

Windows Embedded

Microsoft Embedded Technologies

VIGNESH DORAISWAMY
PRINCIPAL CONSULTANT
AXIOM ETC
HTTP://WWW.AXIOMETC.COM
VIGNESH@AXIOMETC.COM

Windows Embedded

Agenda

- Build an Embedded System
- Windows Embedded History and Trends
- Windows Embedded Systems overview

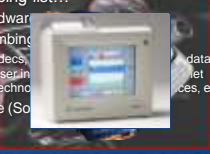
Windows Embedded

Build an embedded system

Windows Embedded

Let's build an Embedded System...

- Shopping list...
 - Hardware
 - Plumbing
 - Codecs, User Interface, etc...
 - Glue (Software)



Windows Embedded

Hardware

- Fast to develop/ship
- Great for low # devices
- Off the shelf BSPs/Drivers
- 3rd party peripherals

Off the shelf

- High up front development cost/time
- Great for mass production
- Requires BSP/Driver development

Build your own


Windows Embedded

Plumbing – OS infrastructure

Not the "value" of your device
Should just ... be here (invisible)

Plumbing includes :

- Networking stacks
- Communications infrastructure
- Media playback Engine
- Codecs
- Drivers
- Servers
- GUI development APIs



Windows Embedded

Operating Systems choices

Monolithic/ General Purpose	Monolithic/ Specialized	Componentized/ Specialized
Off the shelf hardware	Off the shelf hardware/Custom Hardware	Off the shelf/custom Hardware
Ease of development		Potentially requires BSP/Driver development
Speed of development	Speed of development for specific use-cases (APTs)	Componentization provides device flexibility (extra functionality)
Size could be an issue		Locked down device image
Flexibility of underlying operating system and drivers	Hardware/Driver support	

Windows Embedded

A little bit of history...

Windows Embedded

Technology Trends

Fixed function to multi-function devices

Analog to Digital Media

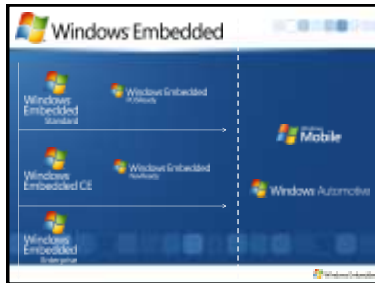
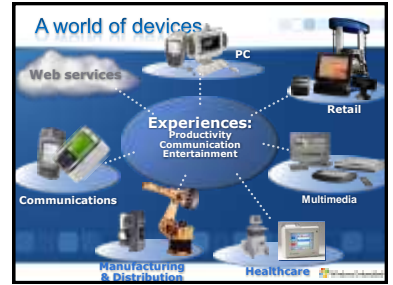
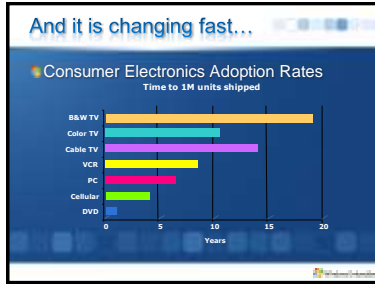
Availability of commercial software

8 bit to 16-bit and 32 bit processors

Exposure to consistent application development platform

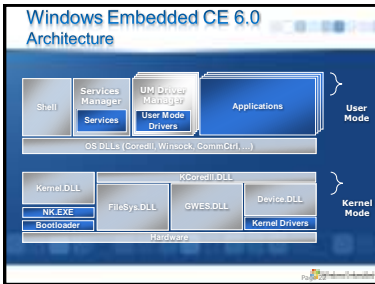
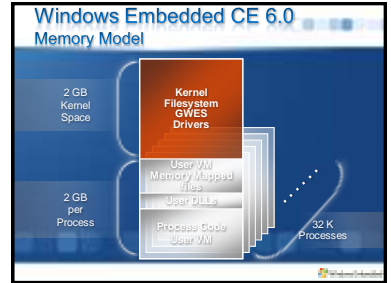
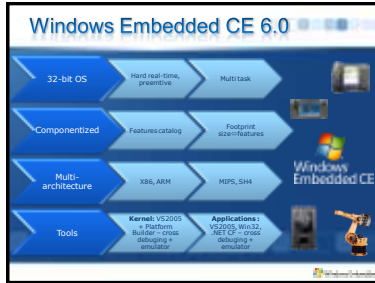
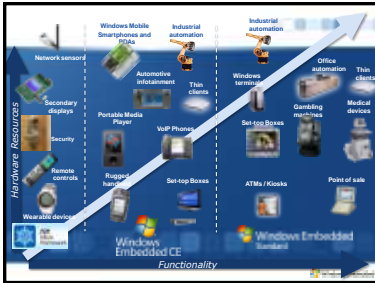
None to Limited, fixed-line bandwidth



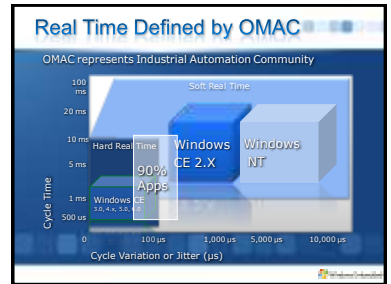


Choose an Embedded platform

	Windows Embedded Standard	Windows Embedded CE
Examples	Sensor Nodes, Side/low display, Health Monitoring, Remote Controls	GPS, Handhelds, PDA's, Digital Picture Frame
Features	Connected, Lightweight, "portable", single GUI	Connected, GUI, Server, Browser, RAS, DirectX
Footprint size	200-400KB Managed Code Included	300KBs without .NET CF 12MBs with .NET CF
Power	Very low consumption	Low consumption
CPU	ARM7, ARM9 without MMU	X86, MIPS, SH4, ARM, with MMU
Real-Time	No hard Real-Time	Hard Real-Time
Managed vs. Native code	Native: Interop only, Managed: .NET MF	Native: natively supported, Managed: .NET CF



- ### Real-Time Overview
- Real time: Applications where specific timings are requested
 - Hard real time: Applications where system fails if timings are not met
 - Soft real time: Applications where system tolerates large latencies
 - Actual timing requirements are system-specific



Windows CE Test Results

- We did a quick test using the following configuration:
 - Samsung SMDK2410 development board
 - 200 MHz ARM with 16x16 cache
 - Windows CE 5.0 with full UI
 - Running a WMV video
- Here's what we got:

	ISR starts	IST starts
minimum	1.2	31.7
average	3.3	67.2
Maximum	13.3	103.0

Time in microseconds (us)

