Computers that you wear help your work and play

Introduction

The pace of electronics miniaturization has created a growing interest in personal devices that can be worn to monitor activity, provide user notifications and store and transmit information to remote servers. Wearable devices such as smart watches can provide message notifications and act as a remote display for tablets or smartphones while the main product is carried in a pocket or stored in a backpack. Other wearable devices can be used to measure fitness and provide workout data while communicating with heart-rate monitors or pedometers. Applications providing navigation or location information can be used for sporting or recreational activities like sailing or hiking. Personal healthcare sensors can monitor and record patient wellbeing, uploading diagnostic data to remote servers and providing alerts after critical events such as a fall.

Hardware and software designers are able to take advantage of the increased performance and integration of microprocessors and flash-based microcontrollers to develop the next generation of portable or wearable devices. These processors are at the heart of the products that communicate wirelessly with smartphones, PCs and tablets while measuring and storing application data.

Designers face many challenges in implementing effective wearable devices, including:

- Maximizing battery life and time between battery charges.
- Minimizing the physical dimensions and weight of the product.
- Implementing a range of wireless protocols to communicate with external devices.
- Receiving, interpreting and storing data from local and remote sensors.
- Providing an intuitive user interface.

Maximizing battery life

Wearable devices monitor sensors and communicate through wired and wireless connections while periodically updating a display and responding to user input. The device enters a low power state to preserve battery life at other times. The designer selects components that can efficiently communicate and process sensor inputs while awake, and draw very little current from the battery while asleep. Battery size and life are connected – the larger the battery, the larger the charge it can hold and the longer the user can go without recharging the product. However, most wearable devices need to be as small and light as possible, forcing the designer to minimize both active and standby power consumption.

Minimizing physical dimensions

Form factor is very important when designs are worn round the neck or on the wrist. End products need to be small and light, forcing the circuit board to be as small as possible with high levels of device integration. As previously noted, size or thickness constraints often force designers to select a battery with lower capacity, increasing the need for a power effective design.

Implementing wireless protocols

Many different wireless devices and protocols are used in wearable computing. Designers select protocols based on throughput, power consumption and solution cost.

- Devices that communicate with smartwatches or tablets can use 802.11 Wi-Fi through an access point or Wi-Fi Direct. Wi-Fi Direct extends the 802.11 standard by allowing devices to form ad hoc networks, allowing a watch to directly communicate with a tablet without requiring an access point. 802.11 connections typically offer the highest bandwidth in the system.
- Bluetooth is used in many medical and fitness sensors, as well as cellphones and laptops, allowing sensor networks to communicate with a wearable device. As Wi-Fi Direct becomes more

popular, designers can select between Wi-Fi Direct and Bluetooth Low Energy depending on overall system requirements and power consumption.

- The IEEE 802.15.4 wireless standard is used in Zigbee, 6LoWPAN and can be used for very low power, lower data rate communication. 802.15.4-based networks typically support mesh networking, which can be used to extend communication reliability and range.

The only common wired interface commonly used by wearable devices is USB when uploading data to a PC, updating firmware and recharging the battery.

Receiving, interpreting and storing data from local and remote sensors.

Sensors measure real-world signals such as rotation or light intensity and translate the measurement into an electrical format. Local sensors communicate with the processor by analog or digital channels (such as I2C). Remote sensors use wireless protocols. Once the raw sensor data is received, the microcontroller processes the data, often combining inputs from multiple sensors to enhance the accuracy of parameters such as compass headings.

Many sensors measure relatively slowly changing signals, allowing the device to save power by sleeping between each sample. This requires the device to wake up quickly when the next sample is due. Sample processing efficiency is determined by the computational performance of the microcontroller. 32-bit performance, hardware multiply and divide units, and a Floating Point Unit may all be important based on sensor type and application.

Providing an intuitive user interface

Wearable devices typically often show data like time, measurements, maps or messages on a display. Displays range from simple, low cost monochromatic segment displays up to full color TFT LCDs depending on application and cost. Users can interact with the device through mechanical buttons or through touch interfaces. Capacitive interfaces provide a very intuitive means of interfacing with a wearable device. The designer can place buttons or sliders on a bezel or the side of the product, or even add a touch sensor over the display to support multi-touch interaction.

Wearable Computing with Atmel Microcontrollers and processors

Atmel provides a broad range of wearable computing system solutions based on microcontrollers and microprocessors.

- Atmel microcontrollers and microprocessors are designed for ultra-low power consumption in active and standby modes. Atmel's EventSystem™ with SleepWalking™ allows peripherals to automatically connect with each other even in ultra low power modes, greatly simplifying sensor interfacing and further improving power consumption. Wakeup times are minimized to allow low power modes to be used without fear of missing communications data or sensor events.
- Atmel devices integrate many features to save circuit board space, such as USB transceivers and embedded termination resistors. Many devices are offered in very small form factor packages.
- The Atmel Software Framework (ASF) contains communications libraries to support external Wi-Fi and Bluetooth radios, mesh and point-to-point networking on Atmel's 802.15.4/Zigbee AT86RF radios and a full range of USB drivers.
- The Atmel Software Framework (ASF) contains libraries and driver functions for many popular third-party sensors such as accelerometers, gyroscopes and magnetometers. Atmel microcontrollers contain core functions, such as hardware multiply and divide, and peripheral functions, such as the EventSystem™, that support effective sensor processing.
- Atmel microcontrollers and microprocessors that directly drive TFT or segment LCD are available. All device families support parallel or serial interface displays.
- Atmel is a leading manufacturer of capacitive touch solutions. Standalone controllers support off-the-shelf capacitive buttons, sliders and wheel (BSW) implementations. All Atmel microcontrollers can directly manage capacitive buttons through Atmel-provided software libraries. Atmel also offers the maXTouch series of capacitive touchscreen controllers that directly manage optically clear touch sensors overlaid on LCD displays. Atmel's touch solutions may be tuned to function when moisture is present, which is often a key requirement for wearable applications.

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**