EUROTECH

Using Windows Embedded Standard to Create a Battery Powered Device

Whitepaper

By: Lawrence Ricci

Abstract

New low power x86 Silicon is now available, so Windows Embedded Standard (WES) properly configured can be productively employed for battery powered devices. We show WES running on multiple version of Silicon, in multiple hardware configurations including consumer, commercial and industrial computers. A series of tests is run to determine processing power per watt, and some indicators of energy consumption per functional operations as well.

www.eurotech.com

DIGITAL TECHNOLOGIES FOR A BETTER WORLD

Introduction

As 32 Bit RISC system increase in processing power, the desktop x86 architectures are decreasing their electrical power demands. Now, the battery powered mobile device can be reasonably engineered around either platform. But desktop applications need some special consideration when they are deployed in a true mobile device and Window Embedded Standard is the OS to accommodate these needs. Windows Embedded Standard (WES for short) is more than just a rebadging of the desktop OS. It is a different environment, with new features specifically targeted for low power, mobile and disconnected operation. This paper and presentation will show how these features can be productively employed.

Desktop & Mobile are Different

Security

A mobile device could be lost, and the security of the data contained can be compromised. A cell phone might have a few dozen phone numbers. A WES device might have the US National phone directory. Clearly the liability is greater. One class of data is especially critical- Biometrics. If an account number or phone number is stolen, it can be changed. If a biometric database is compromised, we have a different sort of problem.

For “Data at Rest” security, WES offers Data encryption at the disk, directory, or file level as EFS (Encrypted File Systems) as part of the basic NTFS file system. While WES stops short of Vista Level “Bitlocker” security there are ways to build an equivalent system.

A variety of third party ‘whole disk’ encryption solutions are available, some of them validated to FIP140-2. Some of these are packaged with the hard drive technology.

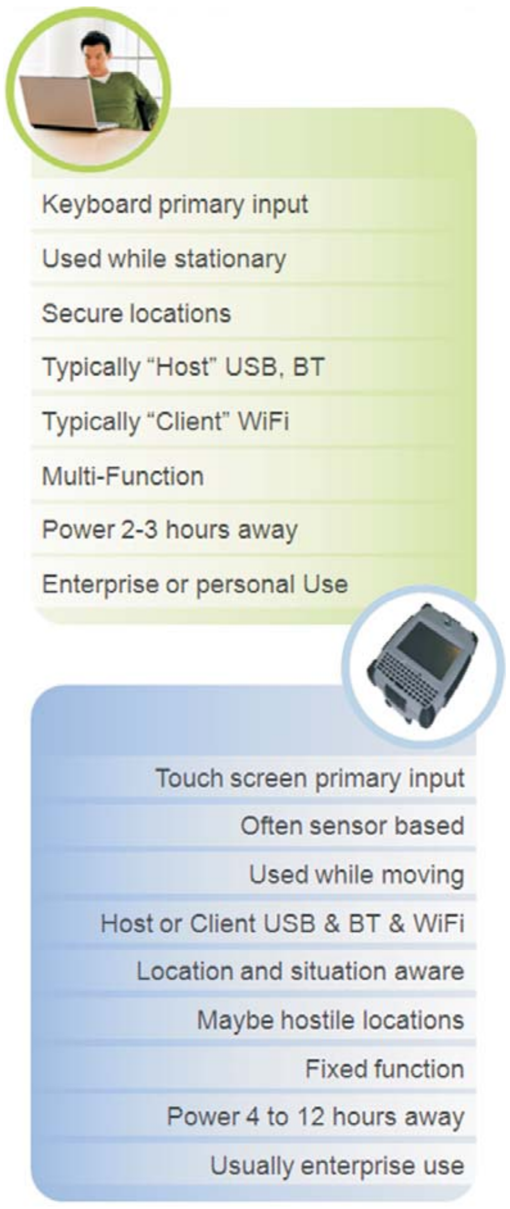


Figure 1- a comparison of "laptop" & "device" design criteria.

For all of these, security starts with identity, and spoofing the identify of a mobile device is a real threat. We expect mobile devices to be applied outside a secure perimeter, so we need to know if it is really the specific device that ‘phones home’. Also, with a mobile device, another type of ‘spoofing’ is possible- a clone or twin device can be switched with the target device, and a perhaps unknowing stooge could carry the ‘evil twin’ into secure facility. Or, perhaps, the Hard Drive can be lifted out of the mobile device and placed in anther computer for analysis and attack.

Notice that good three factor security *of the user* (what you have- badge, who you are-biometric, what you know- password) is not enough to combat this sort of spoofing threat. A properly authenticated user could still log onto a network via compromised device (e.g. with a key-logger installed) and security could be breached. The device itself (not just the software on the disk) needs to be part of the secure path of data to and from the network. This is best done with a hardware solution such specified by TCG (Trusted Computer Group).

Hardware components like the Atmel TPM (Trusted Platform Module) perform important functions- in particular “Secure boot”, which is supported by 3rd party full disk encryption file systems. The idea here is as the bootstrap software is loaded, a hashcode of software and device identity is generated that is essentially unique. This hashcode is stored deep inside the module and is unknown to even user or administrator. Only the system knows what software it is supposed to boot, and the software knows run on only that hardware.

The TPM module also offers a secure repository for passwords and hashed passwords, keys and certificates. Data – like a password- ‘at rest‘ in RAM or on disk sector can be lifted using a commercial tool like WINHEX and revealed. If they are in a disk sector unencrypted by the basic NTFS (for example spooling, temp, swap or print) they might be in clear text. But even if encrypted, by using a fast off line computer to hash billions of clear text strings until they ‘match the hash’, the password or key can be revealed. The best way to keep critical key data secure is to store it deep inside the silicon, where there are tamper detection and response circuits to protect data even from bad actors with what are called “national assets”.

Even away from National Security applications, strong data protection is often required. A case in point is High Definition Media. Many mobile x86 devices are being planned, and playback of HD Video is often one of the target applications. Virtually all commercial HD Video content is protected by strong DRM such as Sony Blu-Ray

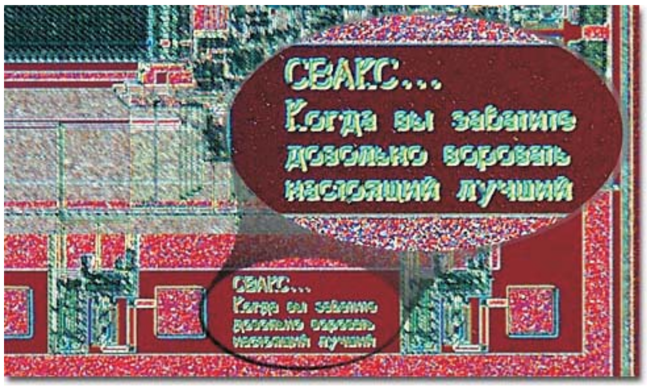


Figure 2- Bad actors will try hard to penetrate security

Florida State University “Molecular Expressions” stumbled across this message while examining the scribe lane on a Digital MicroVAX computer. The text is Russian for the phrase: "VAX - when you care enough to steal the very best". This joke reflects on the black-hat white-hat rivalry of the 1970’s. Where do you think the technology is today?

technology, and there is an ongoing battle fought in the labs and in the courts between content owners and those who want to copy and distribute content without royalty. So far, by use of smart and easily updated VM’s for security, and legislation like WIPO copyright treaty, 1996[1], the HD Content owners seem to be winning. For software built into embedded devices that must cross borders, the contest is not even close. Downloaded “Cracking” software that can run on a PC is perhaps still a subject for extended debate in perhaps a Norwegian court[2], but devices with infringing software can be stopped, at the border, with a simple injunction, causing a severe cash flow problem for both the manufacturer and importer.

These DRM schemes typically require device authentication from the media to the player to the computer and then to the display device where the picture is imaged. Silicon protected identity, like a TCG approved module, is mandatory for such systems. WES, like Vista, can work with such software and can be made complaint with the DMC-1998. Microsoft is doing what it can to make Linux safe from “Tivoization”[3].

[1] Other licenses may restrict this “No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.”   GNU GENERAL PUBLIC LICENSE, Version 3, 29 June 2007

[2] http://www.vnunet.com/vnunet/news/2121179/norwegian-court-clears-dvd-jon

[3] For this ‘community’ term perhaps the best reference is http://en.wikipedia.org/wiki/Tivoization

HMI

The designer of a Mobile WES device should consider the issue of HMI carefully. Even with the graphics power of the Atom chipset, fully supported by out of the box drivers from Microsoft, the device may constrain the size of the display suggesting some HMI redesign. Certainly, the normal "XP" desktop is available, and can be themed with different colors and if desired skinned with significant changes to graphics and function.

Anyone wishing to fully engineer a custom interface for their device should take a good look at Siverlight. Silverlight is targeted for what Microsoft calls RIA (Rich Interactive Applications). Silverlight is developed in Visual studio, under .NET 3.5 and has a direct interface to databases and programming languages in a single IDE. .NET 3.5 is one of the new features available with WES and is not part of older XP embedded.

But Silverlight is a browser based technology, and this might not work for a mobile hand held device. In this case, the preferred interface could be Windows Presentation Foundation (WPF) WPF offers all the features of Silverlight - embedded videos, animation, scaleable vector images etc- but does it all without the browser.

Finally, one more detail about mobile devices: they tend to have a resistive touch screen, not a mouse or track ball. This is a different interface- it has no 'hover' feature for example. Windows Mobile and Windows CE have a convention to provide needed function with tap and hold, tap and drag and double tap actions on the screen, but these have to be built into the touch screen driver, that are different for each touch screen supplier and are not part of WES out of the box. You need to build this driver yourself, or get it from you OEM supplier. A word of caution- this driver, if done improperly, can be a resource and power hog. Use good low power, interrupt driven design techniques to build it.

Networking

Networking for a true mobile device is different. A laptop, while movable, typically operates from a fixed location and maintains a link to a single WiFi access point. A hand held WES device however might be used by a person in motion (Say a worker with an RFID reader/tablet in a warehouse) and might need to 'hop' from one AP to another- while maintaining session connectivity, perhaps securely.

Windows Communication Foundation (WCF) offers this sort of flexibility. WCF, part of the .NET Framework, is used to build applications that inter-communicate. WCF is built into Vista, and can be downloaded and installed by the OEM building a WCF device.

WCF is Microsoft's comprehensive solution to communications, but in this pond Microsoft is not the only big fish. Cisco has close to a monopoly on access points, and their various standards (notable CCX) also are a factor in the environment for an OEM device. Fortunately, these communications technologies are based on compatibility with XP/Vista and are therefore generally plug and play driver installs with a WES system. The OEM should be cautious that for Mobile devices, vendor standards may not be enough. There are strong industry and government standards for Medical devices of different types, DoD information, and various kinds of secure (financial) transactions. Again, fortunately, all of these were developed and proven with Windows PCs devices so OEM adaptations to a WES device is generally pretty easy.

All the answers, however, are not in the software. In particular, the mobile device designer needs to understand where his device fits in a communications scenario. For example, laptops are almost always USB Host. However a mobile device might be USB Slave, or some mix of host and slave roles. This sort of change might require BIOS adjustments.

Power

Power use, or more specifically energy consumption per unit operation, is very critical for battery powered devices. At the end of the day, what is wanted is "Days of Use" at the minimum battery cost and size. The measurement of Clockspeed or MIPS is not a good indicators of a device 'DOU'. The measurement of power is not strictly related to DOU. MIPS per Watt is some degree relevant, but assumes a constant, continues use- rarely the case for a mobile device. But what really counts is Watt Seconds (ergs) per operation.

Mobile applications are interrupt and event driven. For example, a mobile device might have a built in RFID reader. Its battery life will likely be measured not in hours, but in number of reads. If properly designed, the device should consume almost no power unless it is commanded to read a tag. Then, it should spring to action, pulse the RF, read the returning data, sort out false reads, apply some business logic, and then pass the data to local or remote databases, possibly initiating a network connect to do so.

Design for optimal ergs per operation is, as you can see, an integrated effort over hardware, OS and application software. Further ergs/op is very specific to the operations performed and the networks employed. For example, a short SMS might be send with great energy efficiency (nano-ergs per byte) via CDMA or GPRS. A larger document, say a email or graphics, might have its highest efficiency (ergs per megabyte) when transferred over WiFi. WES offers the applications designer a platform to undertake such selective use of networks based on the required bandwidth, security and quality of service .

Power Management with ACPI

Out of the box, WES offers Advanced Configuration & Power Interface (ACPI) which is the same as you would get on a XP Pro Laptop. Built in and out of the box you get basic functions like hibernate. Then, you get the ability to exercise addition power management from the applications level. ACPI level power management is based on simple monitoring of run time idle, as shown in the diagram[4] (over page).

Term	Definition
arm for wake	The procedure that a function driver uses—before the driver powers down the device—to prepare the device to wake itself.
D*x*	A collective term for any of the device low-power states (D1 through D3). D0 is the normal device operating state.
D*x* IRP	A power IRP that the function driver sends to the device stack to transition the device to D*x*.
idle timer	A timer that tracks the time since a device was last used. When the timer reaches the selected idle time-out value, it expires and notifies the function driver to start the power-down process.
power policy owner (PPO)	The driver that controls the device's power state and decides when to change that state. Each device stack has a single PPO, usually the stack's function driver.
runtime idle detection	Conserving energy by powering down an idle device while the system remains in its normal operating state.
S*x*	A collective term for any of the system low-power states (S1 through S5), including standby, hibernate, and so on. S0 is the normal system operating state.
S*x* IRP	A power IRP that is sent to the device stack to notify it that the system is about to transition from S0 to S*x*.
USB selective suspend	A feature that enables a function driver to ask the parent driver to suspend an individual USB device or function while the other devices or functions that are connected to the same parent remain in their normal operating state.
Wake form S0	A device waking itself when the system is in S0. This is the runtime idle detection scenario.
Wake form S*x*	A device waking itself and the system when the system is in S*x*.

[4] Increasing System Power Efficiency through Driver Support for Runtime Idle Detection (MSDN, April 23, 2008)

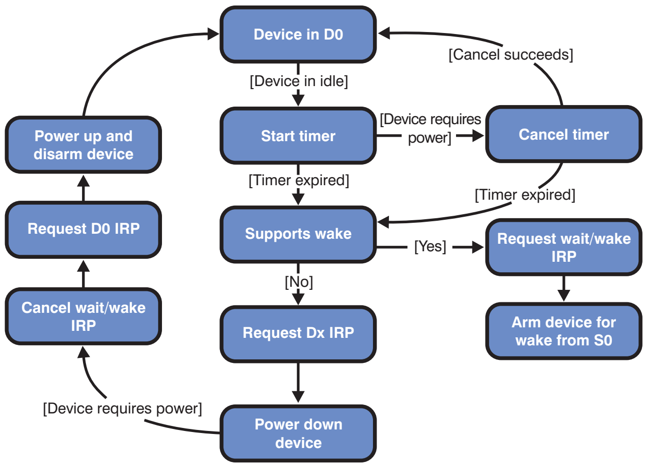


Figure 3- Runtime Idle Detection

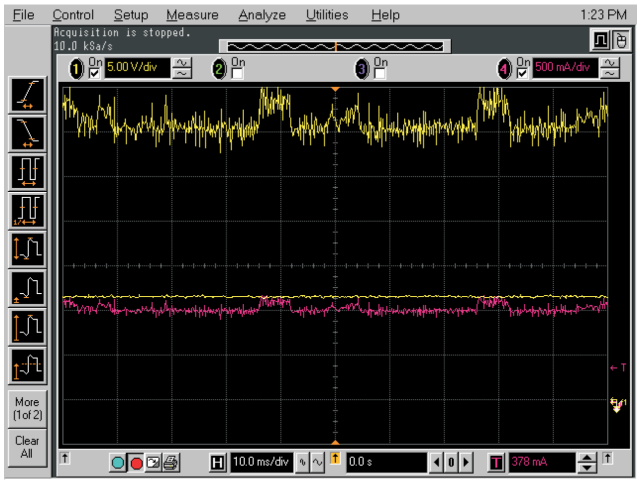
This basic ACPI function, included in WES and as used in laptops, is just the starting point for power management of a mobile device. The 'timer' that paces ACPI is typically set far outside the interval that characterizes the event drive operation of a real time device

Extending Power Management

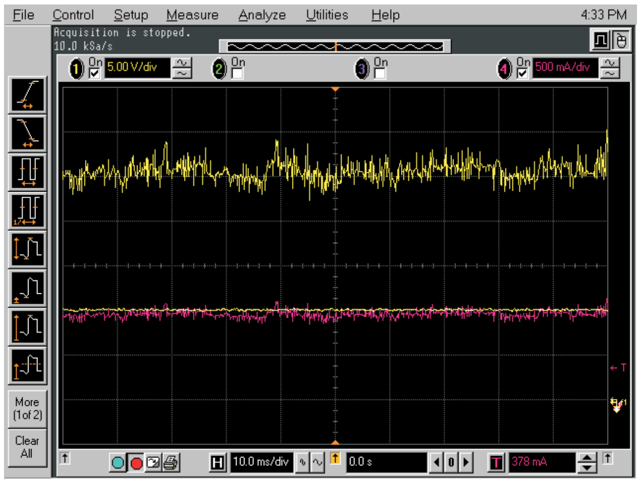
WES 2009 has a new feature (HORM) Hibernate Once, Resume Many. HORM lets a system start up very quickly (nominally a second) to a previously created hibernate file. This feature would, for example. Let an automation system quickly jump into action after a momentary power failure.

For some devices, the OEM should be ready to move past basic ACPI level of power management by integrating it into his application. The OEM or his board supplier can extend the BIOS to provide an API to shift both the CPU and even associated electronics into lower power states. Interestingly, there is significant room for power savings inside fleeting milliseconds of what appears to us to be continuous operations. For example, there are various power drains required for each frame of a MPEG playback: one power level to pull it off the disk, one to decode, one to fill the frame buffer and then very low power (but with the screen backlight left on) to wait for the next frame.

This variable power consumption between the frames of an MPEG is actually visible on a high speed current measuring oscilloscope.



Acquisition is stopped.
10.0 kSa/s



Acquisition is stopped.
10.0 kSa/s

Figure 4- Power Consumption Yellow = power, fuchsia = current, steady Yellow = voltage. MPEG-1 playback, Linux OS is top, WES is bottom. Platform is Catalyst EC.

Battery Charge

Using energy is only half the story, it has to get into the battery first. Lithium chemistry is dangerous, and smart batteries are the norm. Controlling battery charge requires access to I2C or SPI bus interface, from the applications layer, to manage charge and discharge process of the battery. Battery management is a significant hardware/software design challenge and integration with the OS is fundamental to the success of any mobile device.

Enterprise Applications, Maintenance and Administration

WES marks a real paradigm shift in the way mobile computing assets can be integrated in the enterprise. Deployment is as important as device design. Anyone who has had to install a software update, virus removal program or push down a new price list or updated map database to five thousand devices will understand the importance of scalability. With WES efficient deployment can be built into the device.

For lightweight WES devices there is the Device Update Agent (DUA). DUA is a lightweight, 30K, component to perform system updates, update existing applications or application databases, install new components, install new device drivers and generally perform automated, remote or mobile device management. DUA is stand alone; it can poll a Web server, a network share or a local port for CD or USB drive to get its update.

Today most enterprise mobile devices are deployed in a 'star' around some sort of gateway server which deals with maintenance issues like software update, firewall and security, often using third party tools. Since WES is very close to the desktop "XP" it can be integrated more directly to the enterprise, and the device designer needs to consider how he wants his device applied, deployed and maintained. The in place tools to maintain desktops (Microsoft or third party) are an option.

So, for applications development, the programmer has, if he wants, an API to resources like Active Directory, Calendars and so forth. No separate database of names and addresses need be maintained for the embedded devices, and file synchronization facilities are available for 'disconnected' devices.

This means if the OEM and end user wish, the WES device can dial up Microsoft System Update and scan for updates and patches, just like a desktop or laptop today. But realistically, when deployed as a mobile terminal, slot machine, cash register or ATM, the OEM and the enterprise will want more control over the interaction.

This is all available as part of the normal Microsoft enterprise tools set.

Microsoft's standard maintenance console: Windows Server Update Services (WSUS) or Microsoft System Center Configuration Manager (SCCM) can manage WES devices alongside normal desktop and server systems. Stand-alone 'disconnected' devices may require a person to apply updates from a transportable media, like a CD or USB fob.

Another maintenance tool is Watson. Watson is the feature that asks you if you want to report application crash details to Microsoft. An OEM or Enterprise with a complex mobile application can deploy an OEM version of Watson to report crash data directly to them. If the OEM they chooses to get Microsoft involved with the fix, the enterprise or OEM have a way to communicate detailed information.

Finally, part of an update strategy might be to prevent it from happening! Many devices (such as medical equipment) must be operated in a 'approved' configuration. WES includes some write filters that can prevent updates to certain sections of memory, making it impossible to install new applications for example, while still allowing exchange of data.

Summary

Windows Embedded Standard offers new operating system options for the embedded device designer. Now he is free to truly design for the enterprise, for the deployed application .

Copyright ©, Eurotech, Inc, 2009.
All Rights Reserved. This document may not be used for commercial gain without permission of Eurotech, Inc. Any trademarks used within are the property of their respective owners. This document contains technical descriptions that may not be representative of Eurotech product or services.