Sensor-Driven Computing Comes of Age

Roy Want

Sensing has always been an integral part of any computer system; after all, a keyboard is just a collection of finger-sized pressure-sensors conveying user commands to the operating system. Any technology that mediates data from the outside world to a computer system's internals can be considered a sensor. In the '80s, the mouse became a popular PC peripheral—an important technology that enabled the now-universal WIMP interface—and it's fundamentally a 2D linear-motion sensor. Microphones integrated with PCs to support multimedia and communications applications in the '90s are actually sensitive pressure sensors with a frequency response in the human audio range. Likewise, cameras that support video telephony, digital photography, and image processing applications are an array of sensitive light sensors. This list represents an increasingly complex set of sensors that have been added to our computing platforms over time. Fundamentally, accurately measuring complex analog physical quantities requires significant processing power and real-time data manipulation. Microprocessor performance gains, which have enabled an increasingly complex set of PC applications for more than 25 years, have also enabled the integration of an increasingly sophisticated set of sensors into the modern computer platform.

INTERFACING SENSORS 101

Engineers most commonly use the term *sensor* in the context of measuring analog physical parameters such as temperature, pressure, and light level. Any measured value must also be

> **Accurately measuring complex analog physical quantities requires significant processing power and real-time data manipulation.**

recorded, and although analog recording techniques are available, the most flexible recording and processing tool is the computer. Therefore, to exploit its capabilities, we must convert the analog parameter into a digital representation. Using an analog-to-digital converter (ADC)—a system component that can convert a potential difference across its input into a corresponding binary number—is the basis of such interfaces. Most sensors are constructed from materials that vary their electrical characteristics in response to changes in the physical parameter being measured. With some creative electronics, engineers can convert this variation into a proportional change in potential difference and feed it to an ADC.

Thus, a typical solution to a sensing problem requires a sensor, interface circuit, ADC, and connection to a processor's I/O port. If high accuracy is necessary, the ADC must be able to make the conversion with enough bits to represent the measurement and use a process that's linear across the range of values being considered. A classic ADC design uses a network of resistors (sometimes referred to as a *resistor ladder*) as an integral part of the conversion mechanism and, to satisfy the accuracy requirements, the resistive components must be machined within a small tolerance. These considerations increase implementation cost, which limits the market size for mass-produced products. As a result, many modern sensor systems convey their parameters to processing components through signals that use *time* instead of *potential*. They do this by generating a square wave signal and changing its frequency, or duty cycle, in proportion to the physical parameter. From the processor's perspective, analog-to-digital conversion becomes a new type of problem: measuring the time it takes for a signal to transition between a logic-zero and a logic-one. With the development of modern microcontrollers based on clock rates ranging from 10 MHz to 400 MHz and derived from high-accuracy crystal oscillators, this becomes relatively straightforward.

MISSION STATEMENT: *IEEE Pervasive Computing* is a catalyst for advancing research and practice in mobile and ubiquitous computing. It is the premier publishing forum for peer-reviewed articles, industry news, surveys, and tutorials for a broad, multidisciplinary community.

Furthermore, quartz crystal oscillators are very stable, so sensor readings can be made with corresponding accuracy.

BUS STANDARDS

The standardization of serial interconnect buses, such as Philips' Inter-Integrated Circuit (I2C) bus and Motorola's Serial Peripheral Interface (SPI), has been important to sensor interfaces' design. Although engineers originally designed them to reduce the number of multi-wire buses needed to integrate system components, they have also used serial interconnect buses as a general-purpose mechanism for interfacing sensors. The I2C and SPI standards turned out to be a convenient way for sensor manufacturers to integrate physical sensors, interface electronics, and ADC, and then use the interconnect bus as the standardized interface. Microcontroller manufacturers will often include hardware support for these standards so, when programmers need to write system code, they can reuse standard libraries for the basic data-capture operation, thus reducing development time. In the case of I2C, which supports preset device addresses and multiple bus masters, several sensors can be supported by the same bus.

Although I2C, SPI, and several other less pervasive standards have resulted in better system designs, no single bus standard provides for all the features that a sensor manufacturer could wish for. For instance, neither I2C nor SPI provide a standard mechanism for power delivery or high-level control of the sensing process (for example, sample rate and filtering). There's still an opportunity for a visionary manufacturer to step up and create an all-encompassing bus standard that enables the design of sophisticated sensor interfaces for a wide range of applications.

SENSOR NETWORKS

In recent years, computer scientists have given considerable attention to sensor networks, an ad hoc collection of wireless processing nodes that can both capture local sensor data and transfer it hop-by-hop to a centralized processing component. When the complete context of the data is available in one place, scientists can endeavor to understand the processes that generated it—for example, to make predictions about the spread of pollution or to forecast weather more accurately. In this special issue, the guest editors' introduction describes the opportunity for integrating sensor networks with global networks to create a whole new class of application. However, sensors can also play an important role in customizing mobile devices and their applications on the basis of context. This is particularly important because the now-extensive cell phone market is producing high-end handsets with significant processing capabilities that can exploit sensing— and this trend will likely percolate into the lower end of the market over time.

CONTEXT-AWARE OPERATION

Mobile computers, by the very fact they can move, change their context of use. This property potentially lets software developers make applications easier to use. It can then show data that's customized to the current context and hide command choices that are no longer relevant to the current situation, reducing clutter on the display. This type of automatic customization is often called *context-aware* operation. Developers can differentiate and improve mobile computer operation over traditional desktop machines through the use of context, but no general standards exist to do this. In general, we can divide contextual data into two categories—information derived from the platform in isolation or with the help of a cooperating infrastructure.

Developers could obtain platform-based context from sensors measuring parameters that don't require any infrastructure for a computer to exploit it; examples are acceleration, pressure, and magnetic field angle (electronic compass). In fact, these sensor peripherals complement existing platform sensors, enabling the computer to sense its

FROM THE EDITOR IN CHIEF

surroundings in greater detail. However, it's the mobility of a modern platform as used today that makes this type of sensor relevant. An engineer would have no reason to design an accelerometer into a classic desktop computer because it doesn't move, but could use one in a mobile computer to determine its tilt angle, whether it's shaken, and whether it's moving. Furthermore, analyzing the data can provide important supplementary information—for example, a rapid acceleration to a constant speed might indicate use in a car rather than walking or at a desk—and can be used to infer about a user's activity in general.

A good example of infrastructural context is location, which is typically obtained by sensing reference points in the surrounding environment. Developers can use location knowledge to create location-based services (LBS), which have been explored in detail in research organizations but still have many unrealized opportunities commercially. LBS can derive location information from a GPS receiver that relies on signals transmitted from a collection of satellites in orbit, or detect the signal strength of cell phone towers, or detect the presence of nearby RFID tags through interrogation. The former has been widely exploited in automobile navigation systems, such as Hertz's Neverlost, and in handheld navigation devices, such as Garmin's eTrex. The latter provides an opportunity for a rich set of indoor location services and could be catalyzed in the future by the widespread use of RFID technologies employed for item-level tagging (see the Jan.–March 2006 *IEEE Pervasive Computing* special issue on RFID Technology).

MOBILE PRODUCT UIS BASED ON INTEGRATED SENSORS

Although researchers have implemented several mobile research platforms that use novel sensing capabilities to control the user interface, the platforms haven't been widely exploited commercially. An example is PARC's Hikari PDA, which incorporates a 2D accelerometer and GUI that let the user make selections simply by tilting it forward and back. However, general appreciation of the value of putting sensors in the user interface is not yet realized.

The recent announcement of Apple's iPhone emphasizes its sensing capabilities—it uses an accelerometer to detect orientation and adapts its applications to change between landscape and portrait modes. The iPhone can also sense if it's placed next to the operator's ear, turning off the display to extend battery life because it's no longer needed. The display will also dim in low ambient light to conserve power.

Apple has made a good name for its products by stylishly integrating capabilities that other companies have failed to capitalize on. If platform context based on integrated sensors turns out to be a driving success for the iPhone, we can expect a groundswell of companies launching new products with these capabilities. More commercial computer platforms incorporating sensors will give our research community a larger set of experimental opportunities and will likely generate a new wave of pervasive computing applications. Adding sensing capabilities to mobile computing platforms has never been easier, and accuracy and cost are no longer the limiting factors. **P**