

# Open Embedded System Strategies

ARC is a leading international market analysis and consulting firm focused exclusively on enterprise applications and industrial automation. ARC provides strategic planning and technology assessment services to suppliers, solution providers, and manufacturing companies around the world. In this contribution, ARC gives its viewpoint on open embedded systems strategies.

## EMBEDDED SYSTEMS CATCH THE IT WAVE

Most manufacturing plants already have many embedded systems and products performing various operations from simple monitoring to critical control. Examples of embedded products include: PLCs, single loop controllers, all types of smart field devices and a variety of operator displays on plant floor machinery.

Every embedded product has some sort of operating system, ranging from proprietary kernels to commercial RTOS (real-time operating systems). QNX, pSOS, VxWorks, Vrtx, Lynx, VENIX, and iRMX are some examples of proprietary operating systems being used. System developers frequently have been using different development platforms to design and implement these embedded solutions. As a result, the embedded product marketplace is very segmented with numerous suppliers and proprietary platforms.

Windows CE is now in a unique position to satisfy most of the needs for real-time operating systems and to bring the IT programming tools to the plant floor. As a result, all plant floor applications can share a single scalable environment and a common network to make the task of data integration easier. Embedded NT will also emerge as an important technology at the boundaries of embedded and supervisory systems.

Java-based applications can be deployed in low-end devices that do not have the system resources

required to run the larger Windows CE platform. One benefit of Java technology is that a complete Java Virtual Machine (JVM) can be embedded on silicon, with Java-based chips already available from Rockwell Collins and will soon be available from other suppliers including Sun Microsystems.

Today with the growing need for enterprisewide integration, numerous pro-proprietary platforms pose many difficulties. As Ethernet-based networks continue to gain momentum and installation in industrial automation; the network limitations, and importantly costs, are reduced. The performance requirements of each networked device are also reduced, since advanced network protocols will be able to synchronize the clocks on the devices on a frequent basis to ensure coordinated control. With Ethernet-based networks, system integration can be achieved with high levels of scalability for a wide range of manufacturing applications.

Proprietary systems and networks will not disappear overnight, but it has become clear that new installations must move to open and scalable systems with broad integration and system programming tools available in the commercial IT marketplace.

## Windows CE POISED TO UNIFY RTOS MARKET

Windows CE is poised to consolidate the embedded systems market because of the broad array of Win32-compliant development tools that developers can use.

*Information Technology continues its march down the enterprise to the factory floor.*

Subsystem	Proprietary	Java	Windows CE	Embedded NT
Low-end Sensors & Actuators	✓	✓		
High-end Sensors & Actuators	✓	✓	✓	
Low-end HMI Displays			✓	
High-end HMI Workstations				✓
Single Loop Controllers	✓		✓	
Low-end Controllers (PLCs & Others)	✓	✓	✓	
High-end Controllers (PLCs & Others)	✓		✓	✓
Network Servers	✓			✓

Table 1. Different Automation Subsystems Require Different Embedded Platforms.

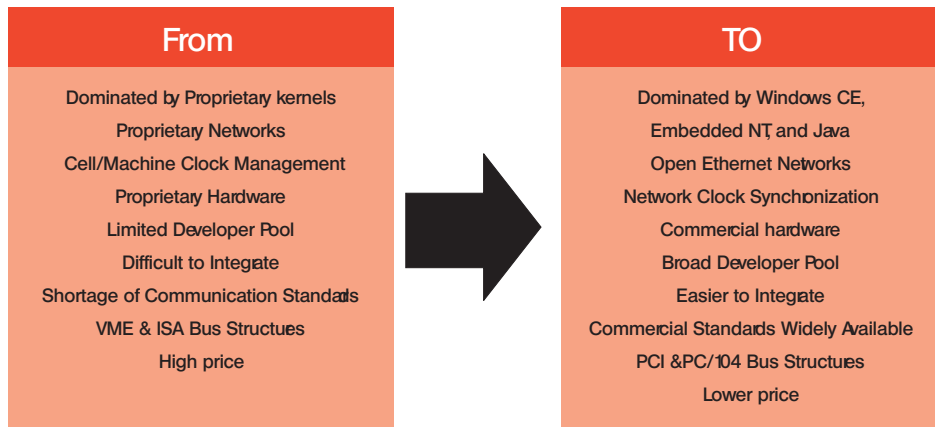


Figure 1. Trends in Embedded Systems.

Windows-based development tools are highly graphical and easy to use compared to the dedicated environments of the individual RTOS options. There is also a large pool of talented software engineers already trained in Win32 APIs for the Windows platforms. These engineers are ready to deploy their expertise on the next class of embedded systems.

In addition to Win32-based APIs, Windows CE shares other component architectures with the Windows NT platform. COM (Component Object Model) and its related distributed model, DCOM, are the cornerstones of the Microsoft DNA (Distributed interNet Application) Architecture. Components written for NT are highly portable to CE because of their common architectures. As developers build libraries of COM objects, they will be able to deploy them across a scalable range of Microsoft platforms and supported networks.

### **CE Appeals to More than Just Developers**

End users, systems integrators and OEMs also find Windows CE compelling, because they will be able to configure CE-based devices with similar tools to their NT supervisory systems. For example, when HMI software is available for both NT and CE, users will be able to use the same screen building tools to develop operator interfaces for either platform. However, users may need to temper their expectations that CE-based flat panel displays will have the same performance and capability that they are accustomed to achieving with NT. Since CE's Win32 API is a subset of NT's, some

compromises will be necessary when porting NT screens to CE-based embedded systems.

Similarly, PC-control suppliers who offer support for both NT and CE platforms give users and OEMs flexibility in choosing the appropriate platform for each application. Control schemes can be saved in object libraries that are later linked together for a particular application. Furthermore, control applications can be

initially developed, tested and debugged on the NT-based development system and later deployed on a variety of platforms, allowing the designer to choose

the optimized system configuration later in the design cycle.

### **CE Not for Every Embedded Application**

The case for architectural uniformity does have its limitations, so users and developers alike should refrain from force-fitting a Windows CE application when it is not appropriate. Although high-end field devices and controllers are ideal target applications for Windows CE, many low-end devices, including switches, actuators, and detectors have price points that do not support the added resources required to support CE. Many specialized applications continue to require very small and fast real-time operating systems. Nevertheless, each new embedded project team should evaluate Windows CE's applicability in light of larger product family strategies. Connectivity with overall NT-based product architectures is causing design engineers to give Windows CE thorough consideration for future product development.

Another issue with Windows CE is that both the OS and applications are configured to suit the hardware platform. This means that instead of purchasing shrink wrapped software solutions that run on a multitude of PC configurations, as is typically the case for Windows NT software, CE-based solutions will be bundled hardware/software solutions. When choosing CE-based devices that complement existing NT-based enterprise applications, the homogeneity link is the software base, not necessarily the hardware platform. ARC expects numerous strategic partnerships to form

*The Windows CE Platform Builder development tool supports the Win32 API, already known by over 4.5 million developers.*

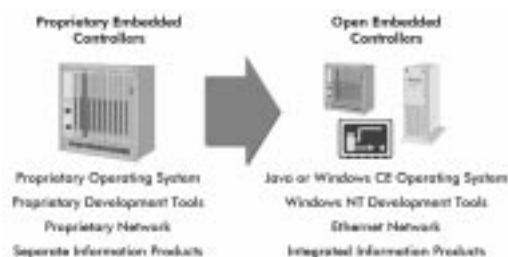


Figure 2. Open Embedded Controllers are on the horizon.

*Windows CE's ties to Windows NT, already a de-facto standard in manufacturing, make it a compelling platform for embedded environments. Although not suited for all applications, CE deserves consideration.*

between hardware and software suppliers interested in developing and marketing bundled CE-based products.

Although Windows CE won't be the platform of choice for all embedded applications, the new version 3.0 expected later this year will help to consolidate a very fractured RTOS and embedded OS market. Since it shares built-in connectivity, a common set of development tools, and synergy with NT's market momentum, CE will go a long way to unify automation architectures under a common and scalable Windows platform.

## WinCE IS A REAL-TIME OPERATING SYSTEM

The design of industrial automation systems using real-time operating systems (RTOS) has for many years been the province of a small cadre of specialists. Today, the number of systems being modernized and newly constructed has created a demand for programmers skilled in real-time processes much larger than the supply. Professional organizations are looking toward the much larger pool of business and commercial programmers to fill the void. In the IT environment, the vast majority of programmers are well trained in C++ and the Win32 Microsoft Foundation Class Library. The challenge is now to use the people with these skills to program and maintain embedded systems. This is one of the largest driving forces behind the acceptance of Windows CE (WinCE) as the operating system for embedded systems.

*Most embedded products today are not networked, and proprietary RTOS technology is appropriately deployed in these cases. As more embedded products are networked together, Windows CE is better suited to enable system integration and data exchange.*

Another important technology is that used for development of Internet web pages. Construction of web pages is based on the use of HTML (HyperText Markup Language), Java, ActiveX Controls, CGI and Pearl scripts. Many tools are used for the development of these pages which are stored on web servers. As users view the web page with a browser running on their client computer, the visual image and text coded into the page becomes visible. Additionally, Java applets and ActiveX Controls can be initiated at either the client or the server to adapt to any client environment. As web server technology becomes available on all networked devices including embedded systems, building and service delivery of web pages becomes more important than real-time programming.

The networked embedded system is much different from the standalone embedded control system. It must offer the user a window into the process or machine

that has never before existed. As a result, the embedded controller must provide the capability for the end user or OEM to support a graphical environment and to work interactively with the process or machine. These are capabilities that are much closer to the desktop environment than the old narrowly focused real-time operating system. WinCE has the capabilities to support both the programming environment and the applications better than the classic real-time operating system.

RTOS program scheduling features for clock-based and cyclic scheduling can be addressed with the proposed WinCE real-time extensions which are based on a base clock running at 1 KHz rather than the Windows scheduling clock running at 60 Hz. RTOS process synchronization typically use semaphores or shared memory counters, while more modern systems are based on message passing between active processes, both of which are available in WinCE version 3.0. Using DCOM, WinCE makes message passing "network aware." In fact, WinCE strongly supports the Component Object Model (COM), the very core of object technology necessary in modern object oriented programming languages such as C++ and Java. The only issue of RTOS vs. WinCE is in terms of speed of response which must be determined for each individual application; obviously very high speed devices may require the response of a low-level high performance RTOS. On the other hand, WinCE Version 3.0 with its target set for 50 microsecond context switching time, will be fast enough for almost all process control or factory automation applications except for some applications in direct machine control.

### Windows CE Is Specific to the Target Platform

Unlike Win98, the Windows NT operating system was designed to run a variety of microprocessors through a "Hardware Abstraction Layer" (HAL), the part of the operating system that translated all functions into the native instructions of the microprocessor. WinCE likewise was intended to run on a variety of microprocessors, but the architecture does not use a HAL. The operating system is actually compiled for code to the specific target microprocessor to keep it fast and efficient. While the Win32 class library is large, only a small subset of it needs to be used in any WinCE real-time application. For example, the large routines necessary for supporting multiple graphic full color windows on a display are not necessary on many embedded controllers.

*Windows CE is not the traditional RTOS, but it may be the first of a new style of networked distributed real-time operating systems.*

WinCE is not the traditional RTOS, but it may be the first of a new style of networked, distributed, object-oriented, real-time operating systems. Like other versions of Windows, it is not keyed for efficiency and tends to use more memory than its trim and slim competitors. However, in consideration of the functionality needed for time-critical activities, WinCE does supply modern

**AD RTS2000**

Criteria	Traditional	PC-based Control	Embedded Control
Network Cost	\$\$\$	\$	\$
Reliability	Excellent	Good	Excellent
Internet Connectivity	Limited	Excellent	Excellent
Distributed Processing	No	Some	Yes
Information Integration	Limited	Excellent	Good
Flexibility	Medium	High	Low
Overall Cost	\$\$\$	\$\$-\$\$\$	\$

Table 2. Embedded products leverage PC technology benefits plus traditional product strengths.

object-oriented means for support in distributed network nodes that are not supported on more traditional RTOS.

## OPEN EMBEDDED PRODUCTS OFFER MANY BENEFITS

Much like the overall push to adopt commercial IT in enterprise and supervisory applications, embedded products represent the next level to benefit from the fast-paced changes coming from Information Technology. The trend toward open architecture based embedded control is also very complementary to the strides already made to bring PC-based control to the factory floor. In some cases, embedded control leverages many of the benefits of Open Control Software (OCS) while preserving the performance and reliability of traditional control solutions including DCS and PLC-based solutions.

Overall system costs will be significantly reduced by open architecture based embedded controllers. One reason is that open embedded control will leverage the commercial market products that are widely available. For example, Ethernet connectivity is cheaper when using standard off-the-shelf network interface cards. FireWire and USB also offer low-cost alternatives

for device communications. Embedded control will be even less expensive than PC-based control because only the required modules will be integrated, instead of paying for additional hardware that is not fully utilized, or is needed to support the underlying architecture but does not solve the control application. For example, Windows NT requires more computer resources than does CE, and in many cases NT can be overkill for many simple tasks.

A big benefit that embedded control holds over PC control is reliability. While PC-based applications are often multi-function boxes with integrated HMI, control and third party software, embedded configurations will typically be fixed-function. In reducing the complexity of the system, reliability should be improved. Another way that reliability will improve is due to the fact that embedded development environments are very modular, so removing hard disks and constraining the user's ability to add software to the platform is possible. The simpler configuration often yields better reliability because there are fewer points of failure.

Perhaps the most compelling reason to consider open architecture based embedded control systems is that they are easier to integrate in an overall enterprise application architecture. Again, Information Technology

*Since embedded products usually have fixed-functionality, their simplicity reduces complexity and improves reliability. In best-of-breed applications, embedded products will be easier to integrate into overall enterprise architectures.*

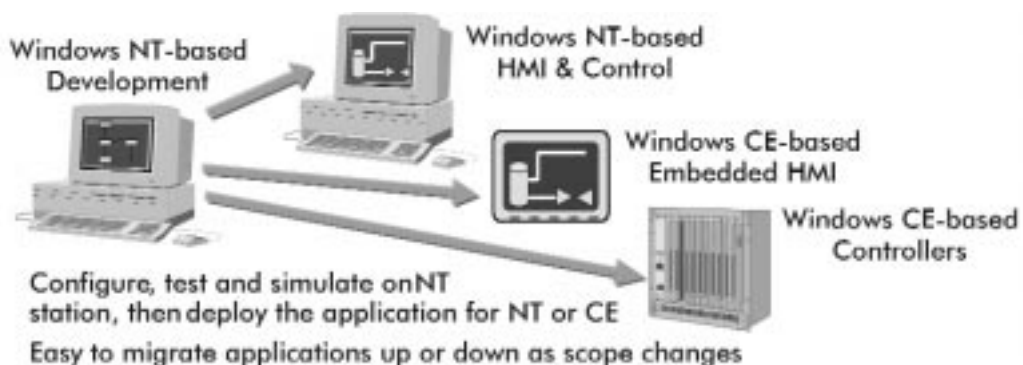


Figure 3. Benefits of Open Embedded Applications.

provides the lowest common denominator for broad integration of data across the en-terprise.

## AUTOMATION SUPPLIERS JOIN THE CE PARADE

Supplier interest in Windows CE is not just academic. Momentum started to build at the ISA conference in October 1998 and has grown since. The recent Embedded Systems Conference and National Manufacturing Week shows featured Microsoft exhibits of a range of CE-based products, from embedded HMI and low-end operator interface platforms to embedded control software. Another aspect worth noting is that support for CE comes from the leading suppliers of traditional control products as well as from smaller software developers providing cutting edge embedded alternatives.

ARC expects that embedded HMI systems, including low-end industrial PCs and operator interface hardware, will constitute the first wave of Windows CE deployment. Companion products, especially handheld computers for mobile troubleshooting and diagnostics will also be among the first applications. HMI software suppliers supporting CE include ICONICS, Intellution, PC Soft, Rockwell Software, Siemens, and Wonderware, with complementary hardware platforms provided by Contec Microelectronics, Dynapro, GE Fanuc's Total Control subsidiary, Rockwell, Siemens and several industrial computer providers.

As the version 3.0 improvements, slated for release by the end of 1999, de-terministic CE-based control products will emerge. While some suppliers promise control products in the very near term, most automation suppliers are waiting for the slated performance improvements of CE 3.0 before re-leasing products for

general distribution. Leading the charge to CE-based control is Think & Do Software, who expects to ship their solution on CE ver-sion 2.1 and ATR who recently demonstrated a 3-axis CNC solution. Other control products are expected to follow from Intellution, GE Fanuc Total Control, PC Soft, Rockwell, Siemens, Schneider, Vision Controls, and Won-derware.

## EMBEDDED NT IMPROVES RELIABILITY

Instead of a technology leap between Windows CE and Windows NT, devel-opers now have an intermediate platform that is better suited for the embedded market: Embedded NT. Microsoft officially announced it last fall, and Beta versions are now shipping to evaluators with a widespread release expected by the end of 1999. Embedded NT has all of the resources of its bigger brother, standard Windows NT, but the environment gives develop-ers more flexibility in tailoring the software components, device support and services necessary for a specific application. Windows NT can be deployed for the masses of general-purpose desk-top machines and network servers, while Embedded NT is suited for high-end embedded applications.

For manufacturing applications, Embedded NT will improve the reliability of the operating system because only the required components are combined via the Target Designer toolkit. Furthermore, runtime applications and third party components are added to the mix with the Component Designer, and the result is a load-able runtime image for the specific target hardware.

Instead of clear breakpoints between the standard NT, Embedded NT and Windows CE, Microsoft has positioned the suite of operating systems in a continuum that provides overlap between functionality while giv-

Microsoft Partner	CE-based Application
ATR	RCS 3-axis Milling Machine control engine
BlueWater Systems	CE device drivers & performance exten-sions
Contec	Single Board Computer
Dynapro	Touch screen computers
FactorySoft/PC Soft	OPC Toolkits, HMI & control software
GE Total Control	HMI & control software,FactoryClient 2000 OI hardware
ICONICS	Genesis HMI Software
Intellution	FIX Dynamics HMI & Control Software
Intrinsyc Software	DeviceCOM, TCP/IP support for COM/DCOM
PLC Direct	WinPLC control platform
Rockwell Automation	OpenController, DriveExplorer & Pocket-Logix Diagnostic PCs
Schneider Electric	Momentum controller
Siebe	Micro I/A controller
Siemens	WinCC HMI, WinAC control on MP270 and Simatic M7
Think & Do Software	PC control software for CE 2.1
VenturCom	RTX API for CE and NT
Vision Controls	Rule-based control engine for NT & CE
Wonderware	InControl PC control software

Table 3. Automation Partners Add Depth to CE Adoption in Manufacturing Applications.

	Windows NT	Embedded NT	Windows CE
Networking	Full	Full	Some
Footprint Size	64M RAM 100M persistent	16M RAM 20M persistent	40K RAM 256K persistent
Headless/Diskless Options	No	Yes	Yes
Rich Graphical User Interface	Yes	Yes	No
Flexibility	Yes	Yes	Fixed
Platform	Intel Pentium & compatible	Intel Pentium & compatible	x86, MIPS, ARM, SHx, PowerPC
Real-time and Determinism	3rd Party	3rd Party	Standard in 3.0
License Cost	\$100-300	\$50-100	\$30 or less

Table 3. Embedded NT Bridges the Gap between Standard NT and CE.

ing devel-ops the flexibility they need. The primary difference between standard Windows NT and Embedded NT is the design modularity of the different services and components. Developers are given more freedom to customize the NT operating system, and consequently the hardware required to run it, with a direct effect on cost. The license cost of Embedded NT will also be highly dependent on the modules chosen and the number of units in the purchase agreement. Higher volumes will provide better price points.

### **Embedded NT Offers Many Benefits for Automation Applications**

Like its bigger brother, Embedded NT supports 32-bit processors, the full Win32 API, and the host of networking and communication protocols. It also supports similar development tools, and smooth application migration between standard NT and Embedded NT should be realized.

While its similarities make it compelling, Embedded NT's differences with the full Windows NT environment make it a more appropriate platform for several factory floor applications. For higher reliability, there is greater support for "disk-less" configurations, where the storage media is solid state instead of rotating hard disks that are prone to failure in industrial settings.

One important example is that the Graphical User Interface (GUI) can be removed from the Embedded NT platform, making "head-less" configurations that don't require keyboard or mouse input, ideal for networked PC-based controllers and other devices that require extensive I/O communications beyond the scope of Windows CE's capabilities.

### **EMBEDDED SYSTEMS REQUIRE NETWORKING**

At the very core of the distributed building blocks of all automation systems is a microcomputer dedicated to a single task or purpose. Most often, this microcomputer is programmed to operate some machine or device on the manufacturing plant floor. It is intended primarily to operate that machine or device, and to communicate operational data with the host computer. Embedded computing systems always communi-

cate with other systems and with process or machine operators.

Embedded systems involved with the control of real-time processes or machine operations must be highly responsive. Frequently the larger process or manufacturing machine will have many embedded systems, each dedicated to part of the task of operations. It is common for all of the embedded systems to be coordinated in such a way that they are in tight synchronization to control the process or machine in some exact pattern required to make the desired product to specification.

Previously, a large PLC or process control system was programmed or configured to control all phases of the manufacturing process. Software complexity was absorbed in the design of these large systems and was left to the skills of the "programmer" or "process engineer." More recently, packaged processes have begun to appear, each supplied with their own embedded controller. The packaged system has now become a true "Automation Object" with only its attributes and documented methods visible to the overall control system. Embedded systems are the embodiment of Object Technology into the overall control structure. It is no longer possible to control all aspects of such systems by a single controller or program, but operations must now be synchronized over an interconnecting network.

This defines the problems of today's embedded systems: multiple computers, programmed by multiple suppliers, with no commonality, but they must closely interoperate and be completely synchronized with each other. This problem is difficult enough for a single supplier, but when multiple suppliers are involved, the integration problems become overwhelming. The integration problems are being resolved with the use of Ethernet and TCP/IP as a common network tech-

*Open networking standards are required to coordinate and synchronize the object-based control scheme. Object interoperability is a critical requirement for next-generation open systems.*

nology and with publisher/subscriber applications to eliminate the time variance.

## **DEVICE SYNCHRONIZATION IS BEST PERFORMED BY REAL-TIME NETWORKS**

When the entire process is controlled by a single controller, its real-time features could be used to synchronize operation of the entire system. Today, embedded controllers in machines, devices, or instruments use their own RTOS to control their own features, but a separate master controller is necessary to synchronize operations between each of the embedded controllers.

*Apart from the normal functions of an Operating System, an RTOS has some dedicated features for coordination and scheduling of Real-Time processes or tasks.*

*These include the following:*

1. *Cyclic scheduling - repeatedly start a process at a synchronous time interval shorter than its worst case execution time*
2. *Deferred scheduling - start a process for one-time execution delayed by a specified time interval*
3. *Event-based (interrupt) scheduling - start a process when a defined external event occurs; optionally after a defined time delay*
4. *Inter-process synchronization - suspend a process until an expected internal event occurs (semaphore)*
5. *Abort a process - abnormally end a process*

The master controller keeps the clock and sends start messages to each slave processor in the embedded systems. Alternatively, each embedded system keeps a real-time clock and starts its tasks on a time schedule, if tasks are time based. The master system keeps all clocks synchronized and organizes the master schedule across the network. If embedded systems are synchronized by events, then synchronization can be based on sensors and used by the cell controller.

An RTOS must control process (task) execution by time schedule or by event occurrence, but it can only do this in a single node of a network of embedded systems. When the RTOS ran in a centralized controller, it had the ability to do all of the scheduling. One possible system architecture is to restore the scheduling function to a centralized "master scheduler," but this defeats one of the main precepts of distributed control - to make each node autonomous, capable of continuing control in the event of any single node failure. The only other system element in common with all nodes is the network itself.

### **SERCOS and Foundation Fieldbus Provide Network Synchronization**

All of the necessary inter-node synchronization can be accomplished by network protocols which are designed specifically for this function. Only two standardized network protocols are designed for node synchronization: SERCOS and Foundation Fieldbus. SERCOS, IEC 61491, was designed for multi-axis motion control in which it is necessary for all axis drives to move in tight time synchronization. SERCOS accomplishes the synchronization by sending the next desired motion setpoint or pattern of setpoints (ramp or curve vs. time) to each axis drive from a centralized

# **AD NÜRNBERG**

master controller. Each node then awaits the master controller's broadcast "start" signal, and all distributed axis controllers move to the desired setpoint or follow the downloaded pattern of setpoints in tight time synchronization.

Foundation Fieldbus, based on ISA/ANS S50.02 and draft IEC 61158, is designed to schedule process control loop execution on an exact time schedule to minimize jitter which can cause errors in the dynamic loop response, and in some cases cause control loop instability. It does this by synchronizing the execution of PID (or any other control calculation) control to a very tight time schedule such that measurement signals are processed "just-in-time" before control calculations, which are then completed and passed to output nodes minimizing all deadtime. Each of these operations, measurement, control and output may be in different network nodes, or the same node. Synchronization is accomplished by time-based scheduling to a clock which has a guaranteed low drift rate by being synchronized to a master clock. Data is transferred by "publishing" to a dedicated list of "subscribers." In Foundation Fieldbus, all synchronization is accomplished as part of the setup function, and does not require any more than a master clock.

In SERCOS and Foundation Fieldbus networks, the time and event-based logic has been distributed to the network nodes and the network protocol itself. The master controller has become only a source for the real-time clock used for synchronization, and as a repository for pattern and recipe information and as a buffer for event reporting. The classic requirements for RTOS are not generally required in the master controller, only that it be "fast enough" to accomplish its purpose when synchronization occurs on the network. System synchronization is moving away from each node maintaining its own real-time clock to network-centric management of all nodes coordinated via the standardized network protocol.

## PCI CAPTURES EMBEDDED SYSTEMS HARDWARE PLATFORM MARKET

In the development of a computing platform for embedded systems, each manufacturer must consider a make-or-buy decision: custom printed circuit board designs cost more to design and support, but should cost less to manufacture. However, standardized board-level electronics operating on standardized buses are usually competitively priced, available from several sources, and keep pace with microcomputer advancement. This is not an easy decision for systems sold in volumes which can be used to cost justify custom designs. Making such decisions more difficult are the rich array of board level products being produced for use in embedded systems at very low prices.

The progression away from proprietary backplane bus technology is now complete with the emergence of newer higher speed bus technology "borrowed" from the PC. Manufacturers have three practical choices for new embedded systems: PCI, Compact PCI or PC/104-Plus (also based on PCI.) While legacy

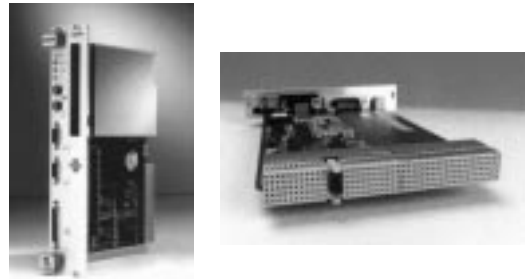


Figure 3 & 4. (Courtesy PCI Industrial Computers Manufacturers Group).

designs and general purpose computers still use buses such as VME and ISA (Industry Standard Architecture), most computing equipment has migrated to the PCI bus and its variants. Most of the high-end servers are using the 64-bit extensions to PCI as their primary I/O bus but use proprietary special purpose memory buses. One of the most promising buses of just a few years ago, Futurebus, is now all but a memory.

## Size and Speed Matter in Embedded Hardware

The choice for embedded systems is primarily to be made on the need for small spatial dimensions. Compact PCI is physically similar to the older VME design, but its bus is notably faster (33 MHz) and more space-efficient. The basic design of CompactPCI is still, like VME, for removable printed circuit boards in a card cage. CompactPCI provides for two sizes of cards (3U and 6U) as defined by IEEE standard 1101, the same as for VME. However, the backplane uses a very different pin-and-socket connector conforming to IEC 917 and IEC 1076 standards. The basic J1 connector, required by all systems, provides 110 pins, and enables 32-bit data transfers. A second J2 connector may be used on a 3U board which adds another 110 pins and enables 64-bit data transfers. Additional connectors (J3 and J4) are included on 6U boards and are allocated for I/O connections made via the backplane. The primary market for Compact PCI has been telecommunications with real-time and environmental requirements similar to those of factory automation.

PC/104-Plus uses a bus architecture very similar to

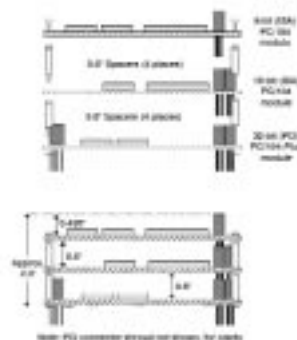


Figure 5. (Courtesy PC/104 Consortium).

CompactPCI (both are based on Intel's 32-bit 33MHz PCI bus), but continues the older PC/104 design of stackable small printed circuit boards with pass-through connectors. The packaging of PC/104-Plus makes a very compact high performance computer possible, and provides a degree of rigidity not possible in any other standard design. The pass-through connector used for PC/104-Plus offers 120 pins (one used for keying) for compatibility with other PCI bus designs. If 64-bit data transfers are required, another identical connector is used, but such designs are not anticipated for embedded systems applications. PC/104-Plus originated with the emergence of low-power high performance Pentium class microprocessors which were not capable of using the older ISA-based bus included on the original PC/104 design. The market for PC/104-Plus is for embedded and mobile computing where physical space is critical and vibration immunity is important.

*Embedded web servers solve system integration problems by enabling seamless data transfer regardless of host operating system.*

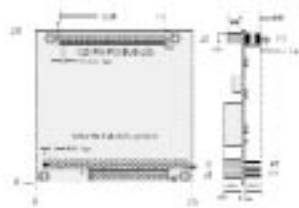


Figure 6. (Courtesy PC/104 Consortium).

## EMBEDDED WEB SERVERS ENABLE INTEGRATION

Embedded systems have typically performed simple control and monitoring functions related to a machine, process or device. One of the tasks has often been to update a controller, database server and/or an operator station with current values of the measurements, outputs and status of the controlled item. Each upstream system has had its own set of demands and most typically has sent polling requests to the embedded controller which has responded with messages for each device in its own format. Software integration has been a problem for such systems. One solution has been to use open standards for message formats such as the Modbus protocol or MMS (Manufacturing Messaging Specification), which is an ISO standard 14226. Another solution has been the use of OPC (OLE for Process Control), developed by the OPC Foundation. OPC specifies the generic object model for a device so that a variety of networked systems can interact with it using the OPC defined interfaces, but this works for host systems only in a Microsoft Windows environment.

The embedded web server is a totally different approach to solve the system integration problem.

*Although PersonalJava and EmbeddedJava are similar, the primary difference is the GUI: PersonalJava-based devices have displays, while EmbeddedJava devices have limited displays or none at all.*

While the previous approaches require the field device to conform to the needs of the host system, the embedded web server adds the ability of the field device to manage its data presentation at the host operator display using the Internet HTML protocol. The great

attraction of this mechanism is that the host can run any operating system on any microprocessor as long as it offers an Internet web browser. Data can be assembled into packets and transferred using FTP (File Transfer Protocol) protocol. Alarms can be sent to any e-mail address using SMTP (Simple Mail Transfer Protocol), and the remote device can be managed across a network using SNMP (Simple Network Management Protocol), which are other Internet protocols. The host device's Internet interface handles all of the data transfer details using standards set by the WorldWide Web Consortium, which are supported for practically all modern operating systems on all popular micro-processors, workstations and mainframe computers.

Embedded web servers can control the presentation of data and the support of their system without the sometimes incomplete translation of host software. Embedded web servers are one of the first products of new system-on-a-chip technology now becoming available. While some silicon suppliers such as Motorola have offered Ethernet interfaces integrated with their microprocessors, NETsilicon is the first company to offer a family of NET+ARM chips which include all services needed to deliver the embedded web server.

## ROLE OF PERSONALJAVA AND EMBEDDEDJAVA IN EMBEDDED SYSTEMS

Real-time programming is key for manufacturing applications, as it enables devices or systems with deterministic behavior, allowing predictable response times and the ability to coordinate functions without missing a beat. While invisible to the everyday user, real-time software controls are in many of the consumer and industrial devices and supporting network systems behind those devices, including pagers, mobile phones, printers, test and measurement equipment, navigation equipment, medical devices, telephony infrastructure equipment or network routers and switches.

Java technology's promise of "Write Once Run Anywhere" will help alleviate the growing software costs currently found in the real-time domain which employs many diverse execution platforms and operating systems. But is Java technology ready to revolutionize the development of real-time systems?

### Sun Provides Two APIs Geared for Embedded Systems

Recently, many leading real-time operating systems (RTOS) suppliers sought Sun Microsystems' licensing for both PersonalJava and EmbeddedJava technology.

*ARC expects that the RTJWG will be more aggressive in its efforts to extend the real-time Java space than Sun, since the real-time market is more than just a niche to the parties involved.*

gies to be integrated with their products. These companies, which include Acorn, Chorus, GeoWorks, Lucent Technologies, Mentor Graphics' Microtec Division, Microware, QNX and WindRiver Systems represent more than 50 percent of the total market for commercial real-time operating systems, supplying real-time software that allows consumer devices to operate.

PersonalJava is designed specifically for network-enabled products people use in homes, while mobile or in the office. These devices have displays but might not include a keyboard or a mouse.

EmbeddedJava, designed to run on a wide variety of high volume micro-processors, serves manufacturers who want to leverage Java benefits in devices with limited displays, such as instrumentation, low-end mobile phones, pagers, factory automation equipment, fax machines, and network routers/switches.

*Sun's alliances with RTOS suppliers and the formation of Sun's Real-Time Java Expert Group show the firm's commitment to extending the technology. Many developers feel that the Java Community Process is restrictive to innovation, however, so the alternative Real-time Java Working Group will also play an important role in shaping the technology for real-time applications.*

ARC believes that Sun's strategy for real-time is aimed to complement a traditional RTOS, allowing consumer devices to run Java programs and connect to any network. Thus, PersonalJava's primary focus is on devices with sophisticated displays, while EmbeddedJava addresses the needs for devices that have a character-based display or no display at all.

Both technologies will enable RTOS suppliers to offer a more complete software solution to their users. The consumer electronics industry is already standardizing on PersonalJava and EmbeddedJava, which may provide some synergy to the device level and personal electronic appliances. Leading telecommunications providers such as Alcatel, Nortel and Samsung have also announced their plans to adopt PersonalJava into their new webphone products.

### **The Real-time Java Working Group**

In June 1998, the National Institute of Standards and Technology (NIST) began work to coordinate the efforts of developers who were extending Java technology for real-time applications. The purpose of the NIST group was to create a vendor-neutral requirements document, from which a NIST-governed standard might spawn. Members of the initial efforts included Sun as well as its competitors, developers of real-time and embedded implementations of Java, including IBM, HP, NewMonics, Microsoft, Aonix and several others.

At the same time, a Real-time Java Working Group (RTJWG) was formed, primarily as a result of concerns

with Sun's licensing programs. It was also felt that the needs of real-time Java might not be "mainstream" enough to garner full attention from Sun, who has broader expectations for its full offering of Java technology. The group has two primary goals: the first to write a specification for real-time Java that meets the NIST requirements, and secondly to gain sponsorship from a governing body for the resulting open standard.

The RTJWG approached the National Committee for Information Technology Standardization (NCITS) to sponsor the efforts to create the real-time Java standard. Although a subsequent vote to continue the work under the NCITS process failed to pass, the RTJWG, primarily driven by NewMonics and HP, continues to work to gain consensus of the different parties on a specification. ARC expects that the RTJWG will be more aggressive in its efforts to extend the real-time Java space than Sun, since the real-time market is more than just a niche to the parties involved.

*In this era of cost controls, gone are the days of blind reliance on a single supplier. Instead, suppliers are now needed to help create open and economical automation products, based on commercial IT standards.*

### **Sun's Real-time Experts Group**

Early this month Sun announced the Real-time Experts Group, the first technology development group formed under Sun's Java Community Process for developing extension APIs. IBM is leading the group, which is composed of a core team of industry leaders including Cyberonix, Microware, Nortel Networks, QNX, Rockwell Collins, and Sun. However, the group has also an extended team including representatives from Aonix, Apogee, Carnegie Mellon University, Lockheed Martin, Lucent, MITRE, Mitsubishi, Motorola, NSICom, NIST, Schneider Electric, Thomson-CSF and Wind River Systems.

The Group is chartered with developing the real-time specification for extending the Java platform, thereby empowering developers with the technology they need to write real-time applications in the Java programming language. This real-time specification is aimed at broadening the market for Java technologies and moving the platform forward using the collective innovation of the industry.

ARC believes Sun's alliance with RTOS suppliers will enable manufacturers to leverage their development and get products to market more quickly. Real-time software is a critical component in many of the devices manufacturers interact with every day. The involvement of suppliers and users with the Real-Time Java Expert Group will enable IT groups to ensure that the needs of manufacturing applications industry are represented when it comes to determining the best specifications for real-time Java technology.

### **LOWER YOUR TCO WITH OPEN EMBEDDED SYSTEMS**

At one time it was easy to decide which system to purchase: turn to your favorite supplier and purchase their

system. If networks were required, that supplier would take care of your needs with their proprietary network. If critical timing were an issue, the vendor's master system would take care of your needs. If integration with a "foreign device" were necessary, your primary supplier would provide a "black box" to interface it and solve your problems.

The automation world has grown far more complex and cost-conscious. It is no longer accepted to pay a system integrator to assemble an automation system from components supplied by many specialists. Staff reductions prevent the end-user from doing the bulk of the integration. ARC encourages end users to turn to their favorite suppliers and ask them to help solve this dilemma in an economical way. At ARC, we believe that the answer is "Open Embedded Systems".

Open Embedded Systems are attractive to the end user mostly because they allow integration of systems supplied by many suppliers without requiring each supplier to cooperate in any way other than to build to a set of industry standards. The standards for open embedded systems that are being used are those shared with IT for the development of open business systems. The network technology is Ethernet with TCP/IP and the Internet suite of protocols. Data is shared by using a common object model such as Microsoft's DCOM or the Object Management Group's CORBA. Applications are written in Object-oriented languages such as C++ or Java, most often with the Win32 class library for operation on a Windows-based operating system. The same technology being used for the development of business applications is then available and can even be integrated with automation systems.

Users may still be tempted to use legacy systems based on older RTOS and with vendor proprietary networks. These choices are more acceptable in closed systems not required to interoperate or share data with other automation systems or business systems. However tempting these closed systems may be in terms of purchase price, long term experience has shown that the costs of integration for closed systems is much higher in terms of Total Cost of Ownership (TCO) over the life of the embedded system. ARC advises its clients to consider the TCO and to implement all new and retrofit automation systems with Open Embedded Systems ■

---

*Dick Caro, Bill Thompson, Marcus Goncalves,  
Analysts for the ARC Advisory Group.  
Editor: Andy Chatha.*

## ACRONYM REFERENCE:

AI	Artificial Intelligence
ANSI	American National Standards Institute
API	Application Program Interface
APS	Advanced Planning & Scheduling
BIOS	Basic Input/Output System
BOM	Bill of Materials
BPR	Business Process Reengineering
CAD	Computer Aided Design
CAN	Controller Area Network
CEMS	Continuous Emissions Monitoring System
CIM	Computer Integrated Manufacturing
CMMS	Computerized Maintenance Management System
CNC	Computer Numeric Control
CPG	Consumer Packaged Goods
EAI	Enterprise Application Integration
EC	Electronic Commerce
EDI	Electronic Data Interchange
ERP	Enterprise Resource Planning
HMI	Human Machine Interface
ISA	Instrument Society of America
IT	Information Technology
LAN	Local Area Network
MES	Manufacturing Execution System
MIS	Management Information System
OCS	Open Control Software
OLE	Object Linking & Embedding
OPC	OLE for Process Control
PAS	Process Automation System
PID	Proportional Integral Derivative
PIMS	Process Information Management System
PLC	Programmable Logic Controller
ROI	Return on Investment
RTOS	Real-time Operating System
SCP	Supply Chain Planning
TM	Transportation Management
SPC	Statistical Process Control
VAR	Value Added Reseller
WMS	Warehouse Management System
Y2K	Year 2000