Sensors Deliver New Opportunities for Ultrabook™ Software Innovation

The range of motion and location sensors built into Ultrabook™ devices[1] gives software makers a novel basis for user experience innovation. In addition to traditional input devices such as the keyboard and mouse, as well as the novel addition of touch, application developers can now gather a wide variety of inputs from the environment and user actions. By creating new usage models, software companies can use sensor data to extend existing product offerings, as well as to enable new ones.

Mainstream users have grown accustomed to sensor functionality on their smart phones that ranges from navigating using GPS to having the screen orientation change between portrait and landscape as they rotate the device. Such usages have been generally absent from PC-class applications, however, because of the relative rarity of sensors in laptops. With the addition of sensors to Ultrabooks, that is about to change. As the software industry begins to embrace sensor-based usages for the PC, Ultrabooks will drive opportunity from users who anticipate and expect innovation in this area.

To help developers build on the basic capabilities provided by hardware sensors, Intel and Microsoft have provided a significant body of documentation, guidance, tool support, and APIs. By applying these capabilities in novel ways, including manipulating sensor data and combining data from multiple sensors, software makers have begun to differentiate their products within their market segments, working to achieve a competitive advantage.

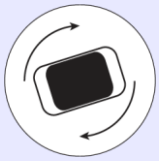
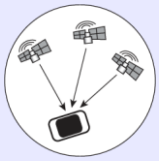
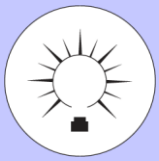
This article introduces the sensors built into the Ultrabook platform and illustrates how application architects and other decision makers at software companies can take advantage of those sensors to enable innovative usage models and product differentiation. It is part of a series of papers that explore key considerations in taking advantage of the Ultrabook market segment with software:

- **Harnessing the Ultrabook™ Experience to Sell More Applications:** http://intel.ly/selling-ultrabook-apps
- **Optimizing the Cost-Effectiveness of Software Development for Ultrabooks™:** http://intel.ly/reduce-dev-costs
- **The Winning Combination of Keyboard and Touch for Ultrabooks™:** http://intel.ly/keyboard-and-touch
- **Ultrabooks™ Broaden Software Usage Models on Touch-Capable Mobile Devices:** http://intel.ly/broad-usage-models
- **Sensors Deliver New Opportunities for Ultrabook™ Software Innovation:** http://intel.ly/sensors-for-software-innovation

Introducing the Sensors that Enable Innovation

A common set of sensors is recommended for Ultrabooks by Intel and supported by both Microsoft Windows* 8 and Windows 7, as shown in Table 1. These hardware devices and the software components that enable applications to take advantage of them comprise a rich set of building blocks for sensor-based software features. Together, they deliver the basis for “wow” factor in next-generation user experiences.

Table 1. First-generation sensors available in Ultrabooks.[1]

	**Magnetometers** measure the strength and/or direction of the earth's magnetic field. That data can be used to reckon the direction a device is pointing, which is useful not only for navigation, but also for implementations such as identifying landmarks in a specific direction from the user's vantage point.
	**Accelerometers** measure linear acceleration—the magnitude and direction of force being applied to the sensor, including gravity (as opposed to coordinate acceleration, which is change in velocity). That data is useful, for example, to change the device's display from portrait to landscape orientation when the user rotates the device.
	**Gyroscopes** measure orientation in space using principles of angular momentum. This data is particularly valuable in conjunction with other sensors, especially accelerometers and magnetometers, for more accurate measurements and adjustments for their inherent limitations.
	**Assisted GPS** identifies the geographic location of the device using satellite radio signals, just as conventional stand-alone GPS does, for use in navigation apps and related usages. Assisted GPS uses network-provided data about the positions of satellites to help reduce the time needed for the device to make an initial connection to the satellites.
	**Ambient Light Sensors** use photodetectors to detect the levels of light present in the immediate vicinity of the device. This data can be used to control the brightness of the device display, helping improve visibility in varying light conditions, for example, or to determine requirements for additional light in still or video photography.

Extended Capabilities through Sensor Fusion

As mentioned above, part of the value of sensor data to software innovation lies in extending those capabilities by combining the data from two or more sensors, a practice commonly referred to as "sensor fusion." Two common examples of sensor fusion are suggested in the description of the gyroscope in Table 1—namely combining data from the magnetometer, accelerometer, and gyroscope to overcome limitations of a single sensor on its own.

A common example of sensor fusion is to use data from a gyroscope to improve upon the accuracy of an accelerometer alone in sensing movement within three-dimensional space. This approach is common in smart phones and game console controllers that capture motion input. Another example is the use of data from a gyroscope together with magnetometer data for more accurate measurement of compass heading than would be possible with the magnetometer alone.

A related approach is the use of multiple sensors to cross-check one another, as a means of overcoming inherent limitations in each sensor individually. For example, consider the following limitations that exist for various approaches to establishing the geographic location of an Ultrabook device:

- **GPS** can be limited when the device is indoors, or because of atmospheric factors, or if the signals are blocked by buildings (including the "urban canyon" scenario) or trees.
- **Cell tower triangulation** can be compromised because of changing configuration by the carrier that changes tower ID information.
- **Wi-Fi triangulation** is subject to inaccuracies in the event that a Wi-Fi access point is moved.

- **IP address resolution** can produce inaccurate results because some IP usage scenarios may produce unexpected effects within wide-area networks.

Future Possibilities for Additional Sensors

The five sensors discussed here represent only a subset of the possible sensors that could be provided in future Ultrabooks. Additional sensors for quantities such as human proximity, atmospheric pressure, and humidity could enable future usage models, as well as improving existing ones with advances in areas such as power management, responsiveness, and security. Another area of interest is that existing sensors such as the temperature sensor commonly used to respond when a system goes outside a specified temperature range could be exposed to applications using APIs.

As data from more types of sensors becomes available to developers, the opportunity to differentiate applications based on novel usages—including those based on new sensor-fusion approaches—can be expected to develop further. That is, a broader range of sensors suggests that application innovation will continue and that more specialized implementations in software will become possible.

Sensor Usage Models Delivering Software Opportunity

Using sensors to enable devices to respond to environmental factors has been commonplace for some time, although it has mostly existed in background processes and other forms that don't necessarily gather a lot of attention from users. The temperature sensor example given above is one use case; another is that accelerometers have long been used to protect hard drives and other moving parts when the sensor detects that the device is being moved or dropped.

The next generation of usages could be more in the foreground, directly impacting (or creating) the user experience, such as in the following examples:

- **Security.** Watchdog applications could potentially sound an alarm in response to movement of the Ultrabook while it is being used to display a presentation at a conference or while left unattended in a coffee shop. If the device suspects theft because of moving away from the owner's cell phone, for example, sensitive data could be locked down, a text message could be sent as an alert, and GPS could track the device to help get it back to its rightful owner.

- **Adapting to context.** Utilities could adapt the system to specific pre-set GPS locations such as home, work, and elsewhere to control factors such as the visibility of alerts from social media, whether sharing is enabled for specific files, whether the webcam is enabled, etc. Similarly, the system could automatically disable Wi-Fi when in specific altitude ranges to comply with air-travel regulations.

- **Lifestyle and travel.** Augmented reality applications using geographical location and compass bearing could overlay point-of-interest (POI) information over an image captured in real time by the Ultrabook's camera, providing a virtual tour guide. In conjunction with conventional navigation functionality, a pedometer could calculate distance travelled and average speed, as well as calculating calories consumed by the effort. Geo-tagging could add location information to vacation photos.

- **Gaming and entertainment.** As described elsewhere in this paper, sensors provide for modalities such as motion input that are well suited to games. Depending on specific system capabilities, it may be possible for some sensor-related functionality to be handled by the microcontroller firmware, freeing processor resources for demanding tasks such as real-time 3D rendering.

Rich Support Available to Assist in Programming for Sensors

An extensive and growing body of tools and techniques are available for developers as they work to build new usage models and application capabilities using sensors. Programming techniques and APIs for accessing sensors are uniform across Intel platforms, including Ultrabooks and other devices such as tablets. Microsoft first introduced a sensor API for Windows 7 in 2009, and with the release of Windows 8, support has been extended for applications based on both the

desktop UI and the Windows 8 new UI, which are shown in Figure 1. The document, "Windows Sensor and Location Platform" (**http://msdn.microsoft.com/en-us/library/windows/hardware/gg463473.aspx**) compares support for the platform in Windows 8 and Windows 7, as well as providing links to valuable resources for each.

Figure 1. The Windows 8 new UI (left) and desktop UI (right).

Sensor-Programming Resources from Microsoft for Windows 8

The latest generation of Ultrabooks benefits from advanced sensor programming capabilities for Windows 8. These mechanisms complement the other capabilities of the OS for Ultrabooks, such as support for touch as a primary input method. Developer resources that add particular value in sensor programming for Windows 8 include the following:

- Supporting Sensors in Windows 8 (**http://blogs.msdn.com/b/b8/archive/2012/01/24/supporting-sensors-in-windows-8.aspx**) is a blog post that provides a technical overview from a developer perspective on the capabilities of Windows 8 programming for sensors.

- Windows.Devices.Sensors Namespace (**http://msdn.microsoft.com/en-us/library/windows/apps/br206408.aspx**) is a set of API references for accessing the various types of sensors (and related sensor data) supported by Windows 8.

- Accelerometer Sensor Sample (**http://code.msdn.microsoft.com/windowsapps/Accelerometer-Sensor-Sample-22982671**) provides sample code, instructions, and discussion related to accessing the accelerometer sensor in C++, C#, JavaScript*, and VBScript*.

Sensor-Programming Resources from Microsoft for Windows 7

Some software makers may choose to develop sensor-capable code to add to existing applications that operate under Windows 7. As mentioned above, a sensor API is available for Windows 7; the following resources provide a foundation for developers as they get started with it:

- Windows Sensor API (**http://msdn.microsoft.com/en-us/library/windows/desktop/dd318953(v=vs.85).aspx**) documents the programmatic means of accessing the native sensor support in Windows 7, including general discussion as well as detailed programming reference materials.
- Windows Sensor and Locations Platform Developer Resources Downloads Page (**http://archive.msdn.microsoft.com/SensorsAndLocation/Release/ProjectReleases.aspx?ReleaseId=2359**) provides sample code, a developer kit, and other tools and resources for developers creating code for sensors under Windows 7.
- Sensors and Location Training Page (**http://msdn.microsoft.com/en-us/windows7trainingcourse_sensorsandlocation_unit.aspx**) directs developers to hands-on labs and training videos related to the Windows 7 sensor platform and its implementation.

Sensor-Programming Resources from Intel

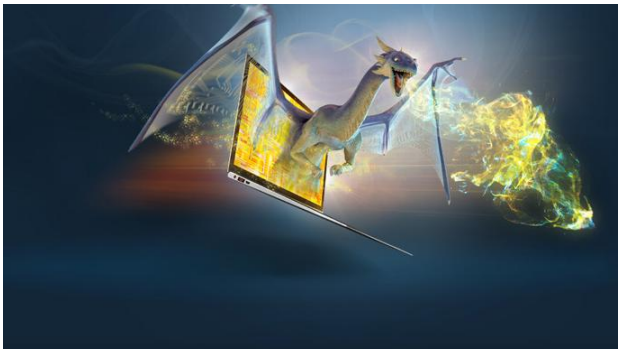
Intel provides extensive resources to help software developers get started quickly building software that takes advantage of the sensors offered in Ultrabooks. The Intel® Developer Zone Ultrabook Community (**http://software.intel.com/en-us/ultrabook**) is an ideal starting point for identifying how sensors fit into the larger opportunity for software that targets Ultrabooks. Intel also maintains a document titled, "Ultrabook Touch and Sensor Resources," (**http://software.intel.com/en-us/articles/ultrabook-touch-and-sensor-resources**) that identifies key articles, guides, and other resources that are more specifically targeted to developing sensor applications.

Development Resources that Help Build a Competitive Advantage

Software makers have a lot to gain from the free-of-charge resources available through the Intel Developer Zone Ultrabook Community. Extend the expertise of your development team to include best practices, tools, and techniques for the Ultrabook category.

Quickly grasping the new opportunities and implementing the skill sets required to take advantage of them helps make applications early to market so they can start building market segment share as soon as possible. A few resources from the community that may prove especially beneficial in driving excitement around your software offerings for Ultrabook devices include the following:

- The Human Touch: Building Ultrabook™ Applications in a Post-PC Age (**http://software.intel.com/en-us/articles/the-human-touch-building-ultrabook-applications-in-a-post-pc-age/**) presents Intel research that investigates which application characteristics create the most compelling user experiences on Ultrabook devices.
- User Interface Design Guidelines for Great Experience Design (**http://software.intel.com/en-us/articles/user-experience-design**) introduces considerations for developers as they produce UIs that deliver outstanding user experiences.
- Adding Touch Support to Desktop Applications for Ultrabook™ Running on Windows* 8 (**http://software.intel.com/en-us/articles/touch-gestures**) provides hands-on coding instructions that demonstrate how to touch-enable an application UI.
- Developing Power Efficient Desktop Applications for Ultrabook™ on Windows* 8 (**http://software.intel.com/en-us/articles/power-management**) is a developer how-to article with in-depth examination of techniques for power-related coding in .NET.
- Re-imagining Apps for Ultrabook™ series (**http://software.intel.com/en-us/blogs/2012/08/09/re-imagining-apps-for-ultrabook-part-1-touch-interfaces/**) offers new ways of thinking and practical design advice that can help fuel innovation at software companies as they explore the new opportunities for Ultrabook devices.



Go set the world on fire:
www.intel.com/ultrabook

About the Author

Matt Gillespie is an independent technology and business writer with a specialty in illuminating the real-world value of emerging hardware and software technologies, mostly working for the microprocessor industry. His previous work experience includes hands-on network IT at California Federal Bank, research writing at the University of California, Davis Center for Neuroscience, and equities writing at Morningstar Inc., the Chicago financial publisher. Matt studied physics and sculpture but eventually received a degree in English from the University of Illinois. You can reach him at spanningtree-at-comcast-dot-net.

[1] Not all sensors are available on all Ultrabook devices. Intel does not specify part numbers or vendors for the sensors themselves – performance may therefore vary depending on components.

Ultrabook™ products are offered in multiple models. Some models may not be available in your market. Consult your Ultrabook™ manufacturer. For more information and details, visit http://www.intel.com/ultrabook.