IntervalZero

ProductBrief

March 2026

MaxRT wRTOS™

Overview

MaxRT wRTOS™ is a key component of the MaxRT product family, the new real-time solutions to address the next generation of industrial control systems. wRTOS provides determinism or hard real-time on multi-core x64 processors while co-resident with the Windows operating system. wRTOS consists of a separate real-time Subsystem (RTSS) that schedules and controls all RTSS applications independently of Windows.

Symmetric multiprocessing-enabled wRTOS takes full advantage of 64-bit memory and performance capabilities. Uniquely, the wRTOS scheduler enables embedded real-time applications to directly access the 512GB of addressable physical memory available on 64-bit Windows.
This is critical to modern day real-time systems and represents a gigantic leap from the 4GB physical memory limit of traditional 32-bit Windows systems. The 4GB barrier has stymied innovation in many industries that depend on real-time systems and that require memory access far beyond 4GB.

Determinism

- Guaranteed Precision – set timer periods down to 1 microsecond, and Interrupt Service Thread (IST) latencies of less than 10 microseconds
- Separation from Windows – Windows processes cannot interfere with real-time applications
- Scalability – one scheduler is used across all real-time processors. Symmetric multiprocessing (SMP) aware scheduler utilizes both priority-driven and pre-emptive algorithms to ensure critical thread context switches; and yields to threads of high priority occur in the sub-microsecond range

Control

- Flexibility to configure how much or little processing capability is used for real-time processes (1 to 63 processors)
- Full control of real-time process threads with the ability to load balance as needed. wRTOS provides the ability to set thread and interrupt affinities
- Peace of mind if Windows issues a STOP message or shutdown; real-time applications have the ability to continue running to safety shutdown

Simplify

- Use a single operating system for applications. wRTOS is supported on Windows 11
- Use commercial off-the shelf (COTS) target system; no special hardware required
- Use one development environment - Visual Studio 2022
- Use common languages (C/C++) for Windows and real-time applications
- Use common Win32 API; same code can be run as a Windows or real-time process
- Use managed code for Windows application and still communicate with your real-time applications
- No driver model to follow; real-time process can talk directly to hardware
- Use standard IPC communication between Windows applications and real-time processes (events, mutexes, and semaphores)
- Use shared memory between Windows and real-time process for sharing of data

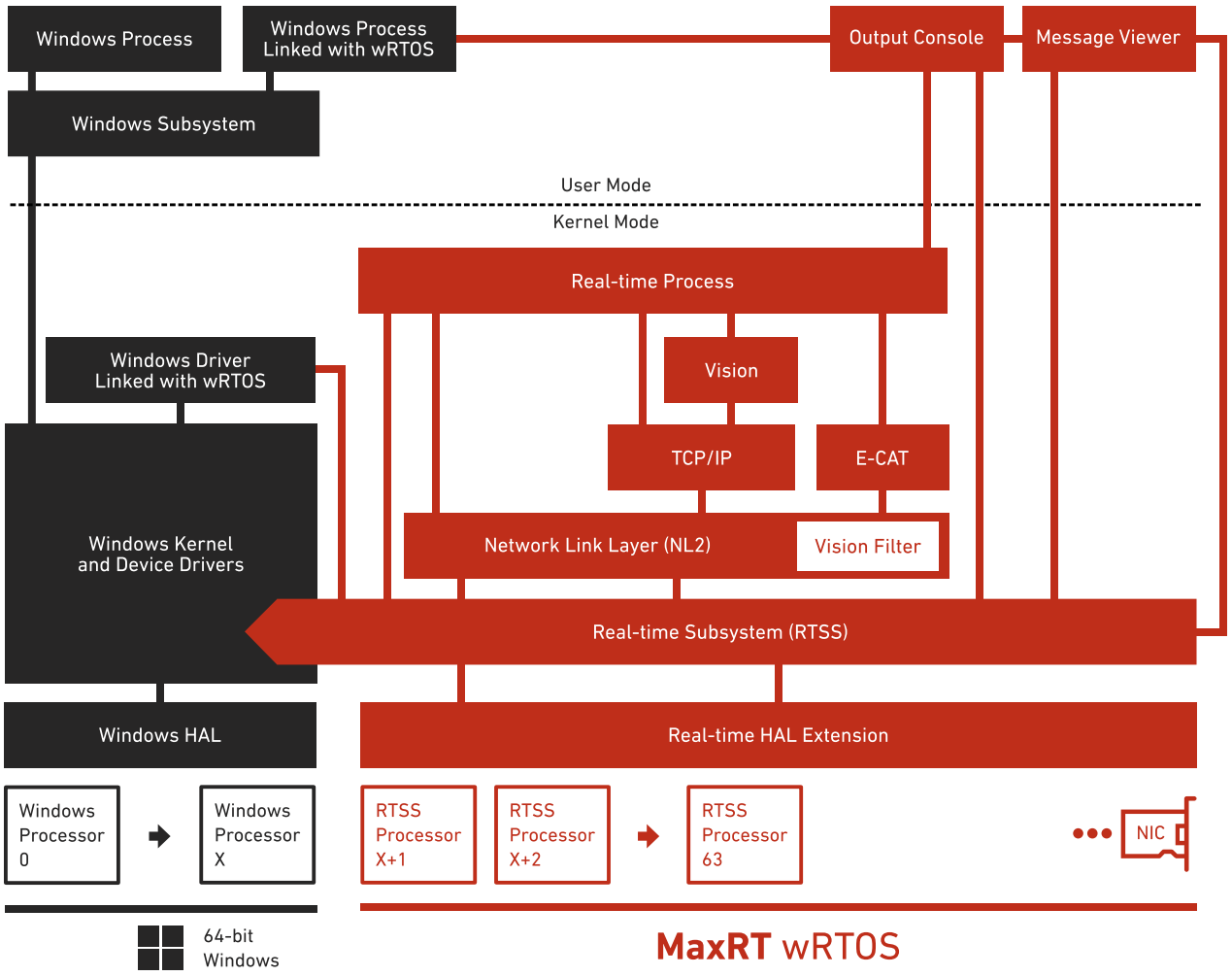
Reduce Costs

- Eliminate additional system to perform the HMI
- Eliminate proprietary controller and communications cards
- Improved asset utilization: Take advantage of underused multi-core capacity
- Reduced manufacturing costs and fewer physical parts

Improve Efficiency

- Eliminate some inventory costs and reduce maintenance costs
- Field upgrades are accomplished through software download rather than board replacement

Architecture



Key Features

Real-time Runtime

- Scalable from 1 to 63 real-time processors
- SMP aware scheduler utilizes both priority-driven and pre-emptive algorithms to ensure critical thread context switches; and yields to threads of high priority occur in the sub-microsecond range
- Configurable thread and interrupt affinity
- Configurable timer period
- Ability to attach to line-based and message-based (MSI/MSI-X) interrupts
- Shutdown handling on Windows STOP or shutdown

- Deterministic memory
- Access to Windows file system and registry
- Ability to set Search paths for process creation and loading of RTDLLs
- Dynamic-link library support through RTDLLs, which can be loaded implicitly or explicitly
- Ability to profile application behavior by monitoring internal objects and custom events
- Real-time Inter Process Communication between Windows user processes or Windows kernel drivers and real-time processes
  - ✓ Native and managed interface for 32-bit or 64-bit Windows processes
  - ✓ Objects available: events, mutexes, and semaphore
  - ✓ Data sharing through shared memory
- Windows user groups for limiting access to wRTOS features

Tools and Utilities

- Analyzer – to output information about the current system environment
- Settings tool – configure the subsystem
- Control Panel – control the subsystem
- SRTM – view system timer to timer handler response on a given core
- Latency View – view and compare system timer response latencies on multiple cores at the same time
- Task Manager – display a list of running RTSS processes and Windows processes and drivers linked to MaxRT wRTOS™, also display pre-process, -thread, and –processor CPU usage
- Message Viewer
- Monitor – to trace the behavior of your real-time applications
- RtObjects – view internal objects and states
- RtMSpaces – view internal memory allocations
- wRTOS Server – to display and/or logs print messages from all wRTOS applications or RTDLLS.
- Network Link Layer (NL2) *
- Network Relay
- Virtual Network – point to point connection between Windows and RTSS
- Utilities (RtssArp, RtssIpConfig, RtssPing, and RtssRoute)

Software Development Kit

- Headers and libraries for application development
  - ✓ Real-time API (RTAPI) similar to Windows Win32 API
  - ✓ Real-time Kernel API (RTKAPI)
  - ✓ Real-time Network API – Network relay API, NIC Driver API, NL2 Filter Driver API, NL2 API, TCP/IP API…
  - ✓ Real-time Network Driver API (RTNDAPI)
  - ✓ Managed API – Subsystem setup and configuration through managed code interface
  - ✓ Configure and Control (RTFW) API – Subsystem setup and configuration through C/C++ interface
- Microsoft Visual Studio 2017, 2019 and 2022 support
  - ✓ Wizard for application and dll development
  - ✓ API Code snippets
  - ✓ C-Runtime support
  - ✓ Local and remote debugger support via launch within Visual Studio
  - ✓ Local and remote attach support
- Microsoft WinDbg extension and RTSS symbols
- Sample source to show basic concepts

Product Documentation

- Documentation consisting of installation and user guides, API references, and details on real-time programming concepts

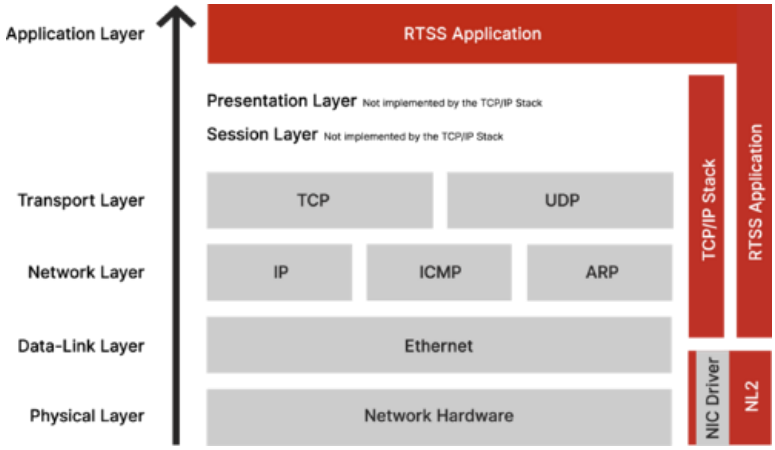
Additional Purchasable Features

MaxRT wRTOS™ Basic Networking

It provides networking capability through a base component called the Network Link Layer (NL2) and a set of optional protocol components above the NL2.

These components run within the RTSS environment:

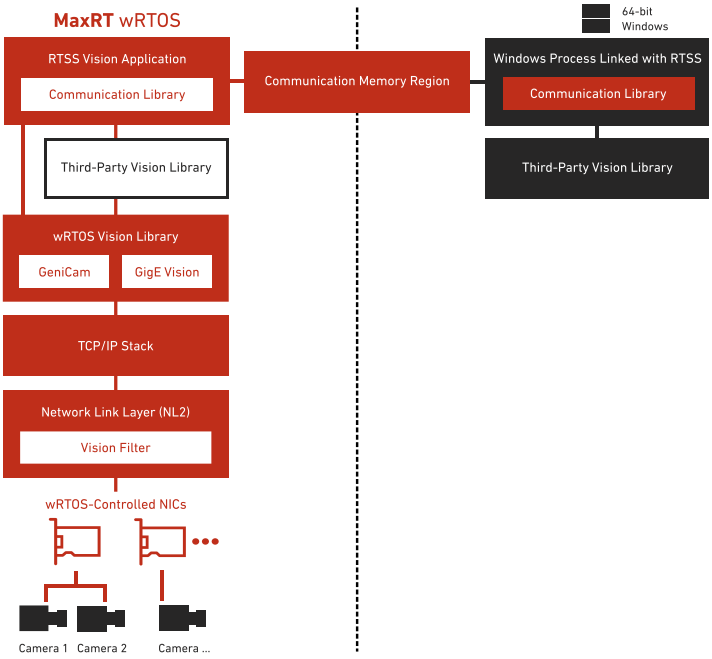
- TCP/UDP/IP networking for wRTOS processes
- Support for IPv4 and IPv6
- Winsock support



MaxRT wRTOS™ GigE Vision

MaxRT wRTOS™ GigE Vision provides functionality for using GigE Vision Cameras within the real-time wRTOS environment:

- Real-time GigE Vision filter driver
- Camera Setup Tool
- Real-time GigE Vision Interface & Communication library
- Built version of OpenCV for use with RTSS Vision applications



MaxRT wRTOS™ Fieldbus E-CAT

wRTOS Fieldbus provides an EtherCAT MainDevice (Master) and tools for wRTOS.

- Tools for configuration, diagnostics, analysis and ESI import
- Auto-discovery & Auto-Configuration deliver plug-and-play feature
- Support all Class A features (FoE, EoE…)
- C/C++ API
- Implemented as a process, allowing customers to debug their application without stopping the EtherCAT MainDevice (Master)
- Multiple MainDevice (Master) Option
- Hot Connect Option
- Cable Redundancy Option
- High-speed Option down to 100 us time cycle

EtherCAT®