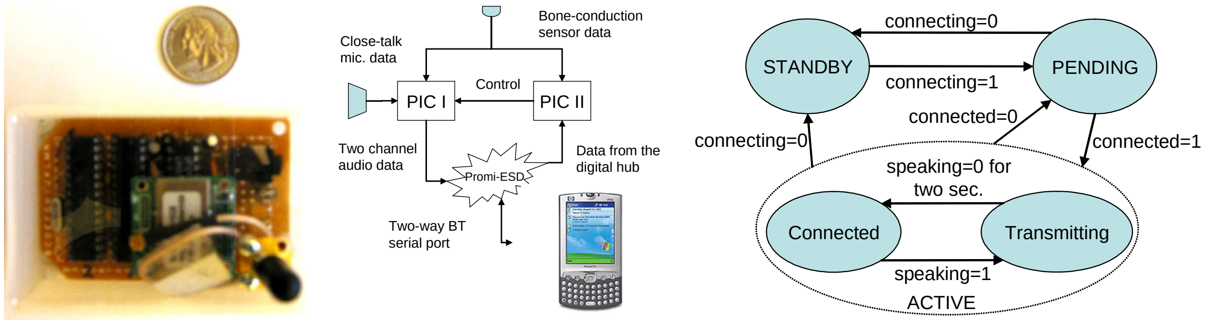
Headset Profile, and utilizes a bone-conduction sensor to provide a clean voice stream for recognition. Extreme care has been taken to pack speech and bone-conduction data into the 230$kbps$ Bluetooth serial port and implement power management using only minimal hardware and computing power.

Hardware design: Figure 5(a) shows a prototype of the smart speech portal. Note that the close-talk microphone and the bone-conduction sensor are plug-ins not shown in this picture. The prototype consists of two Microchip PIC-16LF873 (PIC I and PIC II), one Promi-ESD^{TM} class I module, and one operational amplifier module (National LM6134). A simplified block diagram is offered in Figure 5(b). The prototype draws about 33mW when connected but the user is not talking, and about 125mW when the user is talking.

Software design: PIC II is in charge of speech detection [11] using bone-conduction sensing, receiving data from the handheld and power management. PIC II informs PIC I whether the user is talking or not through the *speaking* control signal. When the Bluetooth is disconnected and the user has been talking for 2 seconds, PIC II raises the *connecting* control signal, asking PIC I to seek a connection with the handheld. When Bluetooth is connected and the user has remained silent for 10 seconds, it lowers the connecting signal, asking PIC I to disconnect. PIC II convey the Bluetooth status information to PIC I via the *connected* control signal. It is also in charge of receiving and interpreting data from the handheld via the Bluetooth module.

PIC I is devoted to packing data, sending them to the handheld, and managing Bluetooth according to control signals from PIC II. Software on PIC I is best illustrated by Figure 5(c). When the user is in a conversation, PIC I is in the ACTIVE mode. If the speaking signal is high, it starts analog-to-digital conversion and data coding/transfer immediately (*Transmitting* sub-mode); if the speaking signal is low for 2 seconds, it stops data transfer to the Bluetooth module (*Connected* sub-mode), which drastically reduces the power consumption. Since the data from the bone-conduction sensor can be used to suppress noise [17], they are also sent to the handheld for further processing. Since the bone-conduction signal is low-pass filtered, it is sampled at a rate half of that for the close-talk microphone. PIC I utilizes a simple differential speech coding algorithm to reduce data rate and power consumption.

The device manager on the handheld decodes the data into a continuous voice stream, which can be either recorded or supplied to a speech recognizer. With the prototype, a user can connect the smart speech portal to the handheld by talking for a few seconds. He/she can disconnect by either remaining silent for a while or simply uttering the corresponding voice command. While the smart speech portal is



(a) Prototype (b) Data and control flow (c) State-machine description for PIC I software

Figure 5: The smart speech portal

connected to the handheld, the user can access the handheld's computing power in a hands-free fashion through robust speech recognition. As voice is an integral part of personal life, the smart speech portal can also be employed for digital diary-like applications [4]. Since it only records the data when the user is talking, it automatically solves the privacy and data volume problem that many digital diary recording devices may face.

4. RELATED WORKS

An excellent survey of wrist-worn devices is offered in [12]. Contrary to our minimalist approach, Microsoft's SPOT watch [10] and IBM's Linux Watchpad [7] are designed to be a computing system, feature-rich and power-hungry, which severely limits their battery life and user acceptance [5]. The TiltType wrist device in [3] communicates with a PC through a Berkeley Dot Mote module and could display text messages from the computer using an XML-like protocol. However, it was primarily an input device. Starner offers an excellent survey in [14] on text entry on mobile devices. Popular tex entry methods, such as Multi-tap (used on cell phones), Thumbscript [15] and Fastap [2], share the same two problems. First, they are physically attached to the host device. The user has to hold the handheld to type. Second, the user can at most employ two fingers for typing. The design of Twiddler [9, 16] looks similar to the violin-pad. However, chording, which is key to Twiddler, imposes a significant memory load on its user. Sensors and bone-vibration have long been used for enhancing speech quality and improving speech recognition [6, 17]. A power-hungry wireless microphones and stereo transmitters are used in the Nomadic Radio system [13], a wireless LAN-based feature-rich audio interface. The Spartan BodyNet [3] is a Berkeley Dot Mote-based low data-rate wireless PAN with three very limited interfacing devices. Other than the TiltType wrist watch, there are a notification ring that has two LEDs to attract user attention, and a phicon whose presence would authorize a certain level of functionality from the PAN.

5. CONCLUSIONS

We envision that through small low-power wireless interfacing devices, the user interacts with a handheld effortlessly and the handheld captures information naturally anytime and anywhere, which we call pervasive interfacing. We believe that developing a pervasive interfacing system is critical to the success of pervasive computing. In this work, we discussed the design and prototype of a handheld-centered PAN of low-power wireless interfacing devices as our first step toward pervasive interfacing. The network is centrally managed by the handheld and interfacing devices can join as add-ons. Three add-ons have been designed and prototyped for this purpose in this work. The violin-pad and the cache-watch provide the handheld the two basic means of interfacing, text-entry and display, at the same time. The smart speech portal provides a hands-free speech portal to support the maturing speech recognition technology. With these devices, the user does not have to physically access or operate the handheld to utilize its computing power, unless there is a need for the larger display. The PAN also provides a platform on which new services, such as personal digital diary and continuous personal health monitoring, can be based.

6. REFERENCES

[1] BTAccess library. *http://www.high-point.com.*

[2] Fastap. *http://www.digitwireless.com/.*

[3] K. P. Fishkin, K. Partridge, and S. Chatterjee. User interface components for lightweight WPANs. *IEEE Pervasive Computing Magazine*, (4), Oct.-Dec. 2002.

[4] J. Gemmell, G. Bell, R. Lueder, S. Drucker, and C. Wong. MyLifeBits: Fulfilling the Memex vision. In *Proc. ACM Int. Conf. Multimedia*, pages 235–238, Nov. 2002.

[5] H. Goldstein. A dog named SPOT. *IEEE Spectrum*, 41(1):72–73, Jan. 2004.

[6] M. Graciarena, H. F. H., K. Sonmez, and H. H. Bratt. Combining standard and throat microphones for robust speech recognition. *IEEE Signal Processing Letters*, 10(3):72–74, Mar. 2003.

[7] IBM Linux watch. *http://www.research.ibm.com/WearableComputing.*

[8] Initium Promi-ESD^{TM}. *http://www.initium.co.kr/english/promi-esd.html.*

[9] K. Lyons, T. Starner, D. Plaisted, J. Fusia, A. Lyons, A. Drew, and E. W. Looney. Twiddler typing: One-handed chording text entry for mobile phones. In *Proc. Conf. Human Factors in Computing Systems*, pages 671–678, Apr. 2004.

[10] Microsoft SPOT. *http://www.spot-watch.com/.*

[11] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition.* Prentice-Hall, Inc., 1993.

[12] M. T. Raghunath and C. Narayanaswami. User interfaces for applications on a wrist watch. *Personal and Ubiquitous Computing*, 6(1):17–30, Feb. 2002.

[13] N. Sawhney and C. Schmandt. Speaking and listening on the run: Design for wearable audio computing. In *Proc. IEEE Int. Symp. Wearable Computers*, page 108, Oct. 1998.

[14] T. Starner. Keyboards redux: Fast mobile text entry. *IEEE Pervasive Computing*, (3):97–101, Jul.-Sept. 2004.

[15] Thumbscript. *http://www.thumbscript.com.*

[16] Twiddler. *http://www.handykey.com.*

[17] Y. Zheng, Z. Liu, Z. Zhang, M. Sinclair, J. Droppo, L. Deng, A. A. Acero, and X. Huang. Air-and bone-conductive integrated microphones for robust speech detection and enhancement. In *Proc. IEEE Wkshp. Automatic Speech Recognition & Understanding*, pages 249–254, Nov.-Dec. 2003.