The Wearable Computer as a Personal Station

Jin Ho Yoo [1], and Sang Ho Lee [2]

[1] Wearable Computing Research Team, ETRI,
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea
yoojh@etri.re.kr
[2] Dept. of Computer Science, Chungbuk National University,
48 Gaesin-dong, Cheongju, 361-763, Korea
shlee@chungbuk.ac.kr

Abstract. This paper introduces a wearable computer as a wearable personal station. We propose and implement wristwatch-style wearable personal station and its I/O devices in this paper. Nowadays the progress in miniaturizing more powerful computer systems and the availability of various devices around wearable computing will bring this technology to the edge of a new quality. Wearable computing is starting to become a product by itself. The function components of our wristwatch-style wearable computer include watch, PDA functions such as PIMS, address book, portable multimedia features, personal communication features and so on. We miniaturized these functionalities into our wearable small device. We also handle a study of USB implementation without wires for human interfaces. This paper describes a platform, hardware specifications and applications we have developed.

1 Introduction

Wearable computing becomes more and more feasible and receives growing attention throughout industry and the consumer marketplaces. Nowadays the progress in miniaturizing more powerful computer systems and the availability of various devices around wearable computing(e.g., wearable computers, high resolution Head Mounted Displays, interaction devices) will bring this technology to the edge of a new quality[4]. Wearable computing is starting to become a product by itself. Basic research is accompanied by an increasing amount of application and transfer related research.

This paper introduces our Wearable Personal Station named WPS as a wearable computer system. We propose and implement wristwatch-style wearable personal station in this paper. Because people generally keep watches on their wrists, watches are less likely to be misplaced compared to phones and pagers. For example a hip holster is not the best place to keep a cellular phone while sitting in a car and so people tend to keep them in the car seat and forget them when they leave the car in the parking lot[4].

Surely the watch form factor requires a relatively small screen size, and there is not much room for input devices or batteries. The value of a wristwatch platform depends on finding good solutions to these issues. To interact with the watch, we need both hands since the hand on which the watch is worn is practically useless for controlling input devices on the watch. This wearable personal station plays a role as electronic secretary, network gateway. Wearable personal station is attached on human body and provides services.

2 System Overview

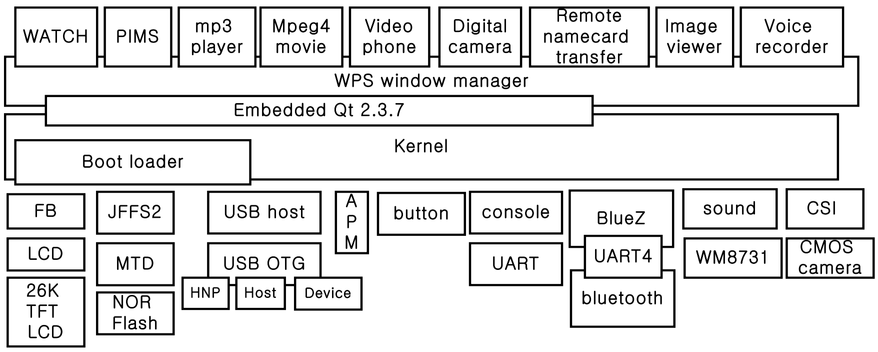

Fig. 1. The System Overview of Wearable Personal Station

Wearable system is a computer system. So wearable system includes several hardware devices, software components, network devices and so on. Fig. 1 shows the system overview of our wearable personal station.

3 Hardware Specification and An Outward Appearance

3.1 The Main Processor

We adopted DragonBall MX, the MC9328MX21(i.MX21) as our main processor[1]. The Dragon family of microprocessors has demonstrated leadership in the portable handheld market. Following on the success of the DragonBall MX(Media eXtensions) series, the MC9328MX21(i.MX21) provides a leap in performance with an ARM926EJ-S^{TM} microprocessor core that provides native security and accelerated java support in addition to highly integrated system functions. The i.MX21 processor features the advanced and power-efficient ARM926EJ-S^{TM} core operating at speeds up to 266MHz. We tried operating at speed up to 296 MHz. On-chip modules such as an MPEG-4 codec, LCD controller, USB OTG, CMOS sensor interface, and an AC97 host controller offer designers a rich suite of peripherals that can enhance any product seeking to provide a rich multimedia experience.

3.2 The Multimedia Modules and Display for the Wearable Personal Station

We use eMMA for multimedia function implementation in i.MX21. eMMA means the enhanced MultiMedia Accelerator which consists of the video Pre-processor(PrP), Encoder(ENC), Decoder(DEC) and Post-processor(PP). These blocks work together to provide video acceleration and off-load the CPU from computation intensive tasks. While the encoder and decoder support only MPEG4-SVIP, the PrP and PP can be used for generic video pre and post processing such as scaling, resizing, and color space conversions.

The features of eMMA support private DMA between CMOS sensor interface module and pre-processor as data input and image scaling function. The eMMA encoder supports MPEG-4 and H.263, full conformance to ISO/IEC 14496-2 Visual Simple Profiles Levels 0 to 3, real-time encoding images of sizes from 32x32 up to CIF at 30 fps and camera stabilization. The eMMA decoder supports MPEG-4 and H.263, full conformance to ISO/IEC 14496-2 Visual Simple Profiles Levels 0 to 3, real-time encoding images of sizes from 32x32 up to CIF at 30 fps and camera stabilization. Post-processor receives the input data of YUV 4:2:0(IYUV, YV12) from system memory and does image resizing with upscaling ranging from 1:1 to 1:4 and downscaling ranging from 1:1 to 2:1 in fractional steps. This ratios provide scaling between QCIF, CIF, QVGA(320x240,240x320).

The Liquid Crystal Display Controller is embedded in the i.MX21 chipset. The Liquid Crystal Display Controller (LCDC) provides display data for external gray-scale or color LCD panels. The LCDC is capable of supporting black-and-white, gray-scale, passive-matrix color (passive color or CSTN), and active-matrix color (active color or TFT) LCD panels.

Camera module in the wearable personal station is controlled by CMOS sensor interface(CSI) characterized by CSI architecture, operation principles, and programming model. The CSI enables the i.MX21 to connect directly to external CMOS image sensors. CMOS image sensors are separated into two classes, dumb and smart. Dumb sensors are those that support only traditional sensor timing (Vertical SYNC and Horizontal SYNC) and output only Bayer and statistics data, while smart sensors support CCIR656 video decoder formats and perform additional processing of the image (for example, image compression, image pre-filtering, and various data output formats).

3.3 Multimedia Card/Secure Digital Host Controller

The MultiMediaCard (MMC) is a universal low cost data storage and communication media that is designed to cover a wide area of applications as electronic toys, organizers, PDAs and smart phones and so on. The MMC communication is based on an advanced 7 pin serial bus designed to operate in a low voltage range, at medium speed.

The Secure Digital Card (SD) is an evolution of MMC with an additional 2 pins as the same form factor and is specifically designed to meet the security, capacity, performance, and environment requirements inherent in newly emerging audio and video consumer electronic devices. An SD card can be categorized as SD Memory or SD

I/O card, commonly known as SDIO. The SDIO card provides high-speed data I/O with low-power consumption for mobile electronic devices.

The Multimedia Card/Secure Digital Host module (MMC/SD) integrates MMC support with SD memory and I/O functions. We also mainly used MMC/SD as storage device. We newly made MMC/SD device driver for its use.

3.4 The Battery and An Outward Appearance

We have to get ultra slim battery for our wrist worn device. We need pocketing and free-stacking technology. This battery is lithium-ion rechargeable batteries tested as high temperature storage test (90 °C, 4hr), humidity test (60 °C, 90% RH, 1week), thermal shock test(-40 °C/60 °C, 10 cycles), safety test such as hot-box test (150 °C, 10min), nail penetration test, short circuit test, overcharge test (3C continuous overcharge) and long period storage test(25°C, 60°C, 80°C)

The outward appearance of our WPS hardware is as follows.

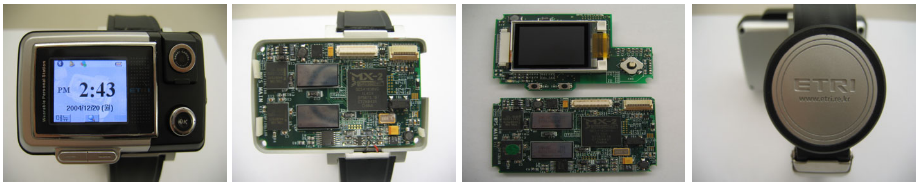

Fig. 2. Front Side, Front Inside, The PCBs, The Battery

4 System Software Specification

4.1 Operating System

We adopted linux operating system for free because open source. We patch linux-2.4.20 with mx2bsp patch. For building operating system, we needed to build a cross-development environment, including cross compilers, assemblers, and binary utilities that would let us generate code for the ARM architecture on our general purpose personal computer. Although there were many resources on the internet to help in this process, it took some time to get everything set up right. We found relevant pieces of source code and patches on the internet. We wrote some of the basic device drivers and modified the memory maps for our hardware configurations. In addition, we wrote some device drivers not supported directly. We used busybox and its init profile on jffs2 filesystem/mtd. We wrote newly mtd device driver for building jffs2 filesystem. We used NOR flash, 512 bytes erase block. We builded and used MMC/SD in evaluation phase. We mainly used MMC/SD, USB memory stick as the secondary memory device for transferring its applications and data. We adopted embedded Qt 2.3.7 as graphical user interface. We made WPS window manager based on embedded Qt 2.3.7. We also implement advanced power management.

4.2 Device Drivers

The CMOS camera is supported by CSI device driver. It supports for 352x288 or 640x480 pixel array, 30 frame/sec, CIF, QCIF, QQCIF or VGA, QVGA, enabling sub-sampling, still image capture.

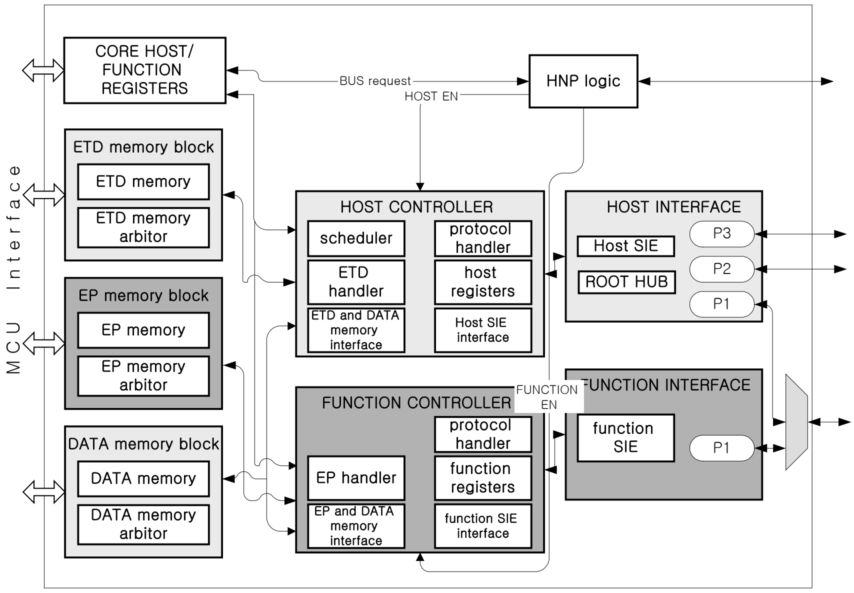

Fig. 3. The USB OTG Block Diagram

LCD driver supports 1.7 inch, 128x128 resolution TFT-LCD. We adopted 26K TFT-LCD and used frame buffer display. We wote MTD device driver newly for jffs2 filesystem. The flash device is interleaved by 2 and supports MTD. We used the flash chip 28F256 of intel.

The APM supports normal mode, doze mode and sleep mode with system software. The normal mode means the main processor and all devices are power on state. The doze mode means the main processor is power off state and all devices are power on state. The sleep mode means the main processor and all devices are power off state. This APM supports frequency scaling, not voltage scaling. The WPS consumes 113 mA in the normal mode, 85 mA in the doze mode, 35 mA in the sleep mode. We are trying to use frequency scaling for power scaling. The button is a simple device driver using interrupt service. We configure and make WM8731 device driver for supporting the sound.

Many of these portable devices would benefit from being able to communicate to each other over the USB interface, yet certain aspects of USB make this difficult to achieve. Specially, USB communication can only take place between a host and a peripheral. So OTG supplement to the USB is needed. The USB OTG block diagram is shown in Fig. 3[1]. We implement USBOTG device driver on the above blocks.

4.3 Making USB Wireless

USB continues to evolve as new technologies and products come to market. USB has already many application interfaces and support versatile device interfaces. It's already the de facto interconnect for PCs, and has proliferated into consumer electronics(CE) and mobile devices as well. Like this, USB has built on many killer applications, many CE devices, many interfaces.

The growing use of wireless technology in PC, CE and mobile communications products, along with the convergence of product functionalities, calls for a common wireless interconnections. If our devices is unwired, it solves the tangle of wires. We want to use Legacy USB functionalities, portabilities, multimedia capabilities with wireless interconnection.

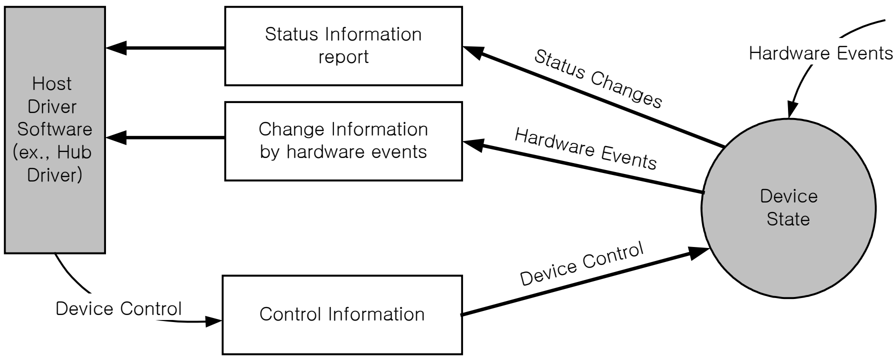

Fig. 4. Relationship of Status, Status Change, and Control Information to Device States

Making USB HUB Wireless. We will make the hub function of USB cordless, so it will connect host with devices without wires. The USB root hub is special device to be able to connect to another devices. We will make USB hub and USB host interface device wireless. There are many status report registers and events in hub. These are implemented by software modules and functions in the device driver level. And the memory uses and several data structures of USB host are also implemented by software modules and functions in the device driver level.

A hub is a standardized type of USB device. A hub is the only USB device that provides connection points for additional USB devices. The root hub is the hub that is an embedded part of every host controller. From an end user perspective, a hub provides the sockets that are used to plug in other USB devices. For making hub wireless, it needs to know the architecture requirements for the USB hub. It contains a description of the three principal sub-blocks: the hub repeater, the hub controller, and the transaction translator. There are major aspects of USB functionality that hubs must support: connectivity behavior, power management, device connect/disconnect detection, bus fault detection and recovery, high-speed, full-speed, and low-speed device support.

In case making USB wireless, it must support the above functionalities. Moreover, It needs the data structure and the resources for host functionalities, e.g. ETD, data memory.

Fig. 4 shows how status, status change, and control information relate to device states[2]. Hub or port status change bits can be set because of hardware or software events. When set, these bits remain set until cleared. While a change bit is set, the hub continues to report a status change when polled until all change bits have been cleared by the USB system software.

The Convergence of IP Side and USB Side on Device Driver Level. This is software issues and the process of device emulation or device adaption. In here wireless media MAC/PHY may become 802.11a/b/g or UWB. We are going to make USB wireless for using legacy LAN and USB applications. When wireless USB device accesses wireless USB host, wireless USB host operates same as wire USB device plugs in wire USB host.

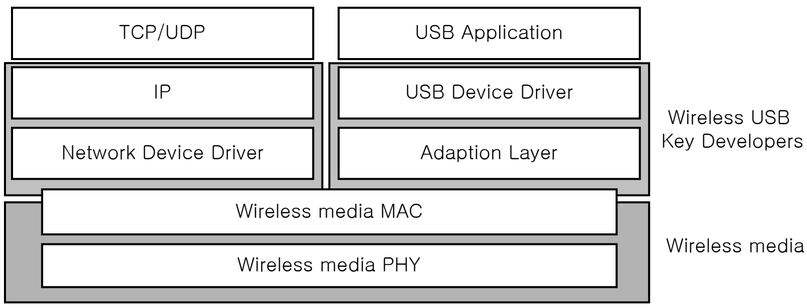

Fig. 5. The Stack of the Wireless Platform with IP

Fig. 5 shows the convergence of IP side and USB side. It will occurrs convergence on wireless media MAC level. The inputed packets should be divided into two parts, IP side and USB side. For that, USB packet should be defined as the special MAC packet. It recognizes whether IP side packet or USB device packet through checking the inputed packets in wireless media MAC device driver. The USB adaption layer has the data structure for USB hub and USB host functionalites. So USB device driver thinks USB adaption layer as USB real device. USB device driver drives USB adaption layer as the legacy USB device driver drives USB real device. USB adaption layer communicates the counterpart USB adaption layer of USB device through wireless media MAC/PHY transparently. Communication of wireless media MAC/PHY should meet timeliness and bandwidth of USB packet. For example, event, status report, interrupt, command for four mode, isochronous mode, bulk mode, interrupt mode, control mode. We should prepare software emulation functionalites for these implementations. This implementation provides the convenient and the effective on device usage with maintaining the legacy devices.

There are many candidates of wireless media. For example, UWB, 802.11a/b/g, Bluetooth. UWB among those candidates is high-powered because low-power, high bandwidth. UWB and the associated networking protocol efforts are in the early stages of development, and several key deployment scenarios are being defined and evaluated. UWB complements currently deployed wireless networks in the WLAN environment, plus it extends high bit-rate, multimedia connectivity to WPANs supporting PC and CE devices.

Many UWB components and systems are already in the testing and demonstration phases, with actual release dates for final consumer products expected in early 2005. We discussed software emulation for making USB wireless, However, This wireless USB will be integrated into one chip solution not as software emulation. When wireless USB device accesses wireless USB host, wireless USB host operates same as wire USB device plugs in wire USB host.

4.4 Bluetooth Protocol Stack

We use the BlueZ bluetooth stack downloaded from internet[3]. The BlueZ has several device drivers, VHCI driver, UART driver, USB driver and so forth on the Bluetooth hardware. The BlueZ utilities and bluetooth applications use HCI sockets, L2CAP protocol, SCO sockets through the protocol interface of the BlueZ core driving VHCI driver, UART driver, USB driver and so on. The blueZ core also manages HCI life cycle and interfaces with the drivers of HCI UART, HCI USB, HCI PCMCIA. The blueZ manages the protocol of L2CAP, LMC/LC, RFCOMM, BNEP, CMTP, HIDP and SDP.

4.5 Videophone

The Wearable Personal Station has MPEG-4/H.263 codec. The video/audio encoding engine receives the MPEG-4 from MPEG-4/H.263 encoder pre-processing module and AAC from AAC encoder. This MPEG-4/H.263 encoder pre-processing module receives YUV from CSI and makes MPEG-4 stream. The video/audio decoding engine makes the AAC output for AAC decoder and makes MPEG-4 for MPEG-4/H.263 decoder post-processing module. LCDC receives RGB data from MPEG-4/H.263 decoder post-processing module and display on a screen.

4.6 Root Filesystem

The root file system is the jffs2 journalling file system on the MTD device for fast recovery. Actually the root file system is tested by system crash situation. The root file system includes / as a top directory, /bin as a directory for utility binary files, /sbin as a directory for system binary files, /dev as a directory for special device files, /drivers as a directory for dynamic kernel module files needed by kernel, /etc as a directory for configuration files, /lib as a directory for several library files, /mnt as a mount directory, /proc as virtual directory for system information, /tmp, /var as a directory for temporary files, /root as a root home directory, /usr as a system user directory. /usr/local as our special libraries and our special binaries.

5 Applications and User Interface

Fig. 6. MP3 Player, Digital Camera, Videophone, Namecard Transfer, MPEG-4 Player, Voice Recoder, Eval B'd with HMD, HMD output and Control Panel

Our applications have watch function, PIMS, multimedia player, videophone, digital camera, namecard transfer and so on. Watch function includes time display, calendar display, alarm, time/date setting. PIMS means personal information management system which has address book, personal plan management, memo, electronic postit, electronic calculator and so on. Multimedia player plays mp3 files, MPEG-4 movie files. Videophone can do visual communication on WPS LCD display or HMD(Head Mounted Display, VGA). For these applications, we provide several intuitive menu screen displays for graphic user interface as shown in Fig. 6.

6 Conclusions

Personal computer(PC) has been evolved from desktop to portable PC such as tablet PC and PDA. Technology innovation on semiconductor have made it possible to package a reasonably powerful processor and memory subsystem with advanced input/output devices. Wearable computing becomes more and more feasible and receives growing attention throughout industry and the consumer marketplaces.

Our hardware technology enough to support many software applications and minimize its size. Our developing issue is a wearable personal station as a wearable personal platform with advanced I/O devices in the ubiquitous world. This wearable personal station will interoperate with the ubiquitous objects using network and I/O device.

Our work has convinced us that a lot of functionality can be packaged into small devices. We combine high-function miniature hardware with a robust operating system and well designed applications with intuitive user interfaces and so the wearable personal station forms the basis of personal station platform for the ubiquitous world.

References

1. Freescale,: i.MX21 application processor specification.
2. USB Implementers Forum, inc.,: On-The-Go Supplement to the USB 2.0 Specification.
3. BlueZ, http://www.bluez.org.
4. Chandra Narayanaswami, M. T. Raghunath,: Application design for a smart watch with a high resolution display, Fourth International Symposium on Wearable Computers, Oct 18-21, (2000)