Title Windows Embedded for Medical Devices

WHITE PAPER

Author: Roshan Dalvi
Abstract

This white paper highlights Windows Embedded technology, which is being widely accepted and proven technology in the field of Medical Device manufacture and production. This white paper is an effort to highlight some of the crucial features and application areas of Windows Embedded technology for Medical Instrumentation and control.
# Table of Contents

1. INTRODUCTION.................................................................................................................. 4
2. Why Windows Embedded for Medical devices ................................................................. 5
3. Key Features of XP Embedded and Windows CE .............................................................. 10
4. Address Critical Design Issues......................................................................................... 11
5. Security............................................................................................................................... 12
6. Remote Management......................................................................................................... 13
1. INTRODUCTION

As the Medical healthcare industry increasingly demands more intelligent and reliable systems that work seamlessly together, the Microsoft Windows Embedded family of operating systems offers a dependable architecture, industry standard support, multiple graphical interface options, comprehensive networking support, and core Windows technologies within a single toolset.

Given the number of different types of medical devices and systems that are being created today, developers who focus on the healthcare industry all share at least one commonality - they seek a customizable, reliable operating system on which to base their design.

Because FDA compliance is required at each phase in medical device design (from concept to clinical trial to product launch and servicing), compliance often results in process latency. However, developers also face additional barriers when they create devices for the healthcare industry, such as:

- Fragmented and disconnected product information due to using various operating systems across a product line
- Lack of consistent business processes across disciplines due to the desire to meet widespread customer needs

Windows Embedded technologies can help mitigate these barriers by providing device manufacturers with cost-effective tools to decrease development efforts. With Microsoft Windows Embedded CE .NET and Microsoft Windows XP Embedded, development efforts can be focused on defining and designing the end-user experience rather than on foundational device development. Both Windows Embedded CE and Windows XP Embedded offer a toolset that includes a single, integrated development environment for building, debugging, and deploying an operating system image. Additionally, both products support application development with Microsoft Visual Studio .NET. Developers can easily leverage and transition their existing knowledge of desktop application development straight to devices.

Medical devices and systems include:

- Blood glucose meters
- Patient monitors
- Ultrasound equipment
- Remote and local diagnostic equipment
- Hospital station displays
- Image enhancement
- Multimedia training systems
2. Why Windows Embedded for Medical devices

On initial inspection, Windows CE and Windows XP Embedded may seem to be similar, since both are componentized operating systems, both expose similar programming interfaces (Win32, MFC, ATL, and support for .NET applications), and both expose similar operating system technologies, which include support for networking, internet browsers, media players, and so on. The choice of operating system becomes easier, however, when you understand the design goals of each operating system.

Windows CE has been designed to be a small footprint, componentized, real-time operating system that runs on multiple processor architectures. Windows XP Embedded, on the other hand, can be considered to be a componentized version of Windows XP Professional Service Pack 2 with additional embedded-enabling features. Let’s examine some of the features of both operating systems.

![XPe Development Overview](image)

**Figure 1.0 Windows XP Embedded Development Cycle**
Since Windows XP Embedded is a componentized version of Windows XP Professional Service Pack 2; the operating system has been broken down into almost 12,000 individual components, approximately 9,000 device drivers, and 3,000 operating system technologies. We have the ability to pick and choose exactly which drivers, services, and applications are included in our final embedded operating system image. If, for example, we don’t need a Media Player, Notepad, or Internet Explorer, then we don’t need to include these features in the operating system image. Having the ability to pick individual components reduces the operating system size, and, of course, it also reduces the surface of attack from malicious code. Windows XP Embedded SP2 has all of the security updates found on Windows XP SP2. This includes having all ports on a system closed by default (except HTTP). A system builder can, of course, decide to open the ports that are needed on his device. An average operating system on XP Embedded build is on the order of 40MB. This scales, depending on which operating system components/technologies we include in our final image.

One of the advantages of using Windows XP Embedded in an embedded system is the speed of development. Windows XP Embedded-based systems run on an x86 processor and PC Architecture hardware, both of which are very well understood. This means that any existing Windows 2000 or Windows XP device driver or application can run on Windows XP Embedded without modification. The Windows XP Embedded operating system ships as a set of pre-built desktop-compatible binary components, which ensures application and driver compatibility.

The typical development cycle for a Windows XP Embedded device, as shown in figure 1.0 above, is to first snapshot the underlying hardware using a tool called Target Analyzer, which generates an XML file that defines all of the hardware components, found on the target device. The XML file can then be imported into the Component Designer tool and saved as a custom component that defines the hardware of our device. The Component Designer tool can also import .INF files, which makes it easy to generate components from existing Windows 2000 or Windows XP Device Drivers. Lastly, the Component Designer tool can also be used to create custom components, perhaps defining a component for a custom shell or application. The component defines the files, registry settings, and operating system dependencies for the component. Once components are defined, they can be checked into the Windows XP Embedded component database and then used in an embedded design.

The next step in developing Windows XP Embedded operating system image is to use the Target Designer tool. This tool exposes all of the Windows XP operating system technologies and drivers and the Embedded Enabling Features. A system developer simply selects the components he needs and adds them to a project workspace. This can include the hardware definition component created by Target Analyzer, some design templates that define a starting point for a number of device categories, including Windows Based Terminal (Thin Client), Point of Sale device, Set Top Box, and so forth, and we also have the ability to pull individual components from the catalog to add to our project design. The Target Designer tool will run a dependency analysis on the project workspace at build time to ensure that all required
operating system features are included in the final operating system image. An example of this would be a developer choosing the .NET Framework for his design, adding his .NET application and then building the operating system image – the system designer doesn’t need to know or understand the dependencies of the .NET Framework to build an operating system image.

Windows XP Embedded ships with some Embedded-specific features, including the ability to boot and run Windows XP Embedded from a CD-ROM, or to boot from Flash media. In both cases, a developer would want to treat the underlying media as Read-Only, since flash only supports a limited number of writes, and a CD-ROM would, in this boot environment, also be read-only. Windows XP Embedded ships with an Enhanced Write Filter component that intercepts operating system and application write instructions, and, instead of writing to the boot media, writes to an in-memory cache instead. This means that on power down, the underlying media cannot be corrupted, and the device always boots in a known good state – add to this support for resuming from a hibernation file multiple times, and we get a robust boot environment that provides consumer electronics level at boot time.

So how does Windows XP Embedded compare with Windows CE? Windows CE has been designed as a small footprint, hard real-time embedded operating system that runs on multiple processor architectures, which includes x86, MIPS, ARM, and SH4. A kernel-only build is approximately 200kb, and of course the operating system scales depending on the components we include in our design. A typical residential gateway configuration that supports 802.11a/b/g, remote administration through a web-based interface, and security would be less than 4MB in size. A fully configured Web Pad image that includes Web Browser, Media Player, Office File Viewers, support for the .NET Compact Framework, and Windows Explorer Shell would be closer to 18MB in size.
There are a number of core differences between Windows CE and Windows XP Embedded. First, Windows CE is a full 32-bit, Unicode operating system that doesn't support MS-DOS or Windows 3.x applications. Also, the Windows CE architecture is different from its desktop cousin. On the desktop you would expect to find three core operating system components: Kernel32, GDI32, and User32. None of these components exist on Windows CE. Their counterparts can be considered to be Coredll (kernel32), and GWES, the Graphics Windowing and Event Subsystem (we can think of this as being a combination of GDI32 and User32), which means that even simple desktop applications will not run on Windows CE. At a minimum, desktop applications would need to be recompiled to run on a Windows CE device, not only because of the architectural differences, but also because the Windows CE device is more than likely running on a non-x86 processor.

Windows CE has been designed to be a hard-real-time embedded operating system supporting interrupt latencies in the sub-ten-microsecond range. This is more than adequate for many real-time embedded designs.

Windows CE ships with a suite of tools that can be used to configure, build, and debug an embedded operating system image. The Platform Builder tool is in many ways similar to the Windows XP Embedded Target Designer tool. A developer has the ability to use a Platform Wizard (similar to the Windows XP Embedded design templates) to configure the initial
operating system configuration, and he can then add or remove features from the operating system design. Platform Builder also contains support for operating system feature dependencies. For example, adding the .NET Compact Framework to a design will also add support for the required operating system technologies.
3. Key Features of XP Embedded and Windows CE

- Compliance to class 4 and class 5 medical devices
- Easy integration and faster time to market
- Higher end x86 offer better processing power
- Takes advantage of available tools
- New technologies .NET framework 3.0 support
- Medical hardware interfacing to PC standard buses e.g. USB or PCI
- Flexible Localization
- Power Management
- Message Box default reply- OS management, pop-ups and error warnings
4. Address Critical Design Issues

- Hides Windows Look and Feel
- Upgrades using Network or Standalone
- Increased Boot Speed
5. Security

- Include windows Firewall, IPsec, Wired Equivalent Privacy (WEP), Wi-Fi protected access (WPA)
- Uses NTFS file format
- Data Execution prevention
- Reduced footprint, reduce attack surface area
- Enable pop-up blocker, biometric or smart card
- Antivirus support
- Custom security Templates
- Security at the POS device
6. Remote Management

- Telnet Server- Remote console access over TCP/IP
- FTP server- Used to copy files to and from remote computer systems on a network using TCP/IP, such as internet
- Remote Desktop/Terminal Services- Provides remote access for a system desktop.
- Windows Management and Instrumentation- Provides management and control in an enterprise environment
- SNMP support
References

1. www.microsoft.com
2. www.msdn.microsoft.com
3. SJJ Micro-solutions.
About the Author

I have done Diploma in electronics & Telecommunications and BE in Electronics Engineering from Mumbai university. I have over eight years of working experience in the field of Medical, Analytical instrumentation, Industrial Automation and Control.
I have joined Wipro Technologies in January 2007; I have worked on Olympus Japan – Non Medical controller project and I am currently working for Olympus America– Endoworks project.
About Wipro Technologies

Wipro is the first PCMM Level 5 and SEI CMMi Level 5 certified R & D, IT and Enterprise Services Company globally. Wipro provides comprehensive IT solutions and services (including systems integration, IS outsourcing, package implementation, software application development and maintenance) and Research & Development services (hardware and software design, development and implementation) to corporations globally.

Wipro's unique value proposition is further delivered through our pioneering Offshore Outsourcing Model and stringent Quality Processes of SEI and Six Sigma.

© Copyright 2002. Wipro Technologies. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without express written permission from Wipro Technologies. Specifications subject to change without notice. All other trademarks mentioned herein are the property of their respective owners. Specifications subject to change without notice.