Windows 7 touch implementation notes

HID interface   Custom driver   API / SDK   Contact

Windows 7 is the most recent release of Windows desktop and, following on from the phenomenal success of the iphone dual touch interface, there has been much publicity given to the dual touch capabilites and gesture recognition built into the OS and key applications. Given the visual nature of dual touch gesture usage many publicity shots, promotional material, news items and YouTube videos feature the touch capabilities of this new Windows offering.

Previously there has been a number of Windows releases with built in touch features (Vista had a number of such flavours), specifically those versions aimed at the tablet market and Windows 7 is capitalising on the increased interest in the touch interface, especially where dual touch offers improved gesturing utilisation.

As we all know, point and click type touch has been around along time and in the right environment is a valued interface, i.e. kiosks, tablets etc. Dual touch capabilities in the mass market is in its infancy but the iphone interface showed that in the right environment and on the right device it is a perfect and intuitive interface. Standard desktop utilisation of dual touch and gesture interface is still an unproven concept but nevertheless there is sufficient interest that both touch hardware and software manufacturers need to understand its implementation in the new Windows release.

This document aims to assist with this process of understanding and represents our understanding of the touch implementation under Windows 7. General multi-touch issues were covered in the multi-touch support document available on our web site.

We have be working extensively on Windows 7 touch integration for current and legacy touch hardware and this document reflects our findings thus far:

Firstly the 'single touch' basics: A touch device outputs a touch data packet which contains absolute touch co-ordinates. A calibration process is used to either rescale the co-ordinates to match video resolution as the co-ordinates are received or to remap the co-ordinates in the touch device such that they are rescaled from source. In most OS this coordinate data is received by a driver and delivered to the operating system pointer interface to move the system pointer to the point of touch and generate a 'mouse click'. For single, non rotated, desktop monitor usage this simple point and click usage is sufficient.

Under Windows 2000, XP, Vista and Windows 7, if a touch device is HID compliant it can utilise the HID driver supplied with the operating system to control 'Human Interface Devices' as long as the HID descriptors and generated data confirm to the HID requirements. A separate calibration procedure is used to remap the co-ordinates in the controller to match the desktop video system. Under Vista and Windows 7 extended touch functionality is also enabled in appropriate environments.

Alternatively HID, non HID and serial devices can use a purpose built, custom driver, such as our own UPDD driver, to handle simple point and click usage and at the same time implement additional features, such as multi-monitor support and rotated desktop and other desirable features.

Multi-touch usage. True 'multi-touch' implementations are mainly confined to specialised or experimental systems and most current uses of the term relating to mass market devices, such as the iPhone or Windows 7 systems, are more commonly used with dual touch devices, that is the device is capable of reporting two separate co-ordinate data streams when two stylus are used. Two simultaneous touches allows for enhanced gesture implementation giving an extra dimension to the touch experience and allows OS and application developers to explore the extra dimension in the user GUI experience.

Dual touch devices need to be integrated with the OS and/or applications and although the following concentrates predominately on Windows 7 most of the interface methods described are appropriate, at least in theory, to any operating system.

Three interface methods are available, namely:

Method	Utilisation
In-built HID device driver	For HID multi-touch compliant devices
Purpose built, custom driver	For HID, non HID and serial multi-touch devices and current and legacy (single touch) hardware
Application Programming Interface (API) and Software Devlopment Kit (SDK)	Supplied with the device. Will offer standard application multi-touch interface to device across Windows platforms, Win 2000, XP, Vista and Windows 7

A device may utilise any or all of the above and is not necessarily restricted to any one implementation method.

Each method is now explored in detail……

In-built HID device driver
Windows 7 extended touch features are automatically enabled for HID compliant devices that comply with the core HID and multi-touch extensions

The devices must appear as a Logo certified HID digitizer and:
- Hold standard device class in HID descriptor

- Automated descriptor query
- Observe all required HID and Multi-touch extension usages

Microsoft is working with a number of touch manufactures to perfect this interface, including ELO, N-Trig, NextWindow and Wacom

The built in touch features caters for consistent multi-touch gestures – double click, right click, panning, zoom and rotate and offers single finger panning. The gestures are utilised by the OS and also available for application use.

Multi-touch enabled PCs and a number of Multi-touch enabled monitors are available to take advantage of the Window 7 MT features.

Further reference material

How to design and test multi-touch hardware solutions for Windows 7
http://www.microsoft.com/whdc/device/input/touch_Design-Test.mspx
http://download.microsoft.com/download/A/D/F/ADF1347D-08DC-41A4-9084-623B1194D4B2/Touch_Design-Test.docx

WinHEC 2008 – Multi-Touch in Windows 7 –PowerPoint
http://download.microsoft.com/download/5/E/6/5E66B27B-988B-4F50-AF3A-C2FF1E62180F/MBL-T787_en.pptx

WinHEC 2008 – Multi-Touch Driver Development –PowerPoint
http://download.microsoft.com/download/5/E/6/5E66B27B-988B-4F50-AF3A-C2FF1E62180F/MBL-T528_WH08.pptx

Digitizer Drivers for Windows Touch and Pen-Based Computers
http://www.microsoft.com/whdc/device/input/DigitizerDrvs_touch.mspx
http://download.microsoft.com/download/a/d/f/adf1347d-08dc-41a4-9084-623b1194d4b2/DigitizerDrvs_touch.docx

Windows Roadmap for Drivers
http://www.microsoft.com/whdc/driver/foundation/DrvRoadmap.mspx

Developing Multi-touch Application – PDC2008 Video
http://channel9.msdn.com/pdc2008/PC03

Usage	Full integration via the Microsoft purpose build HID driver.
Pros	Conforms to Microsoft's implemented multi-touch solution to enable full range of OS and application touch features.
Cons	Windows 7 utilisation only.
	Specialist skills required to implement advanced HID solution.

Purpose built, custom driver

For HID compliant hardware that needs additional functionality or for touch hardware that does not conform to HID requirements (including legacy single touch hardware) a custom driver, such as UPDD, can register the touch device as a compatible virtual HID device to enable the operating systems extended touch features. To this end, custom drivers, such as UPDD will enable as many of the in built touch and gesture features dependant on the hardware's capabilities (i.e. single or multi-touch) and Windows 7 restrictions.

UPDD 4.1.8 defines a Virtual HID for any supported touch devices such that all extended touch features are enabled.

Usage	Non HID compliant solution to Windows 7 extended touch features or access to additional touch features implemented by custom driver.
Pros	Does not require advanced HID controller development and skills.
Cons	Will only work with specific touch hardware support by the custom driver. (In the case of UPDD we would hope to support as much touch hardware as possible – including multi-touch hardware).

API and SDK software

In most cases touch screen manufacturers will supply an API interface that allows applications to receive multi-touch input directly and process as appropriate. The information available from the interface may be as minimal as the different touch data packets or may offer gesture type information. Typical examples of multi-touch APIs are available from Next Windows, IR Touch, N-trig and Nexio as well as our own UPDD driver SDK that caters for direct hardware interfacing. It is likely that most multi-touch hardware will offer some form of SDK for application interfacing.

Usage	Cross platform multi-touch application solution. Typically used to create specialised multi-touch application, such as multi-touch games.
Pros	Not restricted to Windows 7 only but will work across other Windows OS (XP, Vista, Windows 7). Allows application specific touch features to be implemented.
Cons	Will only work with specific multi-touch hardware supported by the API/SDK.

Contact

For further information or technical assistance please email the technical support team at technical@touch-base.com