Data Sheet

The INtime® RTOS

A scalable, deterministic OS for PC-based embedded solutions

The TenAsys® INtime® RTOS is a hard real-time, event-driven OS with determinism measured in microseconds. Optimized for the x86 architecture, the INtime RTOS is ready to run on off-the-shelf hardware. No board support package (BSP) development is required.

INtime also incorporates a robust networking stack with support for real-time communications using a variety of fieldbus protocols. It is also the industry's first Time-Sensitive Networking (TSN) RTOS, supporting the IEEE 1588 Precision Time Protocol (PTP) to enable truly converged network infrastructure.

Time-Sensitive Networking & Real-Time Communications

INtime was built from the ground up for Time-Sensitive Networking (TSN), starting with support for the IEEE 1588 Precision Time Protocol (PTP) and Intel® Time Coordinated Computing (Intel® TCC).

Intel TCC was first introduced on 11th generation Intel® Core™ (Tiger Lake) and Intel® Atom® x6000E (Elkhart Lake) Series devices and builds a single-clock reference into the processor architecture to provide foundational time synchronization. In INtime software deployments, Intel TCC combines with precision inter-process communications (IPC) and cache allocation technology to improve OS determinism, reduce thread jitter, refine I/O interactions, and empower the RTOS to manage time-aware infrastructure across multiple nodes.

Along with the aforementioned IEEE 1588 PTP, this gives TSN developers a sophisticated time-aware subsystem to build from—even for workload-consolidated systems.

In addition to setting the stage for scalable TSN deployments, INtime also contains drivers for a host of other time-critical communications technologies, including:

- INtime High Performance Ethernet (HPE), a foundation for EtherCAT*, Sercos III*, and Profinet*
- Ethernet
- xHCI USB, PCI*/PCIe*, and serial ports

Furthermore, current INtime RTOS environments remain backward compatible to applications developed for prior INtime versions, preserving software investments over the long term.

Flexible Deployment Options

INtime is adaptable to various deployment scenarios, including:

- **Standalone RTOS:** INtime can operate independently, having full control over hardware resources.
- **Alongside Windows:** INtime can run side-by-side with Windows without a hypervisor. In this model, INtime uses explicit partitioning to reserve resources for itself.
- **Hypervisor-Based:** In this scenario, INtime leverages an embedded hypervisor to manage resources. The hypervisor can host multiple OSs as guests, including Linux*.
- **Multi-core, Multi-instance:** On a multicore processor, each core may run a separate instance of INtime.
- **Distributed:** Multiple INtime RTOS instances can be distributed across multiple networked hosts.

This deployment flexibility facilitates unique system design architectures like workload consolidation that enable deployment of multiple heterogeneous OSs on a single x86 multicore platform to reduce system costs. In these systems, INtime can run on top of a variety of hypervisors and alongside any GPOS.

Regardless of deployment model, INtime guarantees consistent performance and reliability while retaining binary portability with any other installation of INtime software.

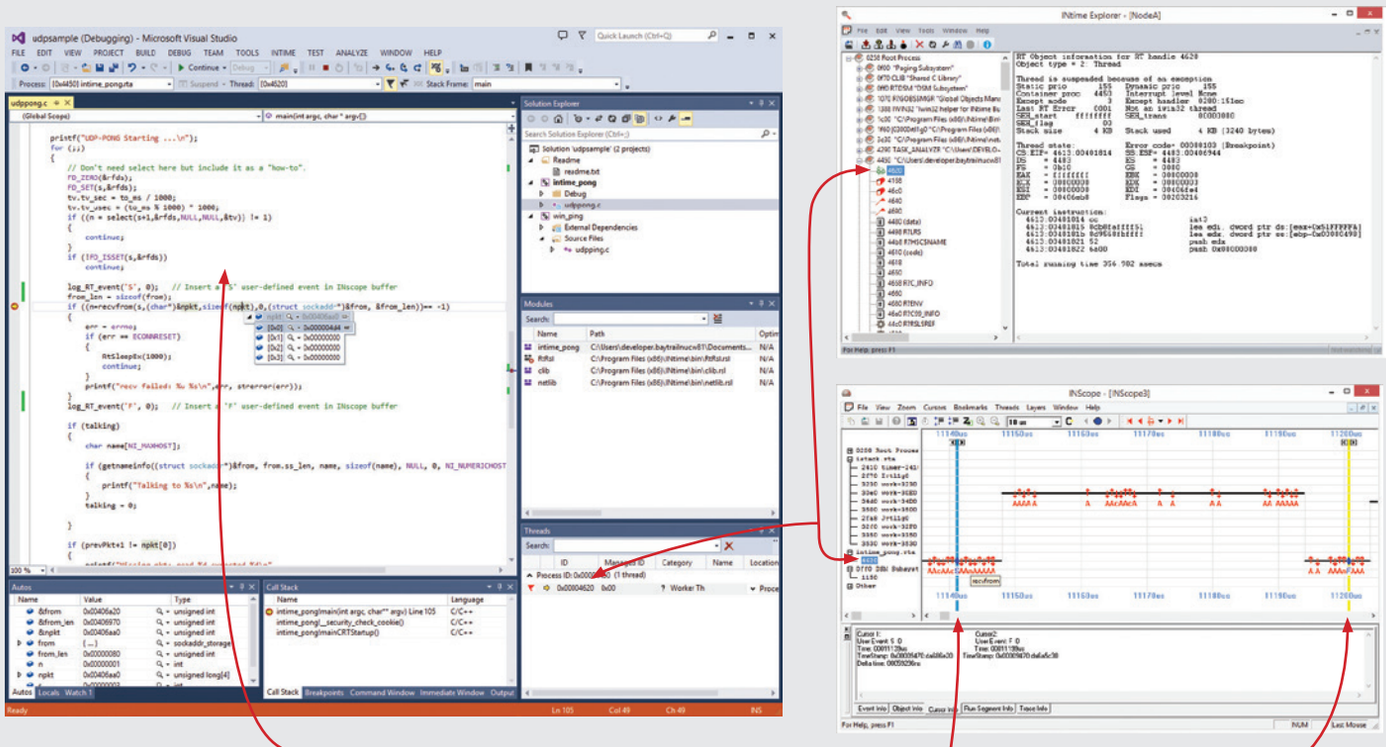
INtime® as a Standalone RTOS

INtime® was conceived as a standalone RTOS with the ability to run independently. It includes all the standard services of a general-purpose OS, which streamlines development. For example, programmers don't have to create custom code in order for INtime to manage multiple system tasks simultaneously—that functionality is native to INtime out of the box.

But at its core, INtime is a hard-real-time RTOS built on an object-based architecture in which the kernel issues and maintains object data structures and related system calls. The RTOS executes processes in protected user mode (Ring 3), delivering the single-digit microsecond latencies and minimal jitter industry expects from a proven RTOS.

The INtime® SDK

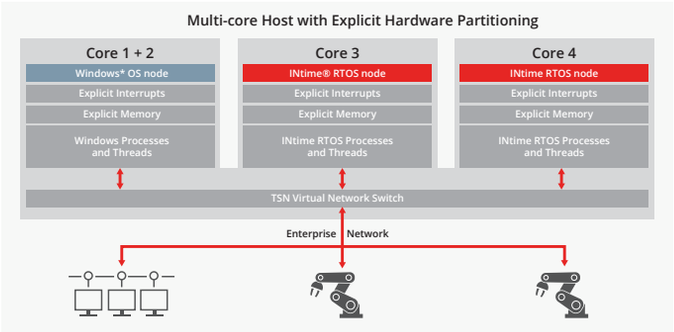
TenAsys offers a wide range of development tools for use with the INtime RTOS through the INtime Software Development Kit (SDK). The INtime SDK contains a software debugger, system debug monitor, dynamic object browser and multi-thread debugger, time analyzer, fault manager, and more. It also integrates with the familiar Microsoft* Visual Studio* environment and supports migration libraries for porting legacy code written for Windows* real-time extension APIs.



INtime® for Windows

Unlike many real-time extensions that are tethered to a host OS, INtime doesn't require any other OS or external management system. However, it can run in tandem with Microsoft* Windows* with, but also without, a hypervisor.

To run alongside Windows without a hypervisor, INtime employs explicit partitioning. Explicit partitioning is a unique product of INtime's built-in embedded virtualization capability that allows the RTOS to be configured such that INtime and Windows instances can occupy separate processor cores, memory spaces, and I/O on the same host.



Depicted here is an example of embedded virtualization at work in an explicitly partitioned system that dedicates two cores each with their own processor core, memory, and I/O resources to time-critical control tasks and an additional two cores to non-time-critical network, database, and user interface functions.

INtime communicates with Windows using the TenAsys NTX dynamic link library (DLL). This inter-process communication (IPC) mechanism can interface with multiple instances of INtime.

Symmetric vs. Asymmetric Multiprocessing

Multicore processors were developed to increase processing power in computers when the clock rate reached a practical limit. To provide maximum utilization of these multiple cores, GPOS like Microsoft* Windows* uses Symmetric Multiprocessing (SMP) to distribute the workload dynamically or "on the fly" to the multiple cores in a system. For many applications this works just fine and increases overall system throughput.

Hard-real-time applications, however, demand deterministic timing for results. To be deterministic, work needs to be prioritized (free from interruptions and interference) and reliable (fixed to hardware resources) with coordinated critical services that scale over the nodes of the system. By definition, this is an asynchronous and asymmetric problem ideally solved with an appropriate Asymmetric Multiprocessing (AMP) system solution.

SMP supports application scalability using processors with different numbers of cores and enables the easy addition of new applications to the workload, but there is a performance hit and some unpredictability in core management overhead. Unlike SMP systems that typically run one instance of a GPOS or node on multiple cores, AMP like that available in the INtime RTOS architecture assigns a different OS instance or node for each core. When each node has a tailored set of characterized and debugged processes, the workload does not change and therefore the application will have deterministic timing.

The TenAsys® INtime® RTOS can support multiple schedulers and multiple environments in an AMP system.

Hypervisor-Based Deployment

While INtime doesn't need a hypervisor it can be adapted to work with a real-time embedded, including leading solutions like the RTS Hypervisor* from Real Time Systems* and others. When running on such a hypervisor, INtime defers to the hypervisor for platform resource allocation and management.

Since hypervisors can isolate and host multiple operating systems in their own virtualized environments, using one permits INtime instances to run alongside a variety of other OSs, including Linux. In hypervisor-managed environments, INtime runs as a paravirtualized instance to provide hardware-like interfaces for:

- Utilization of shared memory segments managed by the hypervisor
- Event signaling through a library provided by the hypervisor
- Communication with other OSs via a TCP stack, facilitated by a virtual switch in the hypervisor

When paired with the right hypervisor, the INtime RTOS can run at near-native performance to meet an application's real-time requirements, even in virtualized environments.

Multiple Instances on a Multicore Processor

Even without a hypervisor, embedded virtualization features allow multiple instances of INtime to operate simultaneously on a multicore processor host. The RTOS's asymmetric multiprocessing (AMP) architecture transforms multicore hosts into distinct processing nodes where each instance of INtime is treated as a node with its own dedicated resources, including a processor core, RAM, and I/O. This approach ensures critical response times are met by protecting the resources of each node from outside interference and adds a layer of security and reliability.

In these deployments, the multiple instances of INtime interact with each other using global objects (GOBS). An extension of the RTOS's object-based architecture, GOBS leverage a shared handle for each object provided by kernels on the host. This allows different INtime instances running on different nodes to access each other's objects seamlessly, without requiring any modifications to application code.

Distributed Systems Manager (DSM)

The Distributed Systems Manager (DSM) is a resource that helps monitor and manage a distributed system composed of multiple INtime® nodes. The DSM can track the status, health, and performance of each node and its processes while also communicating failures, terminations, and dependencies between processes on any node of a system.

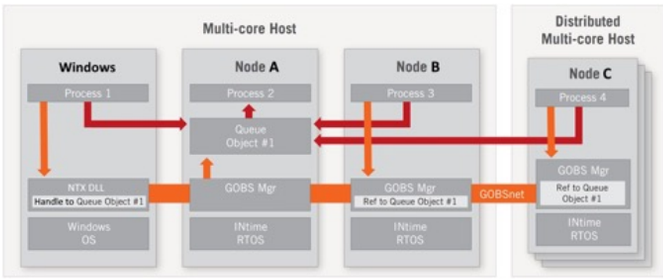
Specifically, the DSM:

- Establishes sponsor and dependent processes, in both Windows* and RT
- Establishes Windows hosts and INtime clients
- Establishes Windows processes and INtime resources
- Tracks these relationships
- Notifies when a process terminates or fails
- Notifies when the real-time client or Windows host terminates, fails, or sleeps
- Cleans up resources owned by a Windows process

DSM processes communicate with each other over Ethernet, PCI, or RS-232 serial links. They work in parallel with GOBS Managers and GOBSnet utilities to support distributed INtime software deployments.

INtime® Distributed RTOS

In addition to running multiple instances on the same host, multiple INtime instances work in parallel while running across multiple hosts. In these distributed application deployments, node-to-node communications are managed by the TenAsys GOBSnet inter-process communications (IPC) mechanism. GOBSnet extends the GOBS architecture to allow instances of INtime running on different hosts to access each other's global objects as if they were on the same processor.



The red arrows in the figure above represent a shared inbound queue for Node A. The orange arrows between the A, B, and Windows Nodes represent references to the handle of Object #1 by different global objects (GOBS) Managers on each node. The GOBSnet bar located between Nodes B and C represents the same handle referencing function as the other orange lines, with the principal difference that the communication request is being made by a distributed host located elsewhere on the network.

GOBSnet provides a robust suite of communication services so that all GOBS system-wide can be catalogued, searched, and utilized regardless of their location—within the same core, on different cores, or even spread across different hosts.

INtime RTOS kernels and the Distributed Systems Manager (DSM) orchestrate GOBSnet connections across different cores or systems, a design choice that offloads this function from the application and developers who would otherwise have to concern themselves with the intricacies of objects and their locations.

System Requirements

Minimum requirements for INtime® for Windows* on a Windows PC host:

- x86-compatible platform
- At least 16 MB of RAM available for INtime RTOS kernel plus RAM for real-time applications
- Operating System & Software Requirements:
  - INtime for Windows:
    - Any 32- or 64-bit version of Windows* 11 or 10 or Windows* Server 2019, 2016, 2012 R2, 2012, or 2008 R2
    - Windows* Embedded versions of the above releases

Minimum requirements for INtime® Distributed RTOS:

- x86-compatible platform with standard BIOS services (UEFI or Legacy)*
- At least 16 MB of RAM available for INtime RTOS kernel plus RAM for real-time applications
- PATA, SATA, NVMe, or soldered eMMC flash drives to load the INtime RTOS and applications
  - USB flash drives can only be used for installation
- Compatible networking hardware:
  - Intel* I210/I2xx, Intel* PRO/1000, or Intel* PRO/100
  - Realtek* RTL 8xxx-100/1000
  - Broadcom* BCM 5xxx-100/1000

**Virtualized deployments running INtime plus Linux* or Windows require at least two logical processor cores or hardware threads*

**Contact TenAsys for hypervisor-based system configuration requirements*

FEATURES	BENEFITS
Deterministic, Event-Driven Processing	Reliable foundation for embedded design
PC Compatibility	Compatible with PC hardware hosts and software for optimized performance and cost
Flexible Topology	Deployable in standalone applications, alongside Windows or other GPOS, workload-consolidated, or distributed systems
C++ and Boost C++ Library Support	Conforming to C++ 2020 standard and enabling use of advanced Boost C++ libraries.
Kernel Services in Object-Based RTOS	Fast development of scalable, reliable solutions using an object-based programming methodology
Deterministic Inter-Process Communication (IPC)	Reliable communications between OS processes across nodes and hosts
Dedicated I/O Interface Partitioning	Explicit allocation of I/O resources eliminates conflicts between OS instances and applications
Fully Featured Real-Time Networking Stack	Supports TCP/IP, Time-Sensitive Networking over Ethernet, and other real-time fieldbus protocols
SIMD Library Support	Uses Intel Math Kernel Library (Intel® MKL) and Intel* Integrated Performance Primitives (Intel® IPP) for optimal mathematics performance in SIMD operations

INtime® Software Order Codes

INtime is licensed software. Licenses can be purchased for a one- or two-instance deployment and installed on hardware with more than two logical processors (INtime for Windows) or multiple PC systems (INtime Distributed RTOS). Licenses can be purchased for single- or multiple-instance deployments as described below.

- **INTIME-RT** – INtime for Windows runtime incorporation fee for redistribution of base operating system and derivative works. This license is used for one INtime kernel instance per system. Does not include a license for Microsoft Windows.
- **INTIME-MCRT** – INtime for Windows multi-core runtime incorporation fee for redistribution of base operating system and derivative works. This license is used for two or more INtime kernel instances per system. Does not include a license for Microsoft Windows.
- **RTOS-RT** – INtime Distributed RTOS runtime incorporation fee for redistribution of base operating system and derivative works. This license is used for up to two INtime kernel instances per system.
- **RTOS-MCRT** – INtime Distributed RTOS runtime incorporation fee for redistribution of base operating system and derivative works. This license is used for three or more INtime kernel instances per system.
- ***-KEY** – Append -KEY to above for USB dongle based license.

Contact Us

With a distributor network that spans the globe, TenAsys has local support to help you launch your next design.

TenAsys Corporation US

Phone: (877) 277 9189
Fax: +1 503 748 4730
Email: sales@tenasys.com

TenAsys Europe GMBH

Phone: +49 89 45 46 9 47 – 0
Fax : +49 89 45 46 9 47 – 07
Email: europe-office@tenasys.com