Legacy RTOS solutions typically consist of separate real-time systems connected to a Windows-based control system. This configuration is typically fraught with communication reliability problems and introduces a non-deterministic node in the control system.

**INtime® for Windows® Version 4.0**

* Add determinism and real-time tasks to standard Windows platforms

**INtime Features**

- Tried and proven, x86-based, hard real-time OS for cycle times under 50 µs to run time-critical I/Os
- Runs alongside standard versions of 32-bit and 64-bit Windows, up to Windows 7
- Scalable multicore platform support from Intel Atom to Xeon processors for up to 31 INtime RTOS kernels
- Deterministic inter-process communication (IPC) within and across nodes (GOBSnet)
- Dedicated access to I/O interfaces like PCI, Ethernet or USB
- Expandable with new, standalone INtime Distributed RTOS solution

Windows has become the standard for implementing human machine interfaces (HMIs) and for other general purpose operating system (GPOS) functions, but Windows alone cannot provide deterministic support for real-time application needs.

**Complete RTOS for Windows platforms**

INtime for Windows is a complete real-time operating system (RTOS) that provides real-time services to a standard Windows 32bit or 64bit platform. It is a multicore RTOS that enables embedded systems to separate deterministic processes from the Windows task management, maximizing system performance and enabling 100% reliability in operation. One or more processor cores can be dedicated to a specific real-time task.

**Expand your possibilities with INtime**

INtime for Windows allows you to combine real-time and Windows applications on a single platform, eliminating the need for redundant hardware to save on system costs and improving overall system reliability. In addition, applications can now be scaled across multiple cores via the Global Objects network technology (GOBSnet). This allows INtime applications to communicate deterministically across multiple INtime kernels – on a multicore processor or over a network to another processor. Applications can also be expanded to run on multiple system platforms with INtime Distributed RTOS.

**Legacy RTOS solution versus integrated INtime for Windows**

INtime real-time and Windows applications run side-by-side on the same multicore. The real-time systems communicate deterministically among themselves and with Windows. Hence, communication reliability problems and the non-deterministic node are removed.

**Datasheet**

Tel: +31 105298827 Fax: +31 105298828 Email: verkoop@ampliconbenelux.com
Improve productivity and reduce time to market

• INtime uses the same Visual Studio tools as Windows application development and debug.

Familiar development environment
INtime applications are edited, compiled and debugged using the same Microsoft Visual Studio tools that are used to build standard Windows applications. There is no need to purchase or learn how to use other development tools.

The INtime development environment includes Visual Studio project wizards that allow basic applications to be built with simple selections of key code building blocks. These can then be customized as necessary. Integration with the online help system ensures quick access to the details of the INtime real-time application programming interface (API).

The INtime real-time programming environment includes a subset of Win32 API which facilitates the process of porting existing Windows C code based applications to the INtime environment. Any .NET language may also be used to create the Windows part of an INtime application, using the INtime NTX interface library to signal and share data between Windows and the INtime application.

Integrated real-time debugger
Real-time processes and variables are monitored and debugged using the Visual Studio source-level debugger. Real-time developers have access to the most powerful features of the Visual Studio debugger, including conditional breakpoints, variable and register inspection, source-level stepping, and watch variables.

Ring 3 - Protected Debugging
All INtime application code is debugged just like Windows applications. Applications run completely protected in the Windows kernel (Ring 3) where errors like bad pointers, stack overruns and page faults don’t compromise the stability of both the real-time environment and the Windows environment.

Local and Remote Debugging
Debugging can be performed locally or remotely, for the added convenience of debugging real-time applications directly on your workstation or on a remote machine accessible via the network.
Real-time system analysis: INscope

INtime real-time performance analyzer, INscope, facilitates acquisition of precise time and sequence data of real-time threads. INscope is a Windows application that is used to trace the execution of a real-time application. The precise timing of real-time events including thread switches, system library calls, and interrupts is displayed on a logic-analyzer type graphical trace.

INscope includes an API to add custom events to the trace log and precisely control trace trigger points. The INscope API can be used to verify operation of time-critical code, as well as proper sequencing of real-time events.

Dynamic object browser: INtex

INtime Explorer (INtex) gives the developer a convenient way to see and interact with the state of processes, threads, and objects while real-time continues running.

Comprehensive Library Support

Full C and C++ support
The INtime C99-compatible C library and ANSI C++ library include full support for STL. An INtime RtClass library provides a C++ interface to standard real-time objects.

PCI & PCI Express access
INtime applications can easily discover and configure PCI devices for real-time control.

Shared real-time libraries: RSLs
INtime Real-time Shared Libraries are similar to Windows DLLs. RSLs provide a means for multiple real-time processes to share a single real-time code library.

Comprehensive SSE library
Utilizing the Intel Integrated Performance Primitives library (IPP), developers can quickly implement a variety of signal processing functions on the INtime real-time platform.
Partitioning I/O Resources

An essential requirement of running Windows and INtime RTOS alongside Windows on the same platform is to be able to partition the platforms' I/O resources. The real-time system invariably monitors and/or controls external devices that interface to the system via an I/O. The only way that determinacy can be preserved is to ensure that those I/Os are under the direct control of INtime.

I/O Resource Allocation

INtime for Windows allows the I/O devices in the system to be dedicated to INtime or Windows via graphical user interface.

Flexible real-time interrupt isolation

The INtime Device Manager simplifies the process of allocating interrupts between the Windows and real-time environments. It supports APIC and MSI interrupt sources and enables hardware to be reserved for real-time use.

Comprehensive I/O Support

Real-time applications always have direct high speed access to I/O. There is no requirement to learn difficult kernel-mode debugging techniques or create unique device drivers in order to access specialized I/O hardware.

Real-time access to TCP/IP networks

INtime's TCP/IP network includes a BSD standard driver with protocol layer 2 access and Berkeley packet filtering (BPF). Diagnostic tools such as TCP dump are provided to readily capture and display real-time network data. For high throughput performance, INtime real-time applications have direct access to a TCP/IP stack through a standard sockets API to communicate with dedicated Ethernet hardware.

Industrial I/O drivers

Custom or third-party control bus and industrial communication protocols are available and can be easily incorporated into INtime applications, as can SATA disk drives. I/O can be dedicated to a specific instance of INtime in multi-instance installations.

USB client development support

A real-time USB stack supports USB and USB 3.0 host controller interfaces. Using these stacks, INtime applications can directly access USB I/O devices for real-time data acquisition and control. The INtime software development kit includes components and examples to enable the creation of USB client software.
Cut system costs through consolidation

Multicore multiplies performance
One or more cores of a multicore processor can support an instance of the INtime for Windows RTOS with the remaining cores hosting Windows. By allowing you to use your legacy software and dedicating cores to specific real-time tasks such as high-bandwidth motion control or image processing, embedded systems can do more with less hardware, without sacrificing performance and determinism.

Reliable microsecond performance and determinism
INtime software delivers accurate and repeatable microsecond-level response without compromising the safety, security, reliability or performance of your system. Computing intensive applications like advanced processing and graphic applications can run on the Windows side, without affecting the INtime real-time domain.

Replace expensive special-purpose processors with a dedicated processor core
Rather than allocating time-critical tasks to an expensive DSP board, you can run an instance of INtime on a dedicated core and provide the same DSP functionality on one CPU board. Utilizing the Intel IPP library with your INtime applications, you can take full advantage of Intel MMX, SSE, and AVX (SIMD) instructions for DSP-level performance thus optimizing the cost of your hardware and software engineering resources.

GOBSnet opens new possibilities

Application Scalability
TenAsys GOBSnet enables deterministic communication across processes running on multiple INtime kernels via semaphores and mailboxes. Using this feature, processes within an application can pass information to each other across different kernel as if they were running on the same kernel and platform. This enables applications to run on diverse platforms without modifying the code. Thus allowing hardware to be scaled in accordance to computing requirements including adding more real-time components on another multicore processor using INtime Distributed RTOS (as shown in diagram).

Low-cost managed communication
Communication across INtime kernels and Windows is made possible with GOBSnet which provides efficient, reliable, low-cost, managed communication. Now, real-time applications that were running on several platforms can be moved to one multicore processor platform.
System requirements
Minimum requirements for a Windows PC-compatible host running INtime:
- At least 16MB of DRAM available for dedicated INtime + real-time application usage
- 32-bit or 64-bit Windows 7, Windows Vista, and Windows XP
- Any Intel or AMD processor that runs Windows-- including single-core, multicore and hyper-threaded processors
- Intel PRO/100, Intel PRO/1000, Realtek RTL 8xxx - 100/1000
  Broadcom BCM 5xxx - 100/1000 *
  * Visit the TenKeys website or contact us directly for information regarding device driver availability.

Ordering information
INTIME4-DK-HWKEY (INtime Development Kit)
Development license for use on a single Windows computer and two development target run-time licenses. Includes all INtime libraries and APIs; installation and configuration tools; Windows-based performance monitoring, debug and characterization tools; Microsoft Visual Studio wizards, on-line help and debugger plug-in, real-time C/C++ libraries and headers; and numerous sample applications. Also includes one year technical support and maintenance agreement. Support and maintenance for subsequent years available (INTIME-DK-MAINT).

INTIME4-DK-NETSRV (INtime Network Development Kit)
INtime software development license and development target run-time licenses (see INTIME4-DK-HWKEY for content details) configured as a floating network license for six networked development workstations with twelve target run-time licenses. Additional seats available with INTIME4-DK-ADDNET.

INTIME-RT and INTIME-MCRT (INtime Production Run-Time Licenses)
One license is required for each INtime kernel deployed. INTIME-RT for single instance of INtime kernel on system, INTIME-MCRT for multiple instances of INtime kernel on the same system. Volume pricing available with signed redistribution agreement.