Making the Most of Embedded Laptop Mice

Victor Marten, Chief Engineer, USAR Systems
Ellen Gooch, Marketing Programs Manager, USAR Systems

Introduction

Laptop and other mobile device OEMs focus primarily on product features such as processor characteristics, memory and display quality when promoting their systems to consumers. Few manufacturers, with the notable exception of IBM, spend advertising dollars extolling the virtues of their system's embedded mouse. Perhaps this is because OEMs perceive little value-added from pointing device technologies. After all, there are only a limited number available – force sticks, touch screens/pads, Hall Effect sensors and that old stand-by, the track ball. Nothing, in short, new.

But what is important to understand is that even pointing devices which use the same sensor may not function alike. Two manufacturers can implement, for example, the same touch pad sensor in their designs and achieve wildly different performance. This is because performance is not merely a function of the sensor technology, but of the firmware of the sensor's controller and the quality of the external circuitry.

Our company, USAR Systems, offers controller ICs and boards for virtually every type of sensor technology. Each of these sensors are, of course, inherently different in terms of their physical characteristics and thus require the use of specific implementation methods. We will cover these methods, in particular for Hall Effect, resistive touch screens/pads and force stick sensors, in some detail further on. However, there are several factors universal to embedded mouse implementation which should be taken into account when designing a system.

Common Pointing Device Board Implementation Ground Rules

1. In order to reduce noise in the circuit, pay attention to grounding. Try to avoid a common digital and analog ground. Implement a "single-point" grounding scheme where all grounds come together at power supply. In addition, use power/ground planes as much as possible.

2. While laying out your board, spend time on floor planning. Do not let your CAD's auto-router place components. The goal is to achieve the shortest possible interconnects between components in low noise/sensitive circuits.

3. If cables are involved, try to position them in the system as far away as possible from a strong source of electro-magnetic interference. For example, try not to place sensor cables near LCD backlights, power supplies or transformers.

Considerations for Hall Effect Sensor (e.g. USAR's HulaPoint™) Implementations

Hall Effect sensors are non-mechanical devices. Without moving parts to break or wear-out, they minimize the problem of dust collection. Hall Effect sensors, like USAR's HulaPoint**™**, work by measuring the varying magnetic field caused by an actuator moving within a small area. A permanent magnet is situated underneath the actuator button, which is mounted on a coil spring positioned on top of a sensor array. The movement of the button is sensed and translated into mouse commands.

Even though the signal from a Hall Effect Sensor is a strong one (approximately +/- .5V), in order to provide smooth motion the A to D conversion must be accurate and have a high resolution in a dynamic range. At the same time, cost issues dictate the use of only an 8 bit A to D converter. Therefore the implementation must rely on firmware techniques to effectively increase the resolution of A to D conversion. Some of the techniques we use are oversampling and averaging of the signal mixed with the output of a low resolution dithering D to A converter.

Figure 1 – Sample Schematic for USAR's HulaPoint™ Hall Effect Pointing Device

Two other factors to consider when designing a board for a Hall Effect sensor is mounting and presence of magnetic fields. Hall Effect sensors require solid mounting. Plus, because they are based on magnetic technology, placement next to low frequency fields like CRT displays or the Bermuda Triangle may not be ideal. However, USAR's solution has been successfully implemented in controls for multi-Kilowatt furnaces.

Pointer Stick/Force Sensing Resistor (e.g. PixiPoint™) Implementation Issues
Force sticks are often chosen for one reason: size. They fit conveniently between the central keys of the keyboard and its buttons can be either its adjacent keys or separate buttons located in a free space. One downside of the technology is that it fails to provide significant user feedback – an end-user can apply pressure to the stick and not effect any movement. Mostly used in laptops, the type of technology is sometimes used in industrial machines, though a cap is usually placed over the stick to provide a wider area for the finger to touch.

One of the implementation challenges of the force stick is that it provides an extremely low output signal. Therefore it requires signal amplification and undithering. For proper performance, the signal needs to be digitized to 8 bit resolution. However, it is riding on top of large offsets created by inaccuracies in the sensor and by low-precision (read: cheap) components used, due to budgetary necessity, in the signal conditioning circuit.

USAR solves this problem by subtracting these offsets in the signal conditioning circuit. Signal correction is developed, via firmware, in a high resolution D to A converter. The highly complex firmware algorithm determines what compensation is needed and does auto zeroing at power-up. While we utilize a simple RC-type integrating D to A converter, conversion resolution is high (16 bits) and output is produced in a short time due to a two-step technique (patent pending).

Figure 2 – Sample Implementation for USAR's PixiPoint™ Force Stick

Implementing a 4 or 8 wire Resistive Touch Screen (e.g. ScreenPoint™)
Resistive touch screens and pads offer the advantage that they may be used with a pen as well as with a gloved finger. This enables more accurate pointing capabilities as well as other types of input such as signatures. These touch pads are also ideal for systems with Optical Character Recognition (OCR) software.

The resolution of a touch screen is, for the most part, determined by the quality of the A to D converter used. The big issue then is not resolution; the issue is noise. Touch screens/pads usually rest on top of CRTs, underlying LCDs or other backlight sources. When noise is not properly eliminated the cursor wobbles. To minimize this problem, a combination of hardware and software techniques needs to be used. Some filtering and signal conditioning can be achieved by the firmware. However, some kinds of interference signals cannot be efficiently filtered in software due to signal aliasing, non-linearities in the A to D converter, etc. Therefore some kind of hardware filtering is necessary. USAR utilizes a simple passive RC network, but other solutions are possible.

Figure Three – Sample Schematic for USAR's ScreenPoint™ Touch Screen Controller

Motion – The Elusive Factor
While we have discussed issues related to signal conditioning and noise, these factors alone do not fully determine mouse performance. Ease-of-use – highly subjective – is also important. Once the firmware and conditioning circuit have combined to create a non-jittery, non-drifting solution, designers must then consider, for example, how sensitive the embedded mouse should be. Contrary to popular belief, sensitivity should not be left to the mouse driver alone. Some things to think about when assessing the ergonomics of a particular pointing device

implementation is who the end-user will be and where the mouse will be mounted. For example, will the mouse be used through thick or thin gloves? This will have an effect not just on the appropriate sensor to use but on the sensitivity of the motion algorithm as well. In regards to placement, if a system designer has mounted the sensor in such a way that the user will find it most comfortable to move it with his thumb, then an algorithm designed for a finger is not going to work well. USAR solves the ergonomic issue by tailoring it's firmware to fit its customer's applications.

Power Consumption

As embedded pointing devices are usually used in battery-operated systems such as laptops, power consumption is a real concern. To minimize consumption, system designers should consider which protocol they employ for mouse communication. USAR supports a variety of protocols, among them RS232, USB, PS/2, Sun and TTL/CMOS serial. Of these, TTL/CMOS level serial offers the greatest advantage in power conservation, followed by PS/2. Sometimes, however, designers are strapped for ports and mice must use an interface not needed by other, more critical system devices. Power consumption still must be minimized even if a less satisfactory protocol is used. One way to achieve low consumption, besides implementing wait modes, is to operate at a lower clock frequency.

External Pointing Device

An important feature, particularly in Laptops, is an external PS/2 port for the connection of an auxiliary desktop mouse. Unfortunately, PS/2 has not been supported since 1987. As a result, mouse manufacturers tend to create their own proprietary commands to the protocol. This leads to incompatibilities from one mouse to the next. To make matters worse, designers of laptop mice with auxiliary PS/2 ports often compound the incompatibility problem by providing a poor external port implementation. Because the code for the extra port requires significant ROM and because, for cost reasons, OEMs want to use a chip with the least ROM possible, designers are tempted to cut corners in the code. USAR minimizes code size in other ways to leave room for an adequate external PS/2 port.

The good news is that the unsupported PS/2 protocol may soon (Microsoft-willing) be replaced by the new Universal Serial Bus standard (USB). While this protocol will not help software minimization, it will solve the incompatibility problem. According to Intel, one million laptops have already been shipped with USB ports, so there is some hope that this connectivity issue at least is nearing the end.

Evaluating a Laptop mouse

Because of the problem with external PS/2 port implementations mentioned above, it is better not to connect a mouse you are evaluating to one of these external ports. Whatever jitteriness you might see is more likely to be the port's problem, not the mouse's.

After selecting a mouse sensor technology appropriate to your application, try to locate every vendor that provides controllers for that sensor. Then test them. Re-create the environment where the device will be used, if applicable. Leave the sensor running on top of the refrigerator or next to the stove. Place it somewhere where it won't be disturbed for a few days and see if it drifts. And get plenty of people to play with it – preferably the same kinds of people that will eventually be using it.

On Cost

Vendors of similar technology will provide, in general, similar quotes. Because USAR offers five other product lines in addition to its pointing device products, we are able to maintain high volumes. We can pass the benefits of this economy of scale to our customers, to provide cost-competitive as well as feature-competitive products. Another cost issue is related to external circuitry. For example, If you are purchasing a sensor and controller chip, but making your own board, price out what the cost of the external circuitry will be. Many vendors, USAR included, provide sample schematics from which you can derive circuitry cost.

In short

A proper implementation of the appropriate sensor to application is a definite value-added to a laptop design. As the mouse is an end-user's gateway to his computer, a good embedded mouse is a big factor in determining whether a system will be a pleasure or frustration to use.