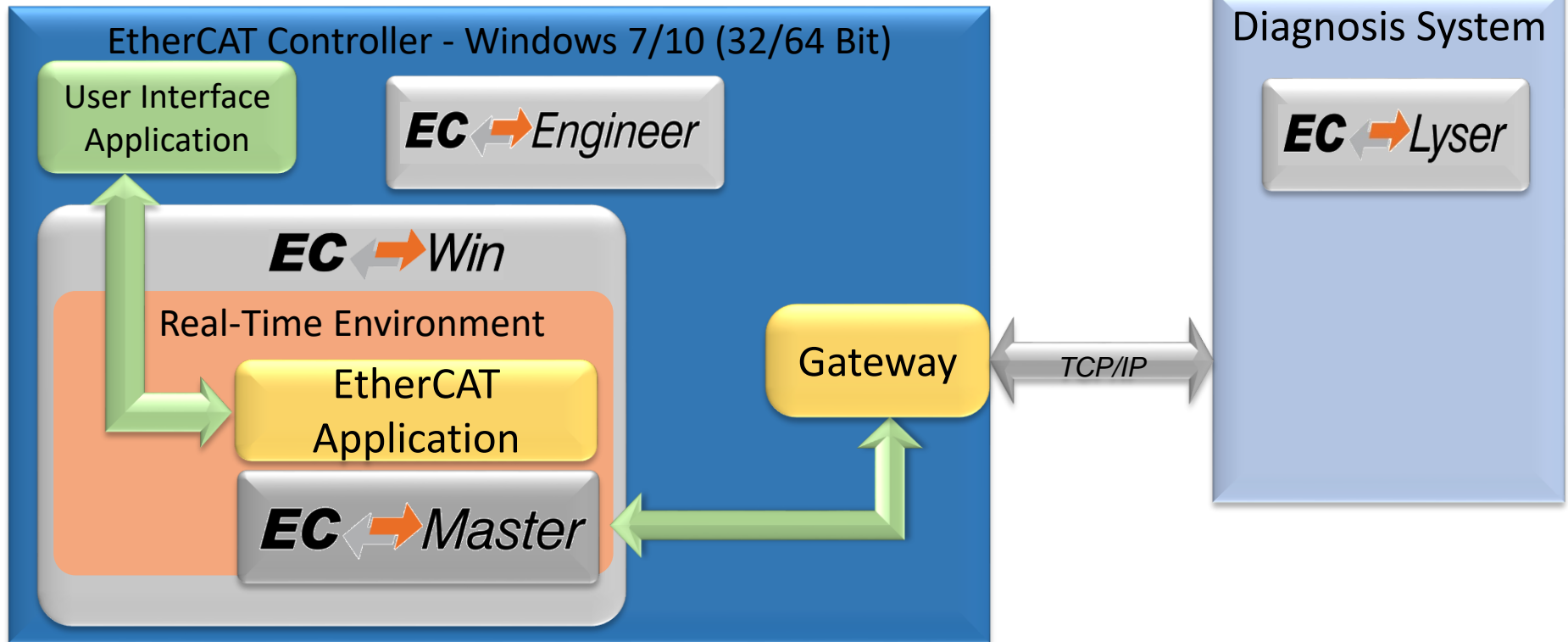




Windows EtherCAT[®] Real-Time Platform

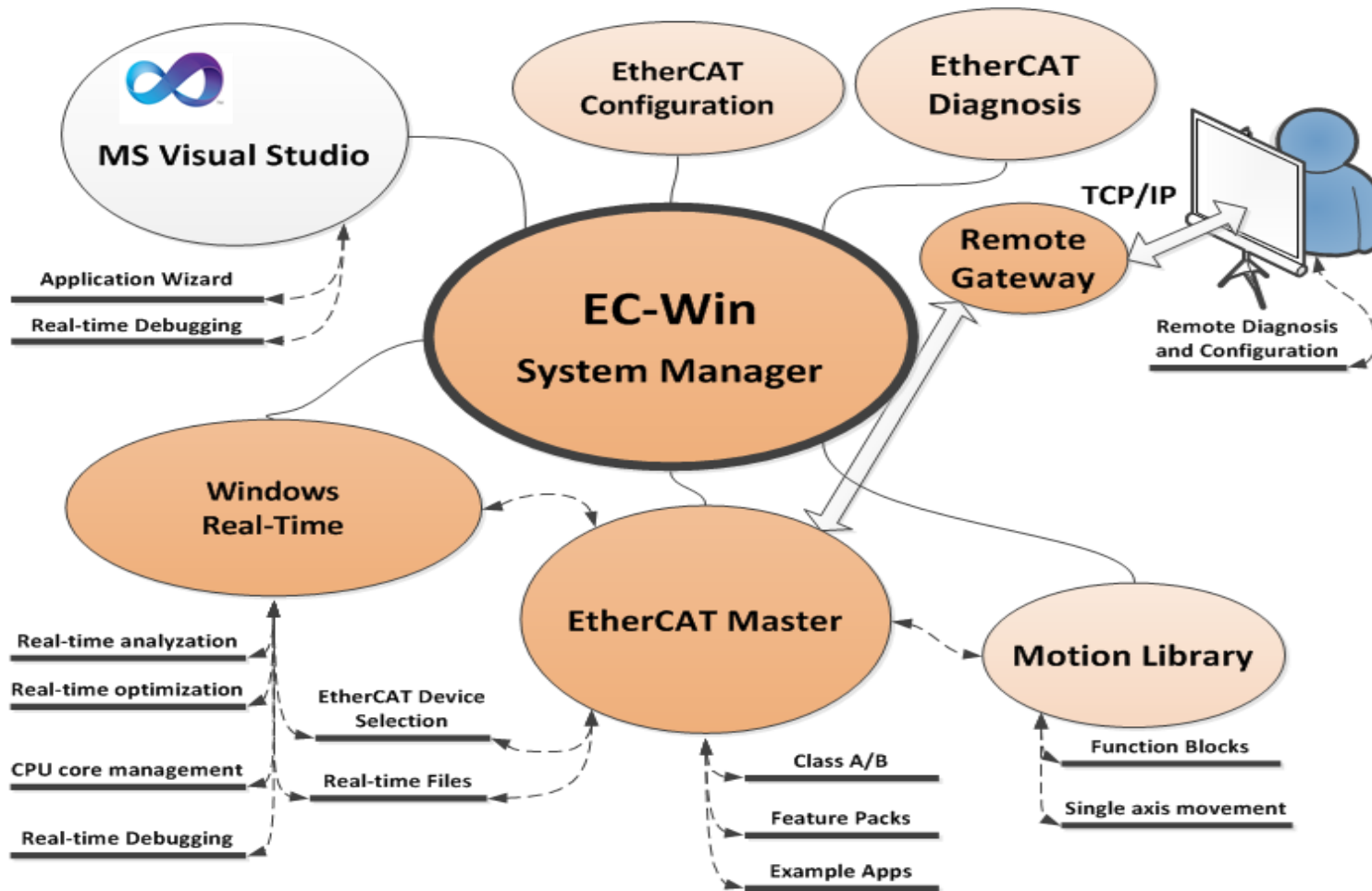
Technical Presentation

EC-Win System Architecture



- Windows: All applications are non deterministic
- Real-time part runs on separate real-time environment

EC-Win: Fully integrated Windows EtherCAT platform



- Windows Real-time extension (standalone product: RT-Win)
- EtherCAT Master Stack (standalone product: EC-Master)
 - Running in the real-time environment
- System Manager
 - The whole development process is controlled from within this easy-to-use tool
- Visual Studio Application Wizard
 - Easily create own EtherCAT projects based on shipped demos
- Remote TCP/IP Gateway to connect external tools with master
 - No extra IP address or network bridging required!
 - Available for acontis tools (EC-Engineer, EC-Lyser) and for customer's own tools

- **EC-Engineer: EtherCAT Configurator Tool**
 - Seamlessly integrated into EC-Win System Manager
- **EC-Lyser: EtherCAT Diagnosis Tool**
 - Seamlessly integrated into EC-Win System Manager
- **EC-Motion: C++ Motion Library**
 - Example application integrated into EC-Win System Manager
- **EtherCAT Master Feature Packs**
 - Hot Connect
 - Cable Redundancy
 - TCP/IP Remote API (to connect with Remote Gateway)

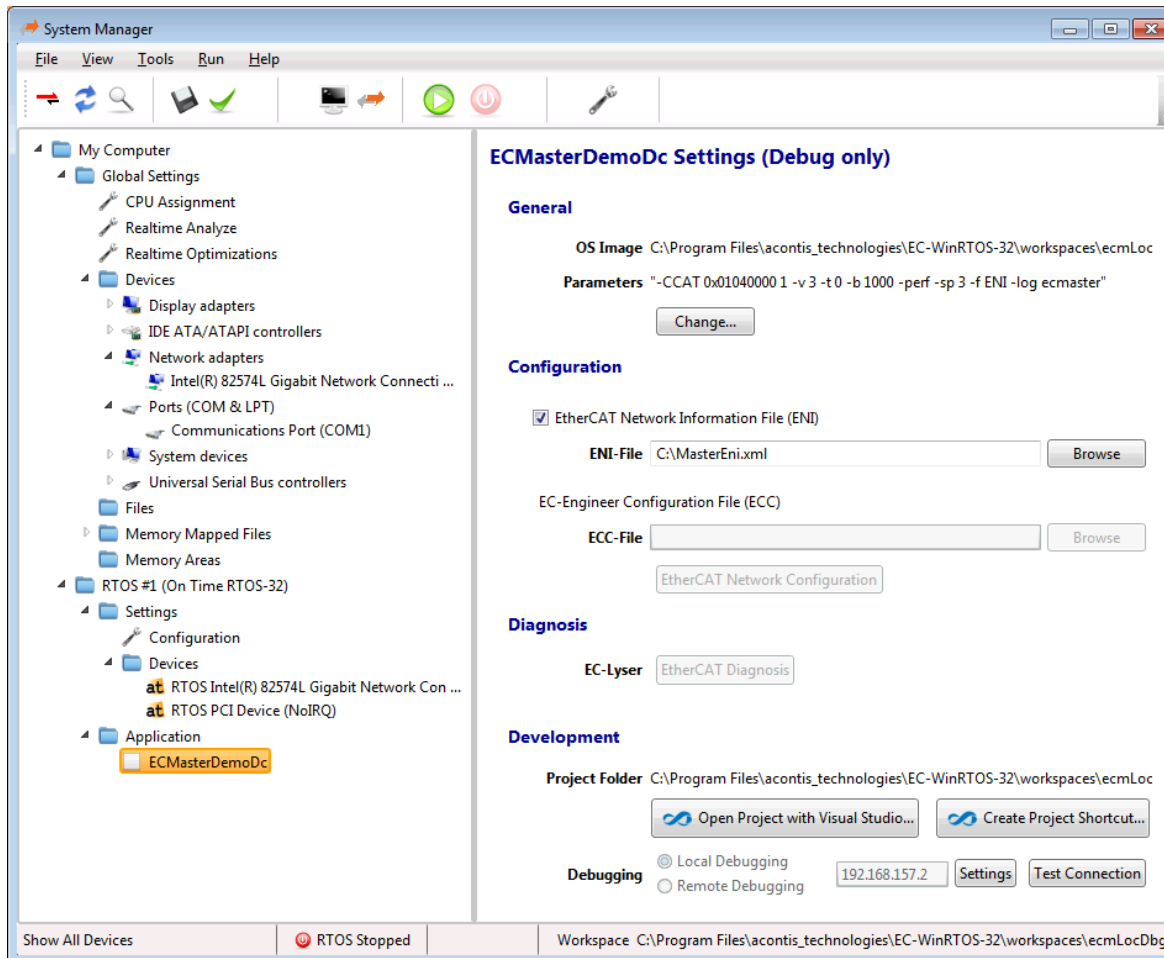
EC  ***Win***

System Manager

Configuration and Development Management Console

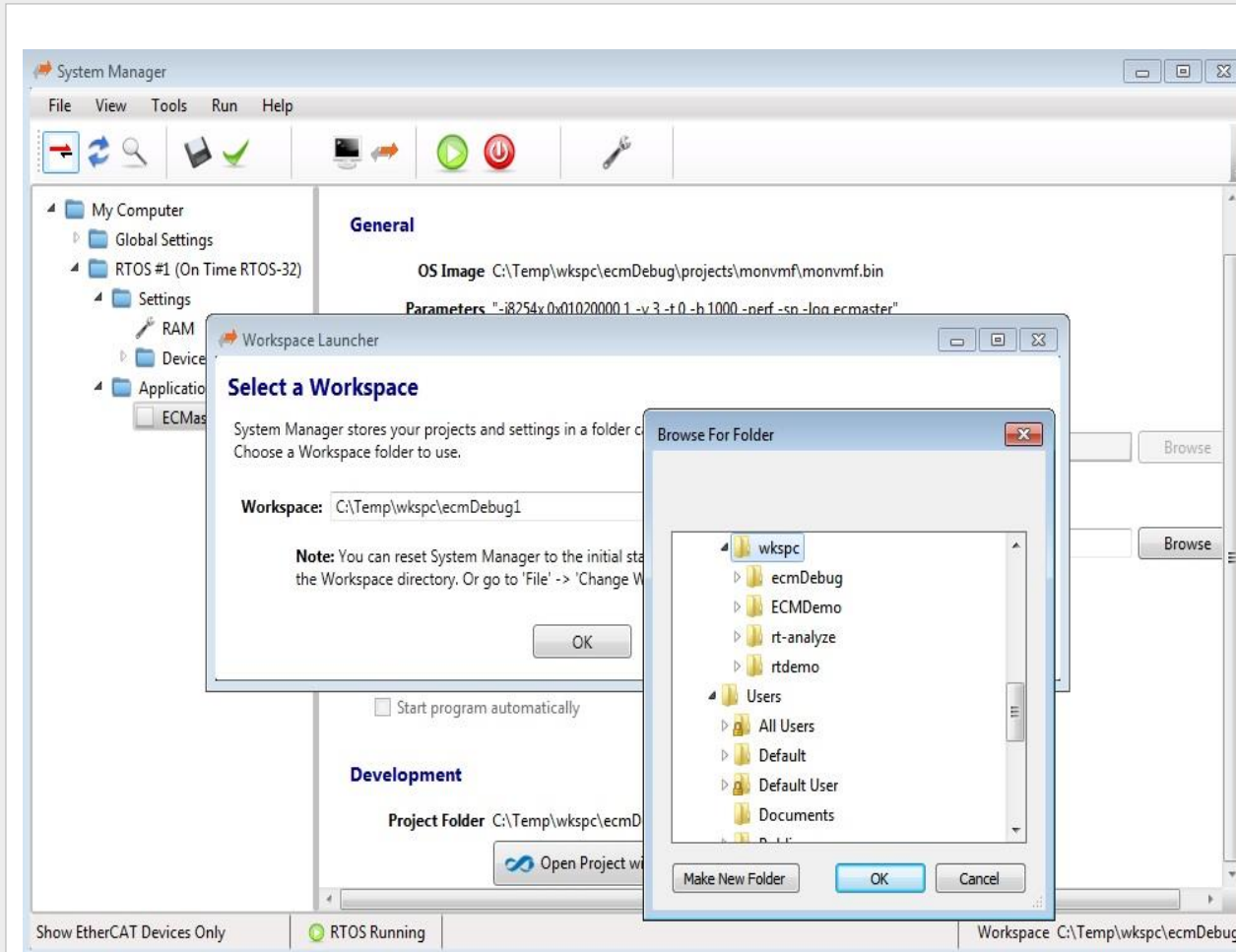
System Manager: Management Console

One GUI for all major development tasks



- System Configuration
 - RAM size for real-time part
 - Devices to be used by real-time
 - Etc.
- Launch Panel
 - start/stop the real-time part
 - ...
- Select and run pre-compiled demos
- Visual Studio Integration
 - Create VS projects based on demo applications
 - Local/remote debugging

System Manager: Workspaces

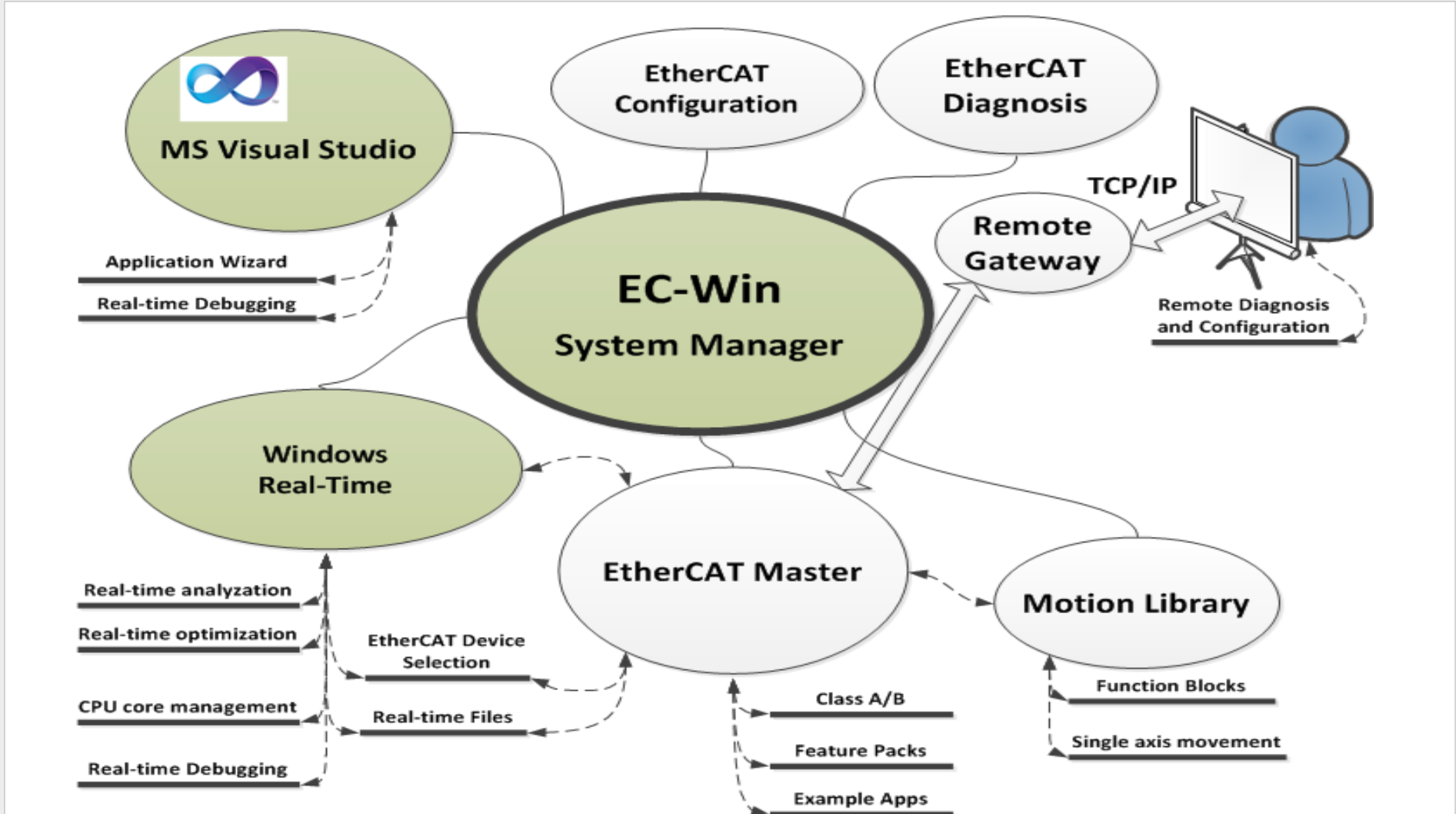


- Manage different projects within different workspaces
- Store all settings
 - Hardware configuration
 - Software configuration
 - Memory areas and files
 - Configuration files
 - Application
- Store source code
- Clone Workspace

EC  ***Win***

Real-time Environment (RTENV)

Real-time related components in EC-Win



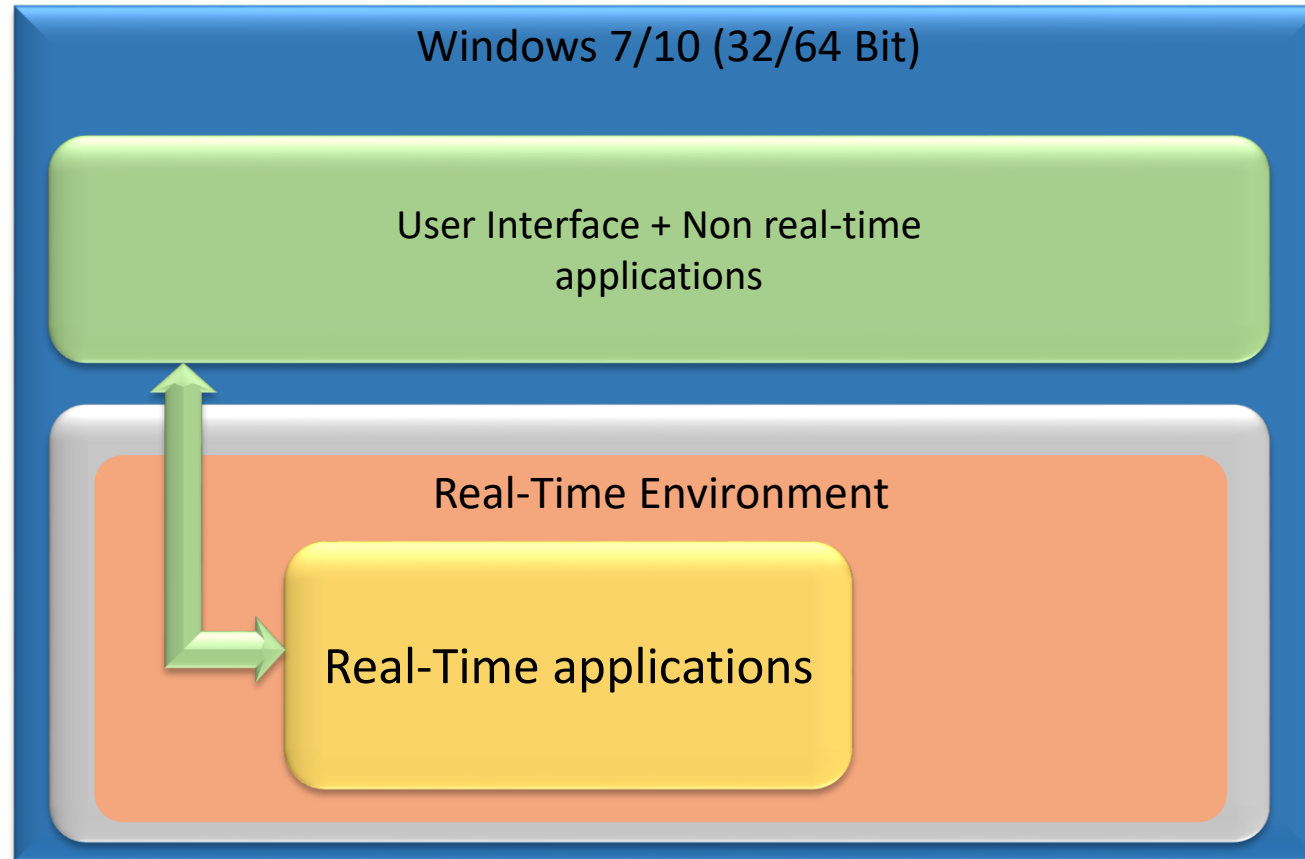
- Runtime environment to run real-time applications
 - Hardware environment (CPU cores, RAM, PCI devices, ...)
 - Software environment (threads, semaphores, scheduler)
- Communication (Windows <-> Real-time environment)
- Development and Debugging tools to create/debug applications
 - VS Project Wizard
- Development Console: System Manager

EC  ***Win***

Real-time Environment (RTENV)

Software

The Real-Time Environment



- RTENV = real-time scheduler + timer + run-time environment
- Option 1: Embedded approach
 - Run the RTENV within Windows context (memory area)
 - Re-use (non time-critical) parts of Windows, e.g. memory management (malloc etc.)
 - not available anymore after BSOD
 - No memory isolation from Windows
 - RTENV may corrupt Windows and vice versa!
- Option 2: Isolated approach
 - Use a full-featured standalone RTENV (including memory management)
 - Full isolation between Windows and RTENV
 - RTENV cannot corrupt Windows and vice versa!
 - **Used by acontis!**

- RTENV based on **On Time RTOS-32** real-time operating system
- The fastest x86 RTOS on the market!
 - Product of On Time Informatik in Germany
 - Quality and reliability through thousands of existing customer applications
- EC-Win includes On Time RTOS-32 license!
- Scalable Solutions can be built
 - Level 1: Embedded Controller without GUI: use native RTOS-32
 - Level 2: Embedded Controller with GUI: use native RTOS-32 and PEG library
 - Level 3: High End Controller with powerful Windows 7 GUI: use EC-Win/RT-Win

- Hard real-time performance (extremely short latencies).
 - Optimized for x86 architecture and highest performance
- Inter-Task Communication
 - Semaphores: counting, binary, event, resource and mutex semaphores, Critical Sections
 - Mailboxes (queues or FIFO buffers). Priority messages can be sent
 - Message passing (synchronous message exchange)
- Deterministic and fast real-time Scheduler
- Interrupt handlers can be written in C/C++ language

- Development and Debugging using Visual Studio (VS2005 ... VS2015)
- Win32 compatible API (e.g. almost all Win32 thread API)
 - Useful for porting of existing Windows applications
- Symmetric Multicore support: use multiple CPU cores for the real-time
- Supplemental Modules
 - CPU load analysis
 - Performance profiler

On Time RTOS-32 real-time performance

Outstanding Real-time performance!

1.4 GHz Pentium IV	Operation
0.07 usec	Round-Robin task switch
0.20 usec	Semaphore task switch
0.18 usec	Semaphore Signal
0.10 usec	Semaphore Wait
0.39 usec	Task activation (Signal, Wait)
0.12 usec	Store data in a mailbox
0.18 usec	Retrieve data from a mailbox
0.34 usec	Task-to-task communication
0.38 usec	Task-to-mailbox-to-task communication

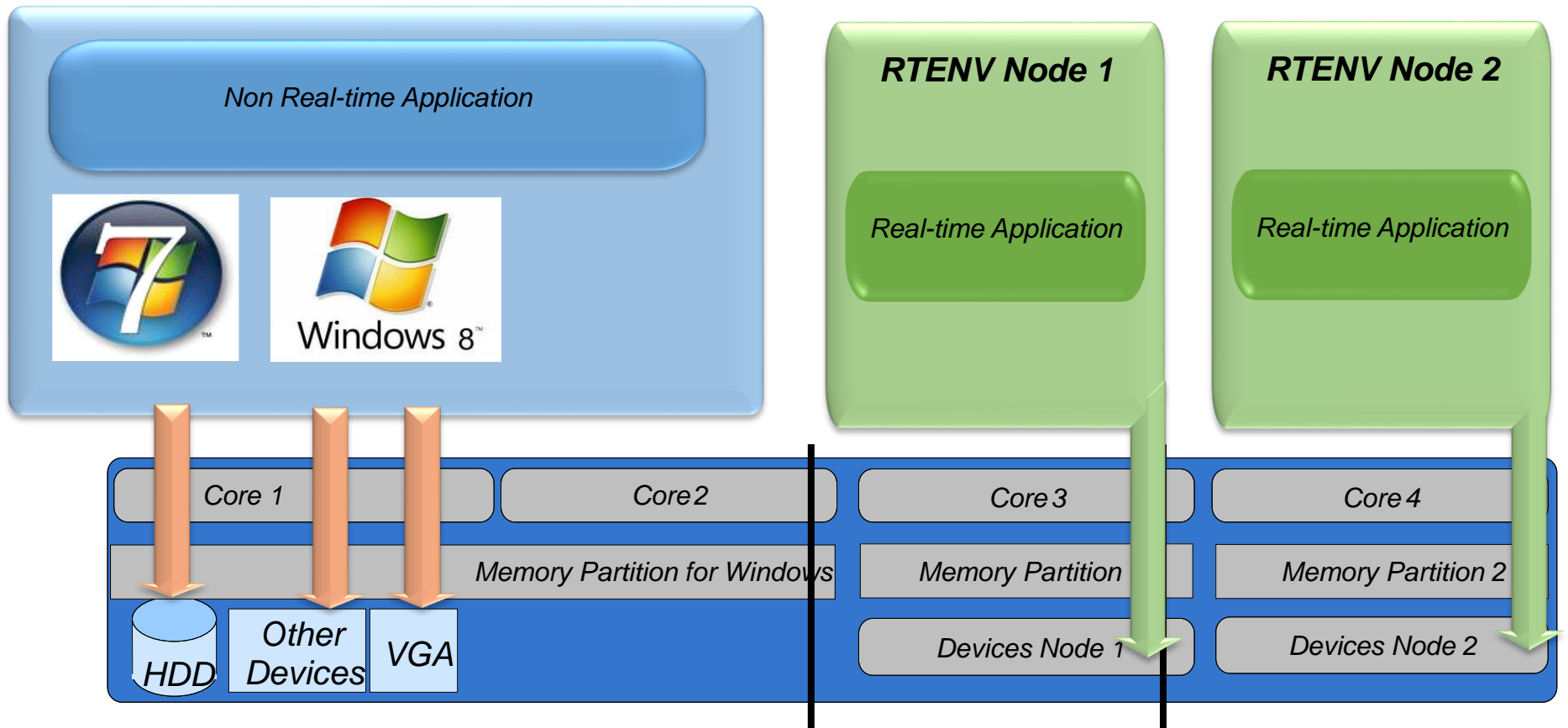
EC  ***Win***

Real-time Environment (RTENV)

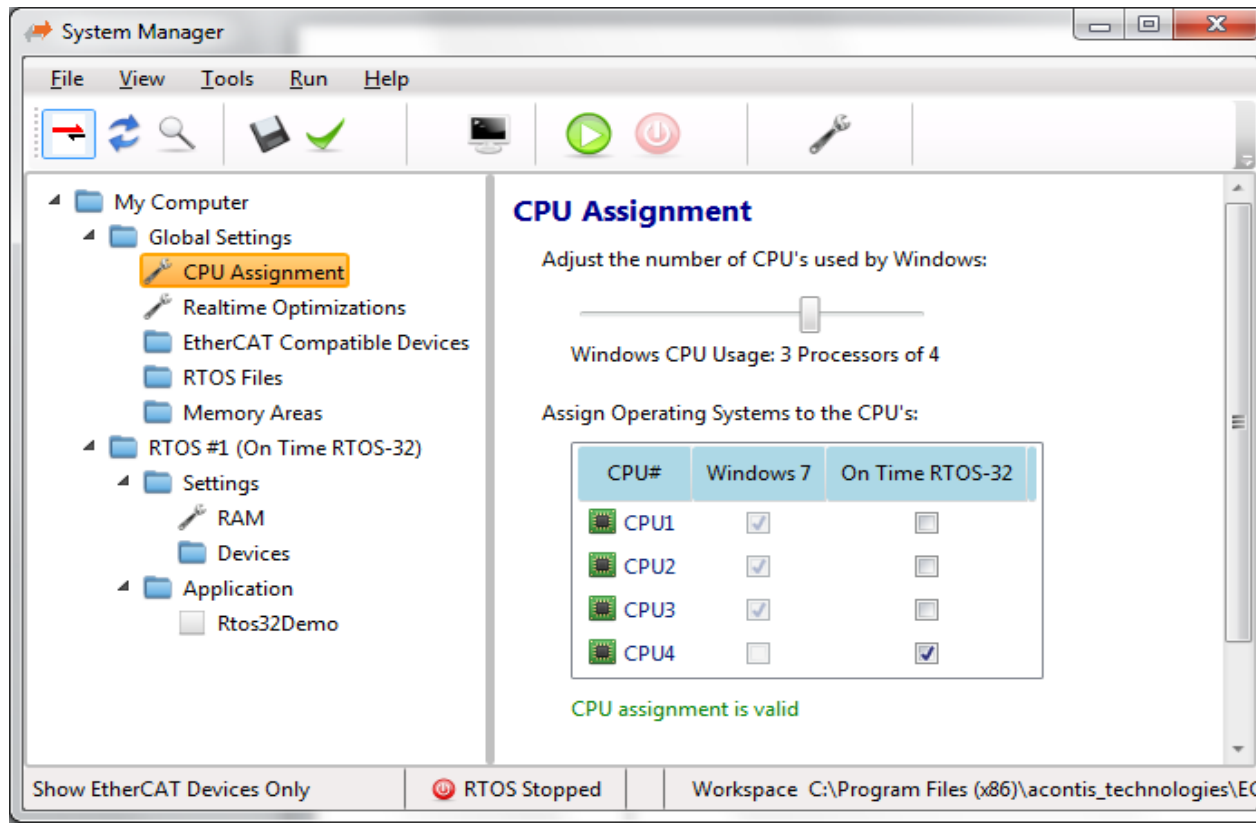
Hardware

Hardware Partitioning

- RTENV needs to be 100% independent from Windows
 - RTENV needs its own part of the PC hardware to be deterministic: CPU, Memory, Devices

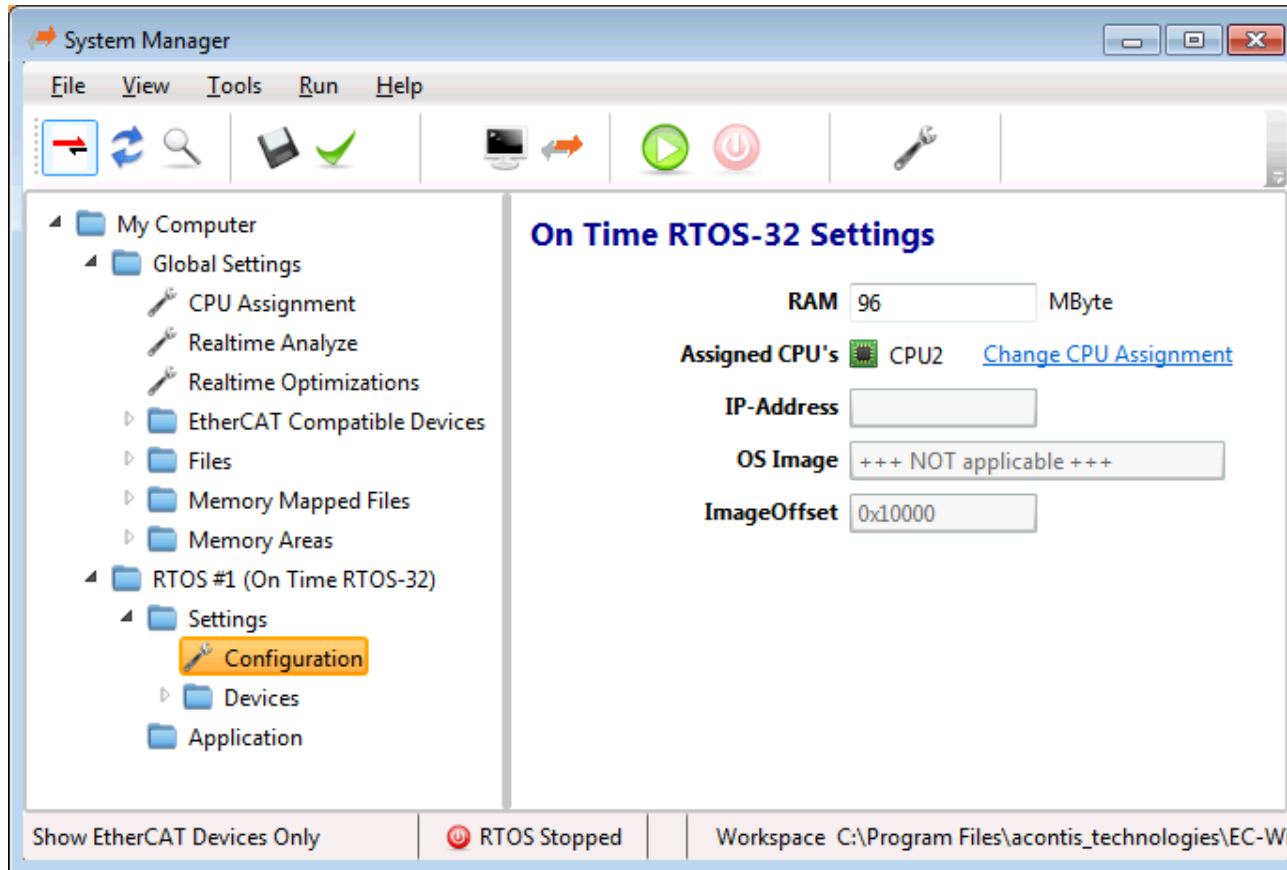


- Select number of CPUs to be used by Windows
- Select CPU(s) to be used by the RTOS

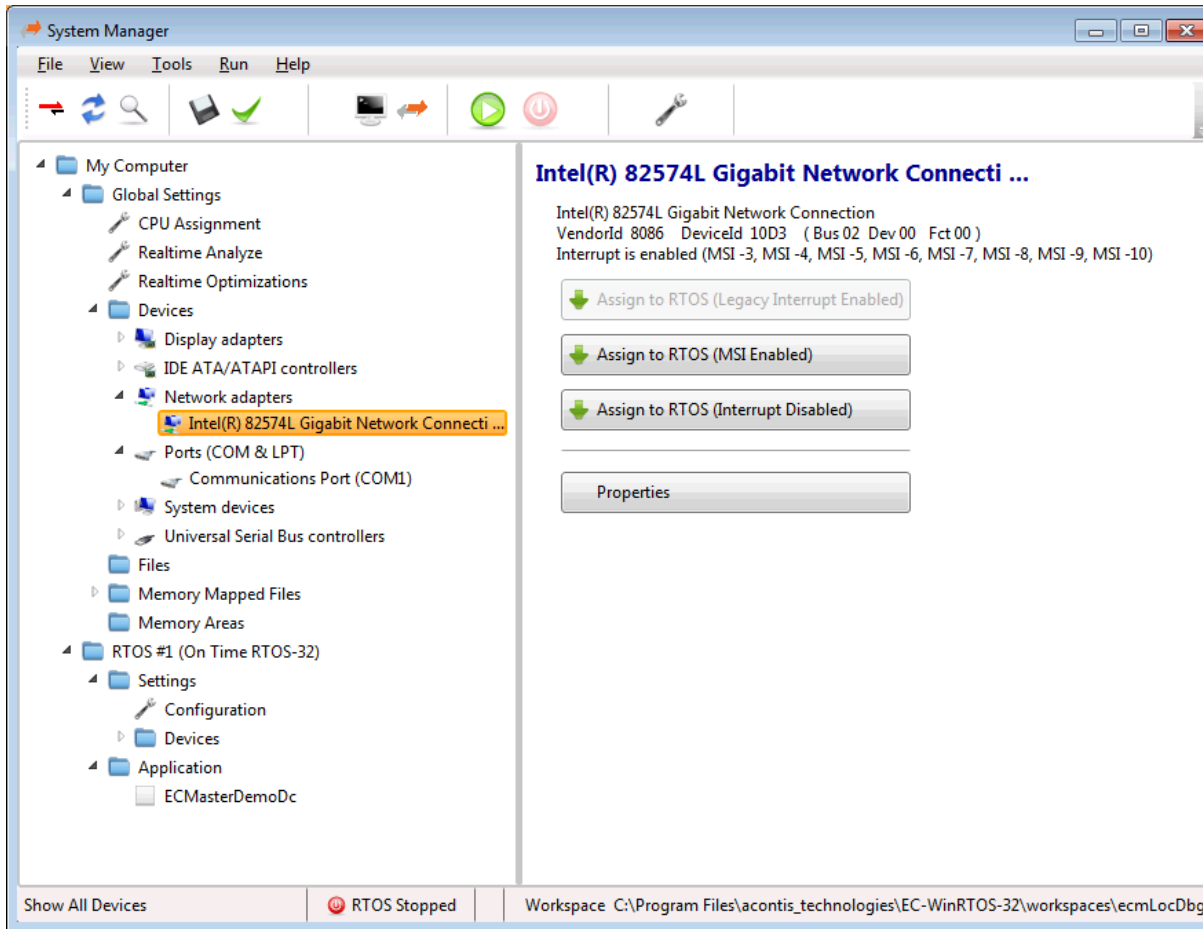


Hardware Partitioning: RAM Assignment

- Set RAM size for the RTOS, will be allocated at early boot stage
- Memory is invisible for Windows (and vice versa)!

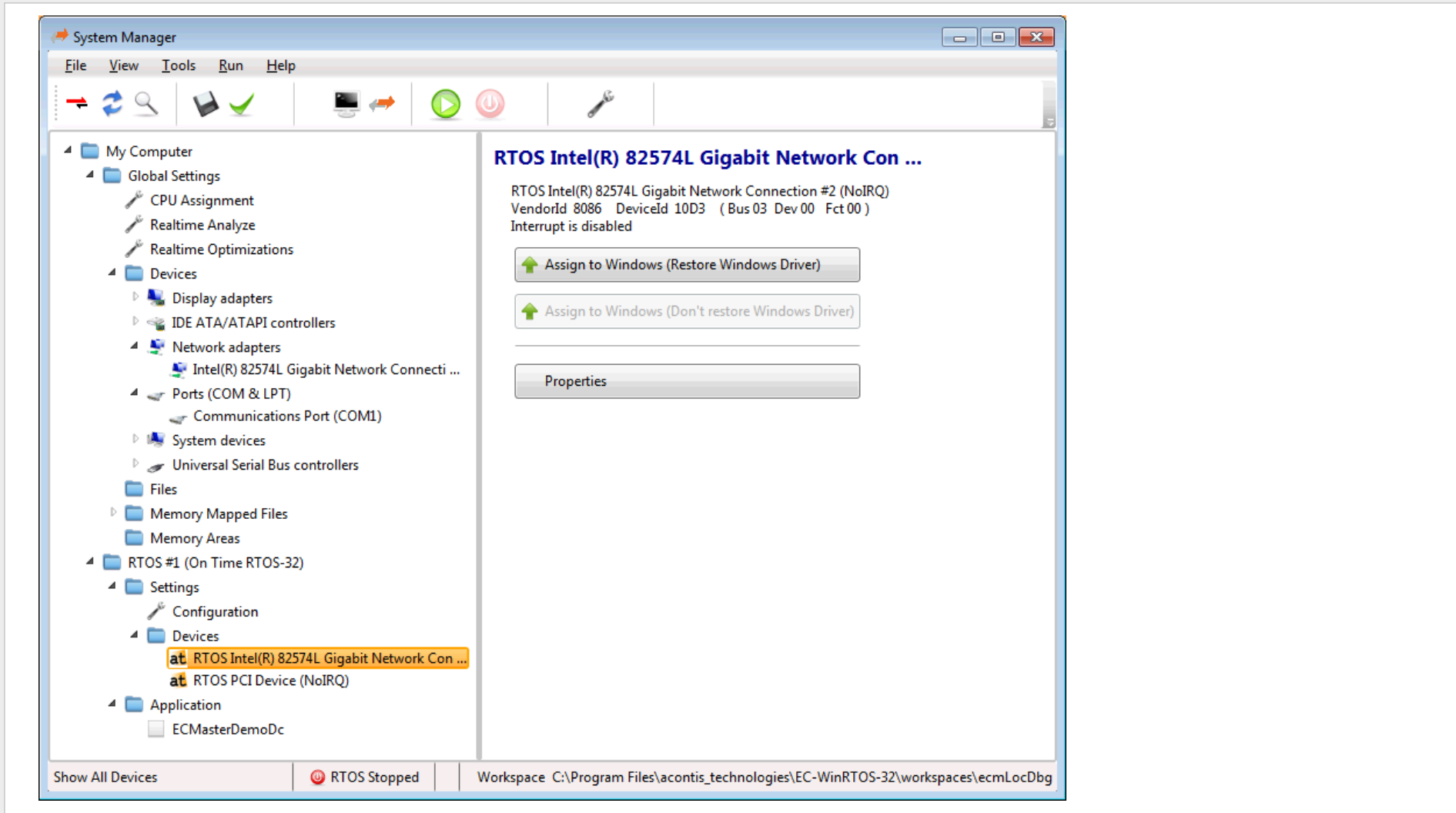


Hardware Partitioning: Device Assignment



- Select Device currently controlled by Windows
- Assign Device to RTOS
- Caveat: PCI interrupt sharing!
- EtherCAT master:
No interrupt required!

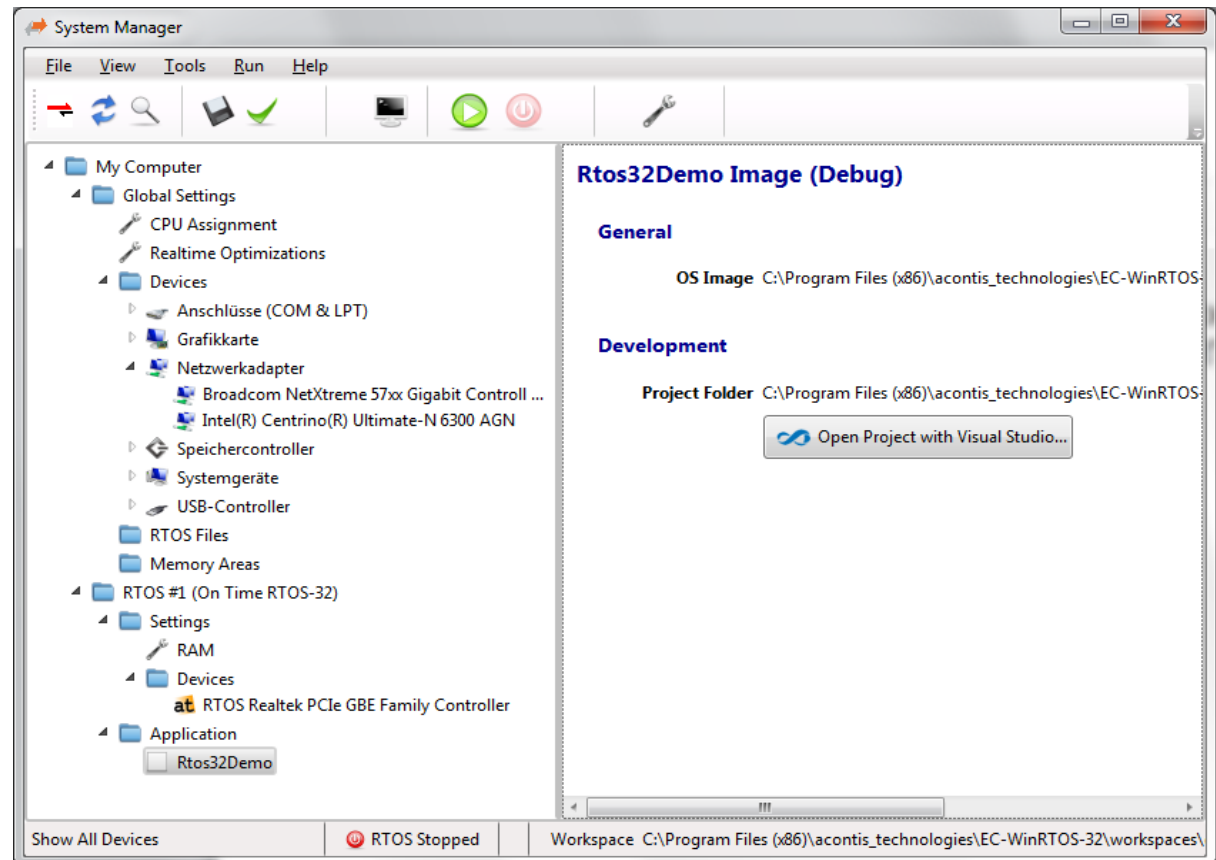
Hardware Partitioning: Device ready to use for the RTOS



EC  ***Win***

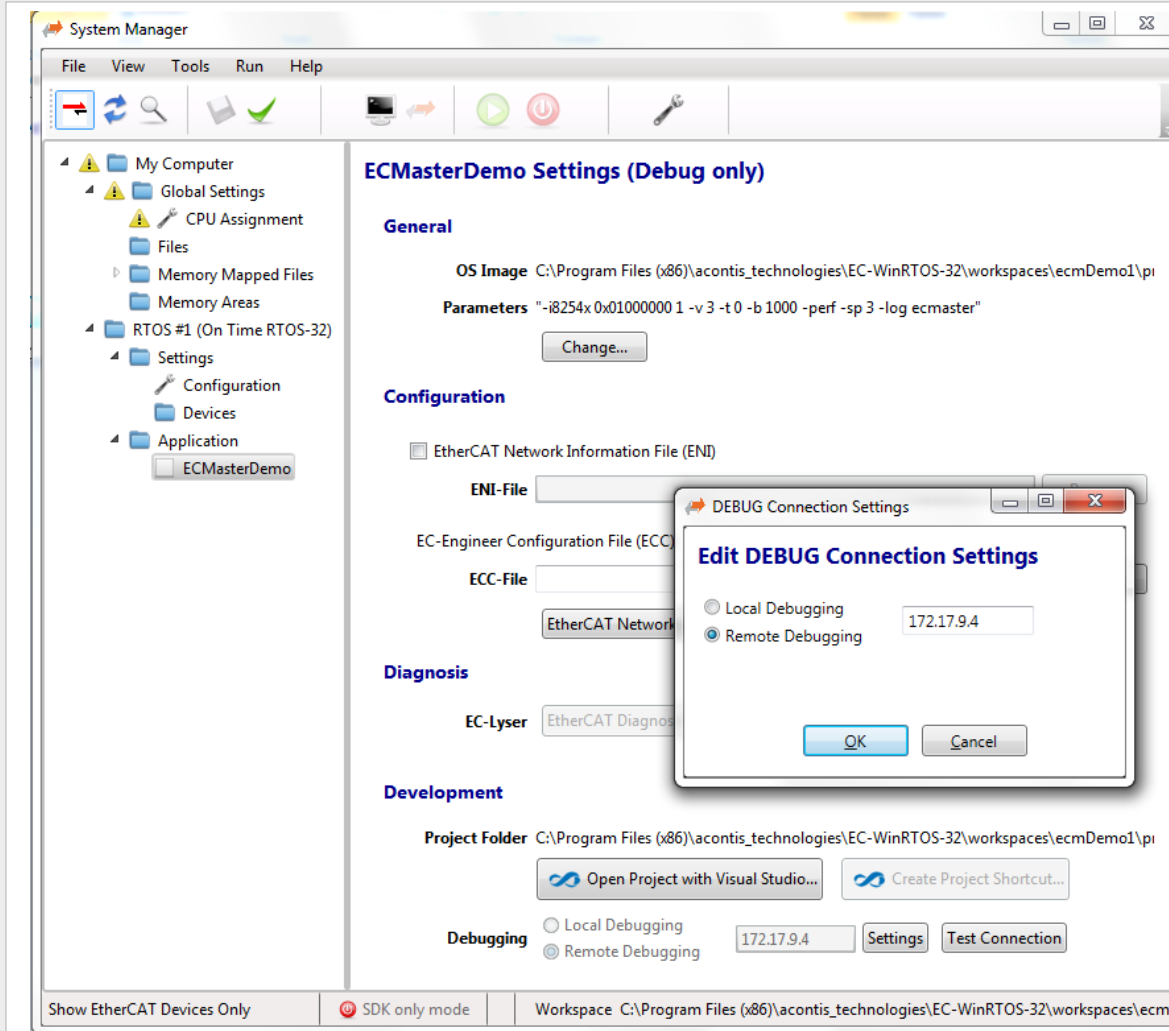
Software Development

- Automatically create new projects based on existing demos
 - Copy files into workspace
 - Create VS solution
 - Add required compiler and linker settings
 - Add all files needed
- Only 2 steps to debug
 - 1.) Build
 - 2.) start debugging



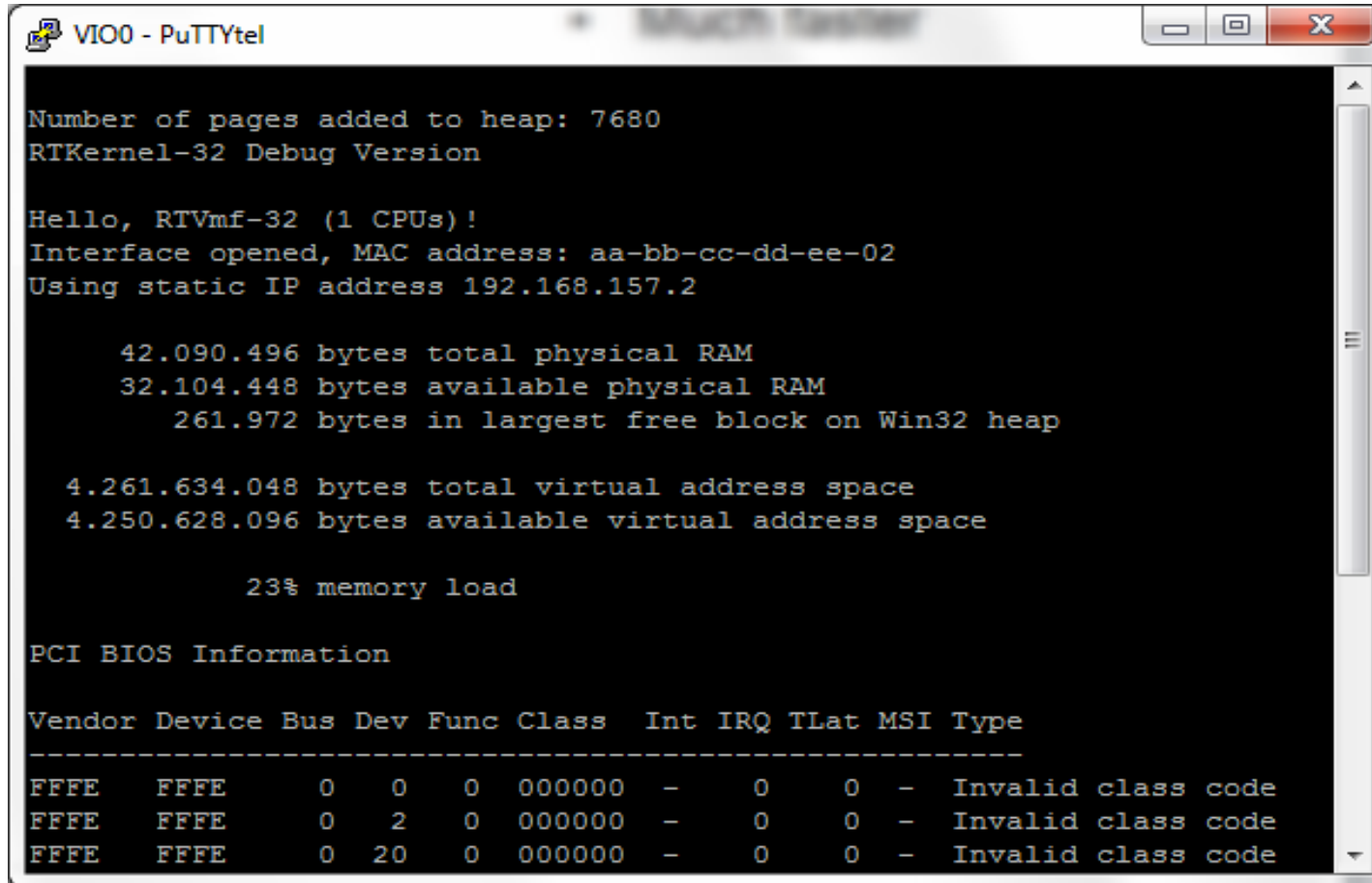
- Support for Visual Studio 2005 up to VS 2015
- Application Wizard
 - Automatically create new real-time application template
 - All required settings, header and library paths set properly
 - Example applications: simply add all files, compile and start debug session
- No difference compared to debugging a Windows application
 - Directly start debugging after linking application
 - Automatically loading application into real-time environment
- Application running in debugger keeps full real-time behavior!

Remote Debugging via TCP/IP



- Target
 - Install runtime components only
 - System Manager launches Debug Monitor
- Host
 - Install SDK components only
 - Select remote IP address.
 - System Manager launches Visual Studio.
 - Applications will be downloaded to remote target via TCP/IP

- Standard Output Device (printf)



```
VIO0 - PuTTYtel

Number of pages added to heap: 7680
RTKernel-32 Debug Version

Hello, RTVmf-32 (1 CPUs)!
Interface opened, MAC address: aa-bb-cc-dd-ee-02
Using static IP address 192.168.157.2

    42.090.496 bytes total physical RAM
    32.104.448 bytes available physical RAM
    261.972 bytes in largest free block on Win32 heap

4.261.634.048 bytes total virtual address space
4.250.628.096 bytes available virtual address space

    23% memory load

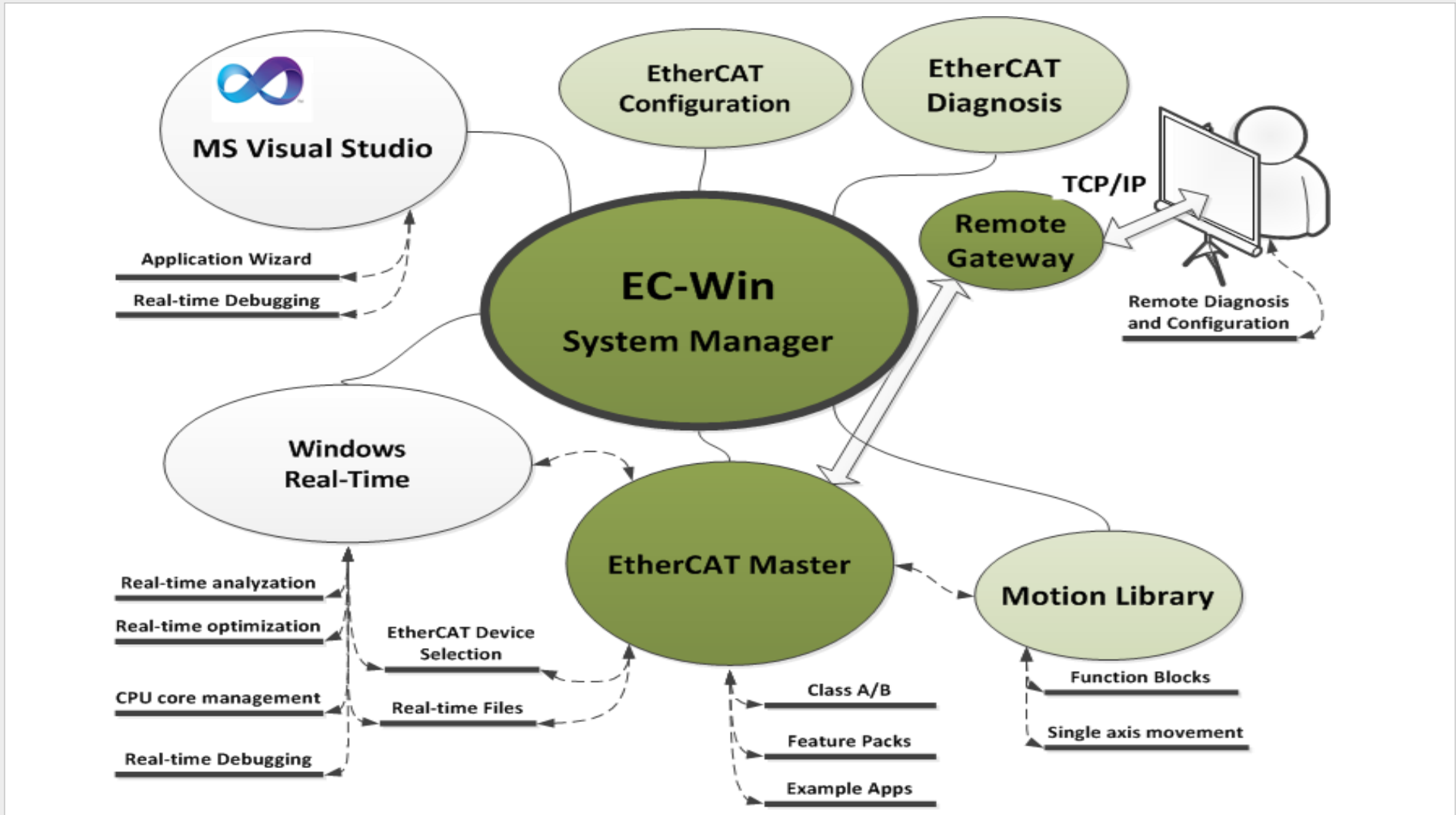
PCI BIOS Information

Vendor Device Bus Dev Func Class  Int  IRQ  TLat  MSI Type
-----
FFFFE  FFFE    0  0  0  000000  -   0    0  -  Invalid class code
FFFFE  FFFE    0  2  0  000000  -   0    0  -  Invalid class code
FFFFE  FFFE    0  20 0  000000  -   0    0  -  Invalid class code
```

EC  ***Win***

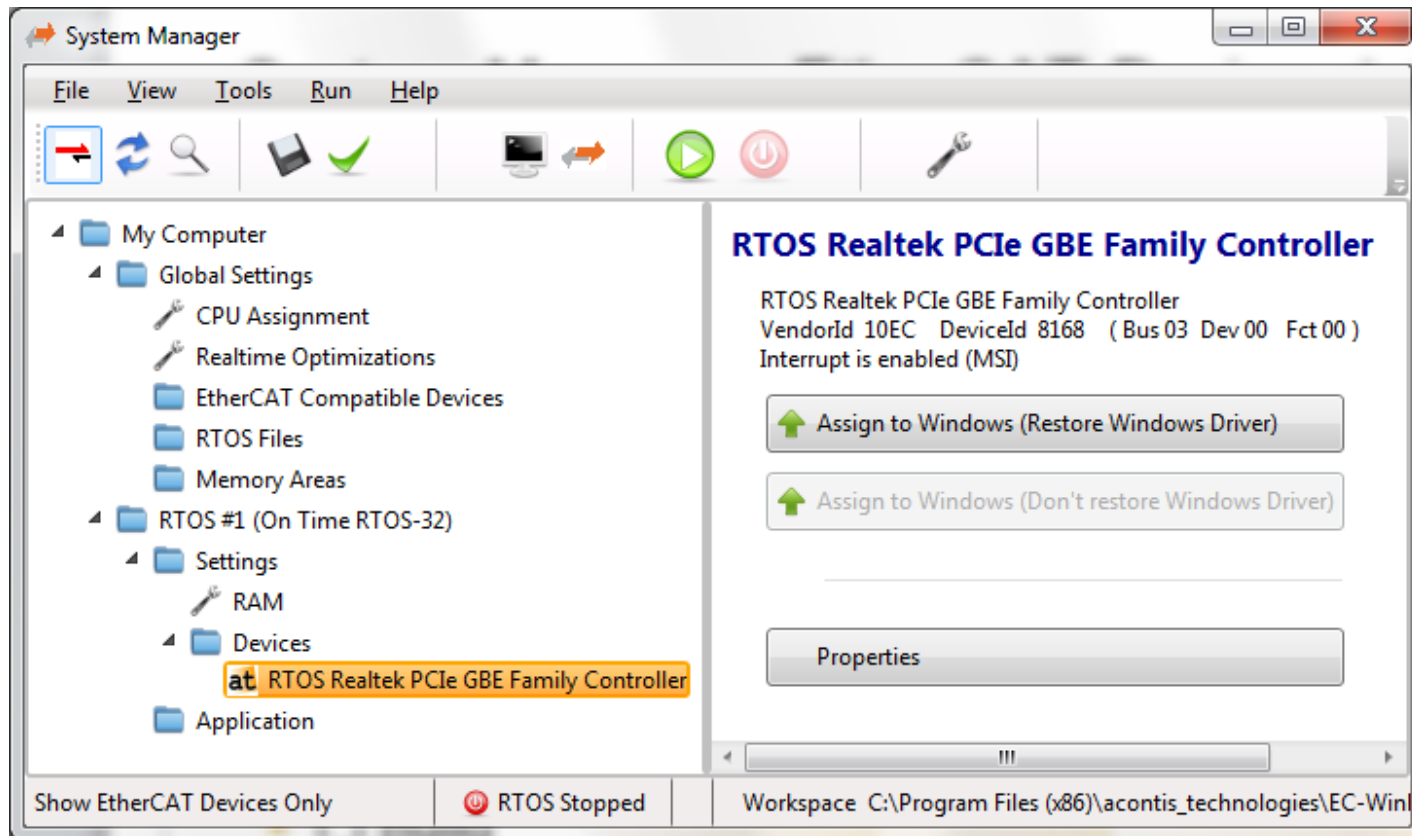
EtherCAT Integration

EtherCAT related components in EC-Win



EtherCAT compatible Ethernet Controller

- Automatically detect all Ethernet Controllers that are supported by EtherCAT master
- Assign to the real-time part

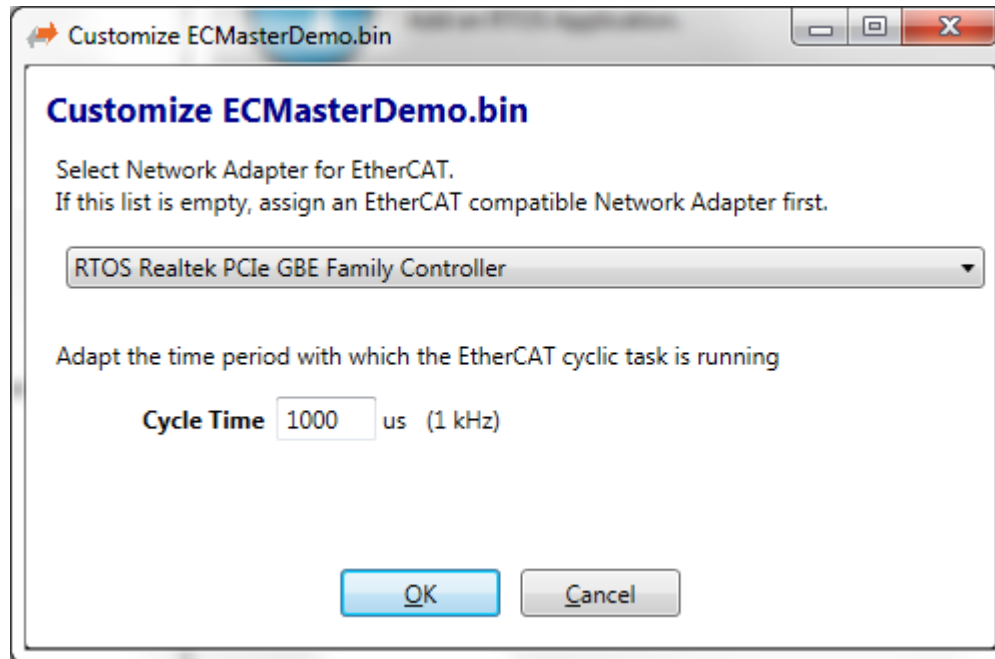


Select EtherCAT example application

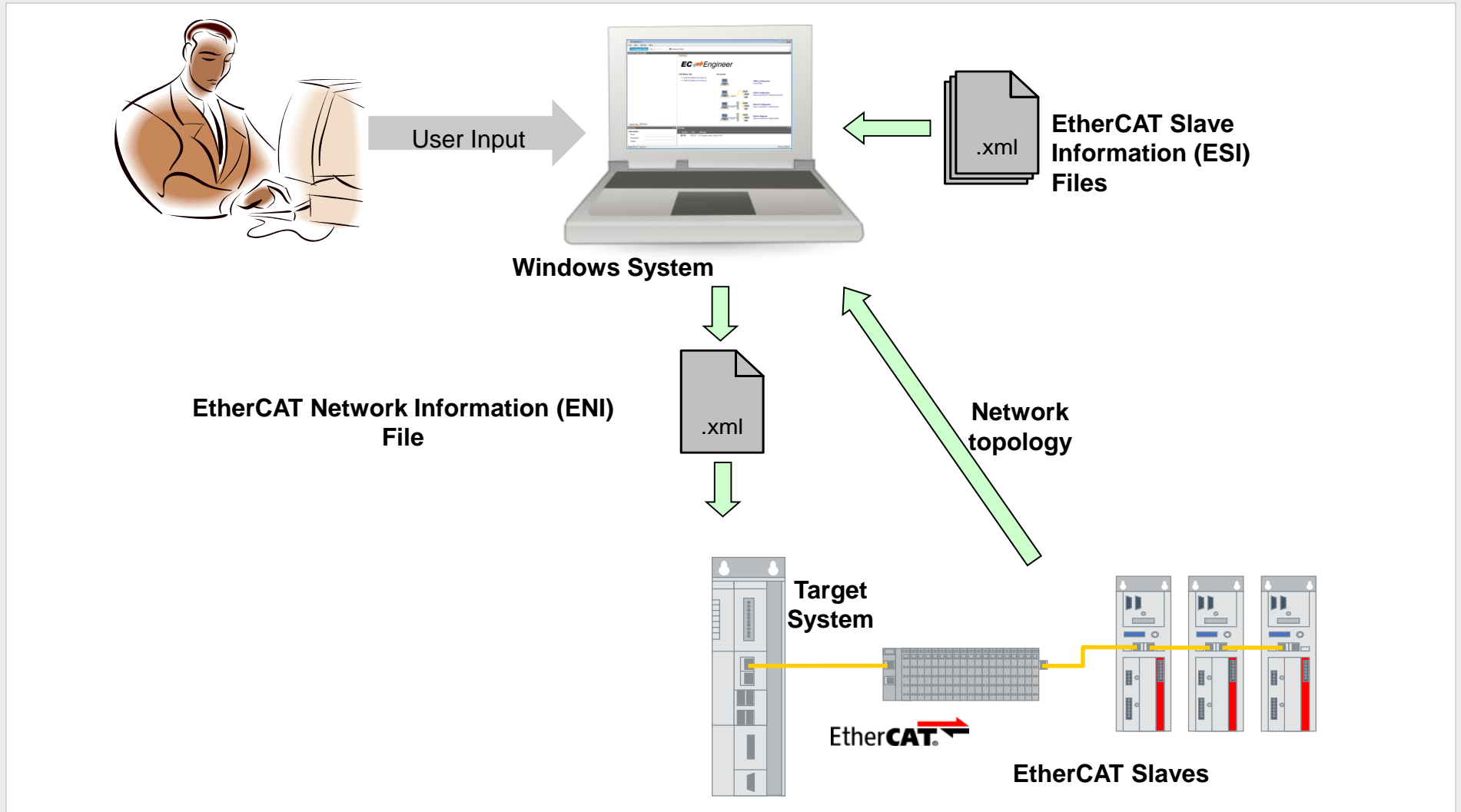
The screenshot displays the 'System Manager' application interface. The main window shows a tree view on the left with folders like 'My Computer', 'Global Settings', 'EtherCAT Compatible Devices', and 'RTOS #1 (On Time RTOS-32)'. The 'Application' folder under 'RTOS #1' is selected. The main area shows an 'Application' section with the text 'Add an RTOS Application.' and two buttons: 'Open Existing Application' and 'Create New Application Project (Debug Only)'. A dialog box titled 'Add New Realtime Application' is open, showing a list of applications to select. The list includes 'ECMasterDemo.bin', 'EC-Motion Demo', 'ECMasterDemoSyncSm.bin', 'ECMasterDemoDc.bin', 'RTOS-32Demo.bin', and 'RealtimeDemo.bin'. The 'ECMasterDemo.bin' item is highlighted. A 'Cancel' button is visible next to the list. The status bar at the bottom shows 'Show EtherCAT Devices Only', 'RTOS Stopped', and the workspace path: 'C:\Program Files (x86)\acontis_technologies\EC-WinRTOS-32\workspaces\defa'.

Example application parameters

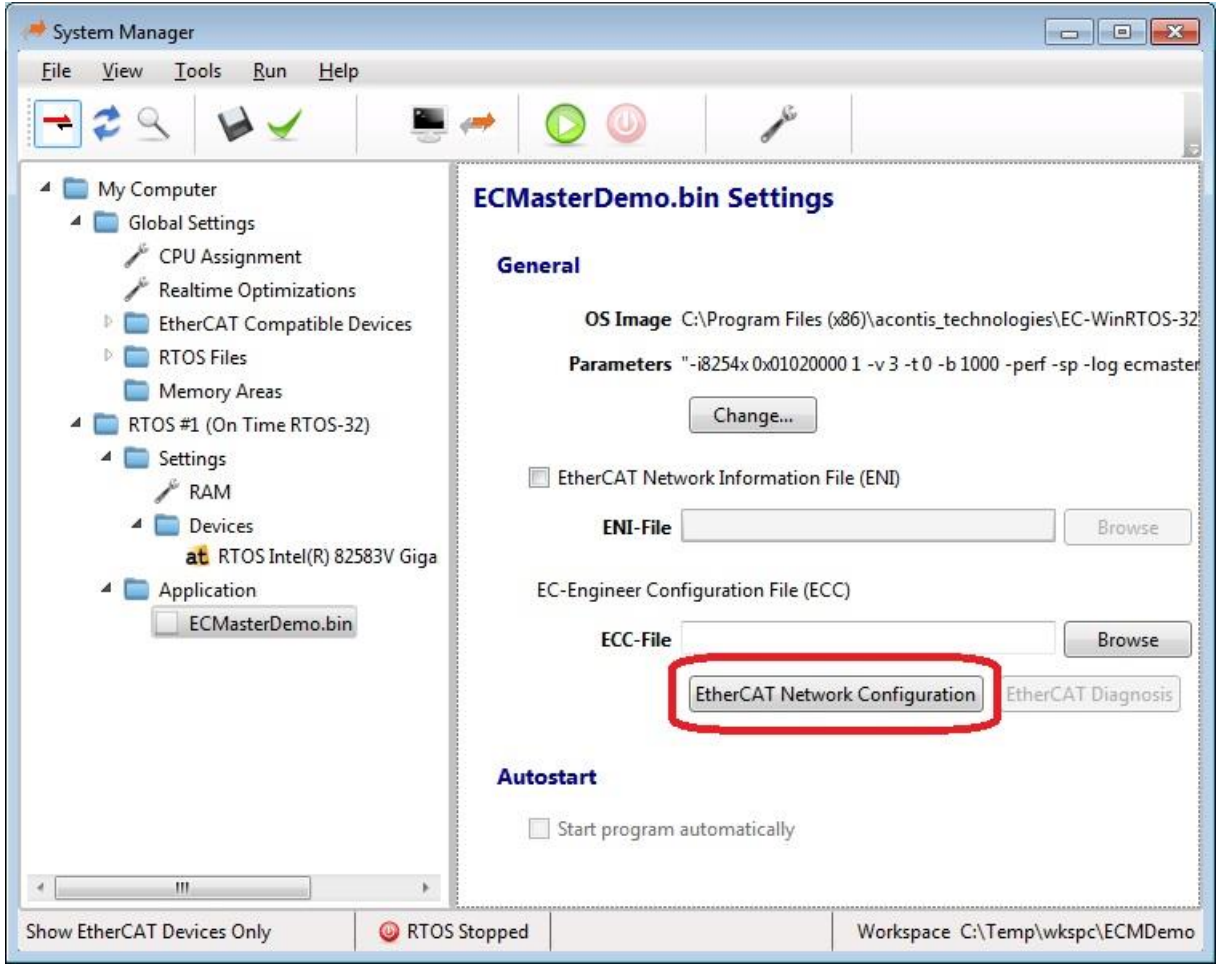
- Select network adapter for EtherCAT
- Adjust cycle time



EtherCAT Network Configuration Data Flow



EtherCAT Network Configuration (1)



- ENI file required to become operational
- EC-Engineer integrated into System Manager

EtherCAT Network Configuration (2)

EC-Engineer [C:\Temp\wkspc\ECMDemo\config\ECMasterDemo_0.ecc]

Project Explorer

- EC-Win Class-A Master
 - Slave_1001 [EK1100] (1001)
 - Slave_1002 [EL2004] (1002)
 - Slave_1003 [EL2004] (1003)
 - Slave_1004 [EL1014] (1004)
 - Slave_1005 [EL1014] (1005)
 - Slave_1006 [EL4132] (1006)
 - Slave_1007 [EK1110] (1007)

Device Editor

Master | Process Data Image | Advanced Options | Distributed Clocks

General

Unit Name: EC-Win Class-A Master

Cycle Time [us]: 1000

Slaves connected to remote system

IP Address: 192.168.157.2

Port: 6000

Master-Instance: 0

Short Info

Information

Name: EC-Win Class-A Ma

Description: EtherCAT Master U

Vendor: acontis technologie

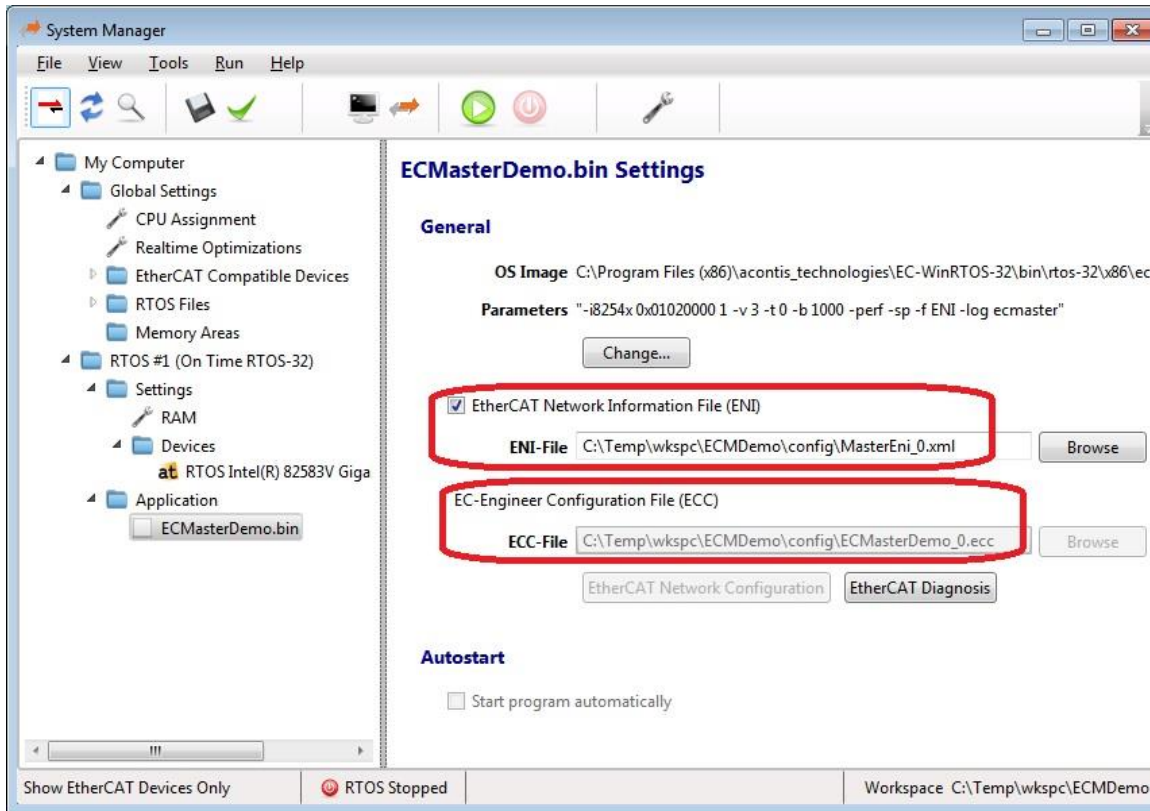
Networks: 1 | Slaves: 7

V100 - PuTTYtel

```
PerfMsmt 'JOB_MasterTimer' (avg/max) [usec]: 0.8/ 27.4
PerfMsmt 'JOB_SendAcycFrames' (avg/max) [usec]: 1.7/ 9.3
PerfMsmt 'Cycle Time' (avg/max) [usec]: 999.9/1020.7
PerfMsmt 'myAppWorkPd' (avg/max) [usec]: 0.0/ 2.7
-----
PerfMsmt 'JOB_ProcessAllRxFrames' (avg/max) [usec]: 0.6/ 22.4
PerfMsmt 'JOB_SendAllCycFrames' (avg/max) [usec]: 7.9/ 17.9
PerfMsmt 'JOB_MasterTimer' (avg/max) [usec]: 0.8/ 27.4
PerfMsmt 'JOB_SendAcycFrames' (avg/max) [usec]: 1.7/ 9.3
PerfMsmt 'Cycle Time' (avg/max) [usec]: 999.9/1020.7
PerfMsmt 'myAppWorkPd' (avg/max) [usec]: 0.0/ 2.7
-----
PerfMsmt 'JOB_ProcessAllRxFrames' (avg/max) [usec]: 0.6/ 22.4
PerfMsmt 'JOB_SendAllCycFrames' (avg/max) [usec]: 7.9/ 17.9
PerfMsmt 'JOB_MasterTimer' (avg/max) [usec]: 0.8/ 27.4
PerfMsmt 'JOB_SendAcycFrames' (avg/max) [usec]: 1.7/ 9.3
PerfMsmt 'Cycle Time' (avg/max) [usec]: 999.9/1020.7
PerfMsmt 'myAppWorkPd' (avg/max) [usec]: 0.0/ 2.7
```

- Master runs in server mode
- EC-Engineer automatically connects to master stack
- Scan EtherCAT network
- Optionally adjust settings
- On close: store results

EtherCAT Network Configuration (3)



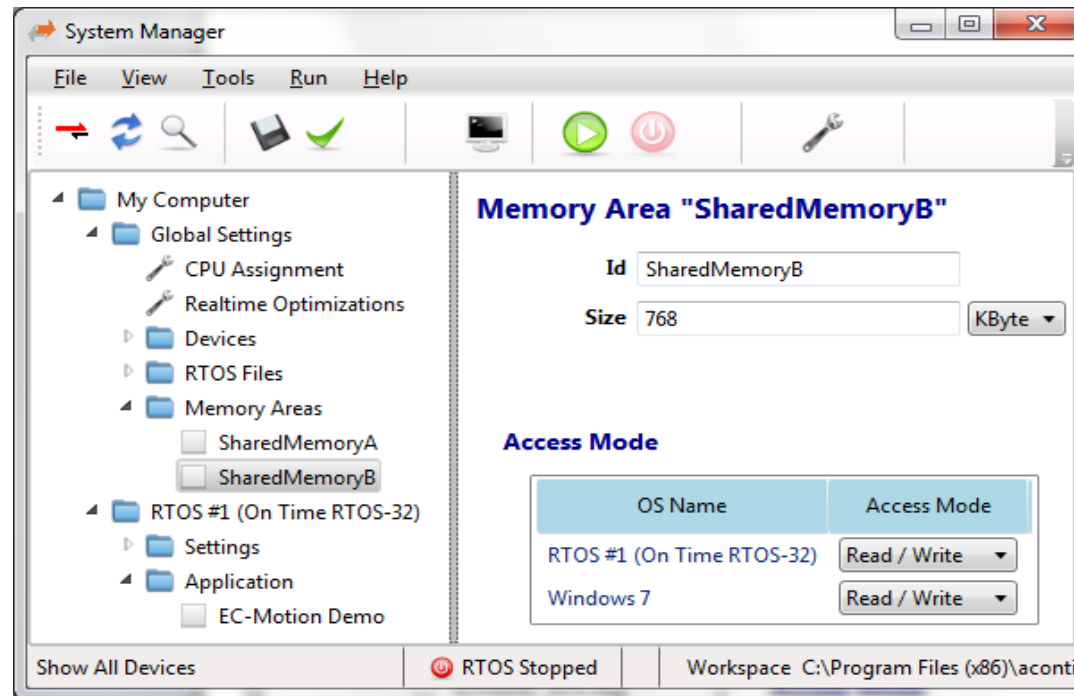
- ECC, ENI stored
- Application now ready to operate
- Start EC-Engineer or EC-Lyser for diagnosis purpose



Communication

Data and information exchange between Windows and the RTOS

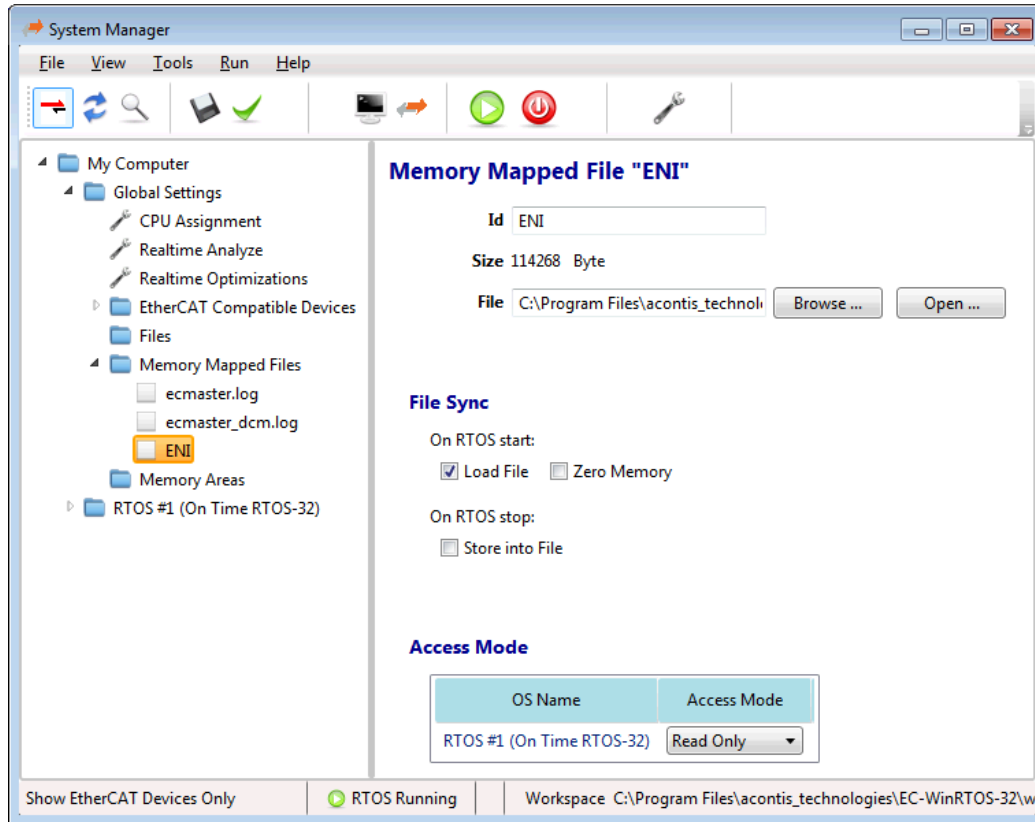
Low level communication: Shared Memory Areas + Events



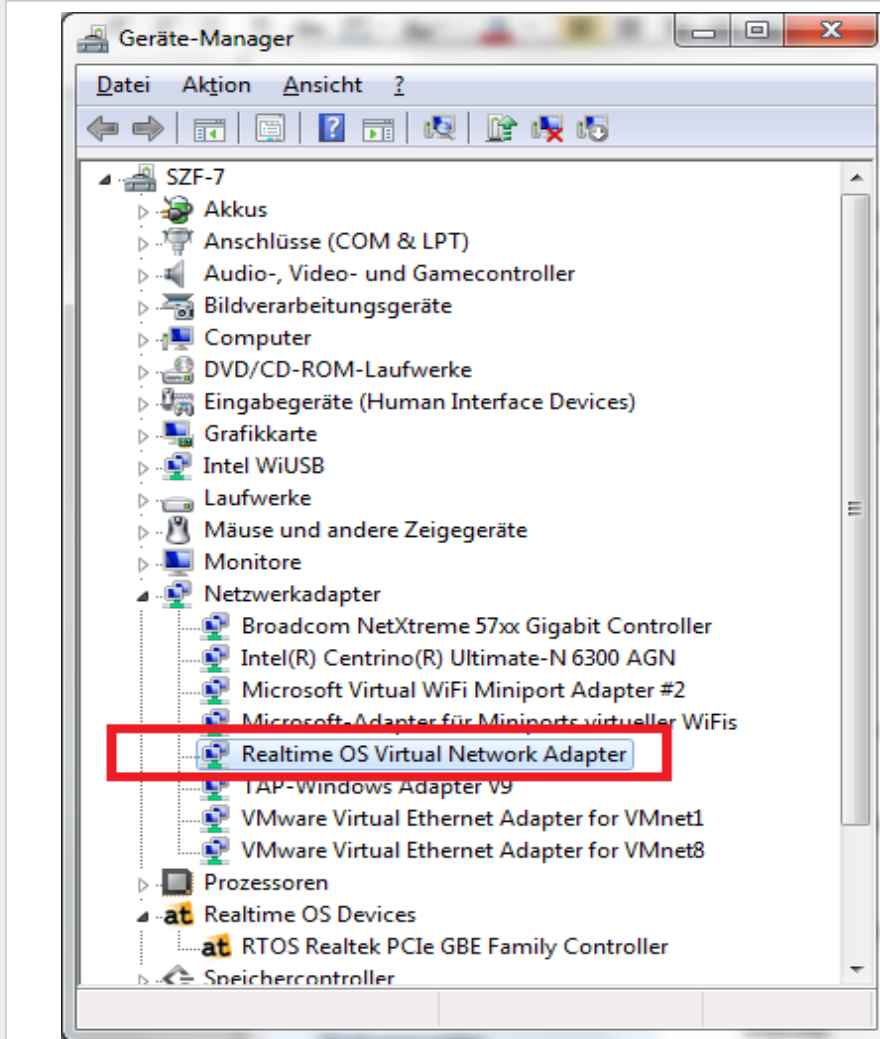
- Direct access via Pointers to memory
- Configurable via System Manager tool
- Events: notification if new data available

- Named message queues and pipes
 - Telegram or stream oriented communication
- Real-time Socket API
 - Create applications which can run locally using Shared Memory or remote using TCP/IP
- Remote TCP Gateway
 - Routing TCP to Shared Memory
 - Connect external tools with Windows IP address
 - automatic forwarding to master stack
- Arbitrary (hard disk) File Access
 - Access (read/write/create/delete) any files stored on the hard disk

Data Exchange: File Access via Memory



- Files stored in memory instead of hard disk
- Much faster access
- Independent from Windows
- Useful for real-time configuration and log files
- Can be accessed via ANSI functions (fopen, fread, ...)



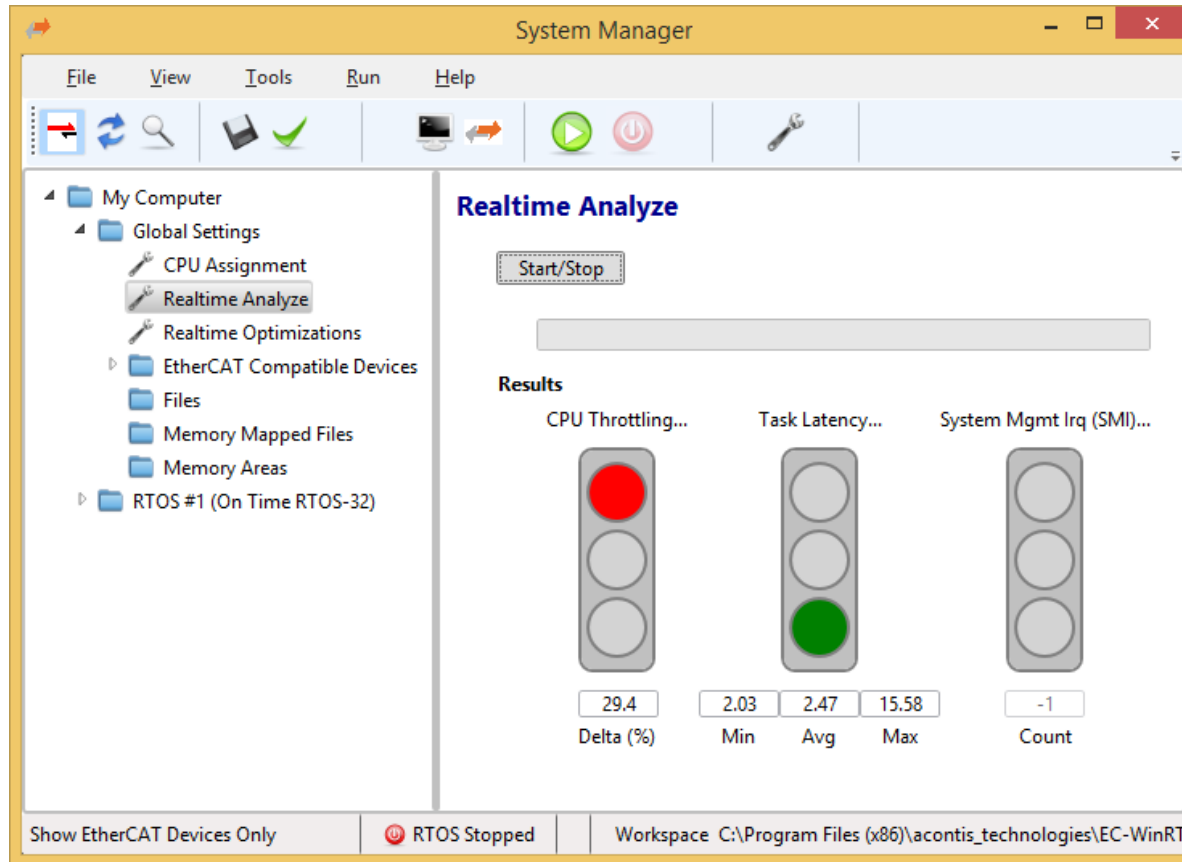
- Used for application debugging
 - Visual Studio Connection to the real-time Debug Monitor
- Optional Package: TCP/IP stack
 - Useful for scalable solutions
 - e.g. same application to communicate with external Windows tools

EC  ***Win***

Real-time on Windows

Real-time analysis and optimization

System Manager: Real-time analysis (no optimization)

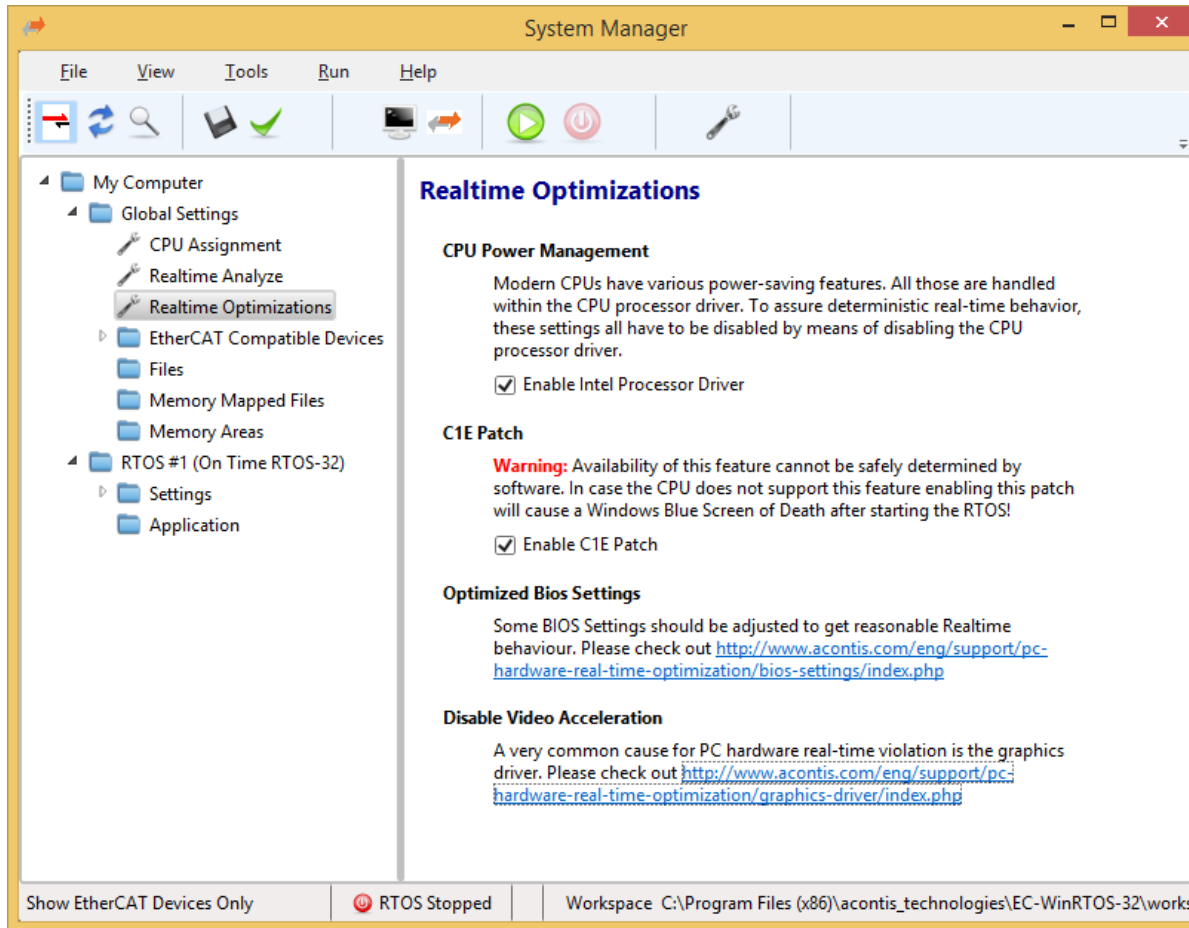


• Results

- CPU clock: throttling active?
- Timer: task level latency
- System Management Interrupts?

On non-optimized PCs real-time often cannot be guaranteed!

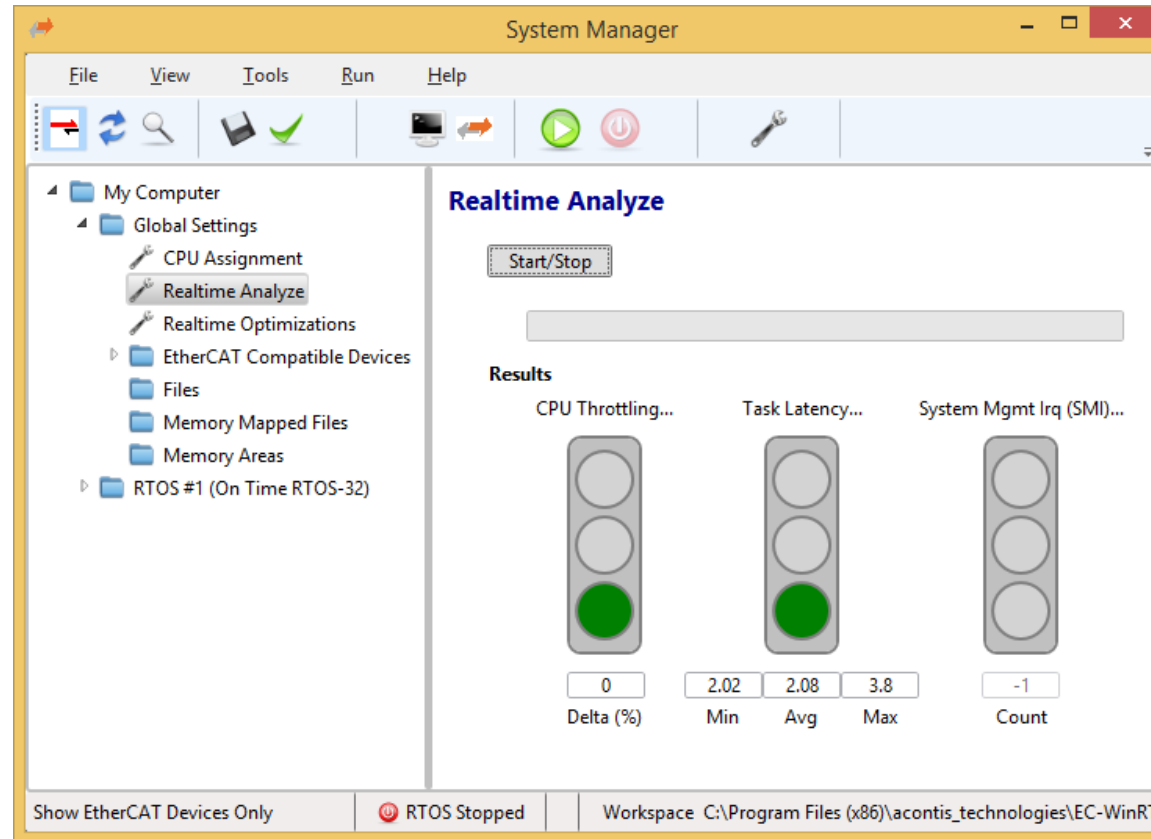
System Manager: PC real-time optimization



• Typical optimization steps

- Disable CPU Power Management
- Apply C1E Patch
- BIOS settings (see hints on acontis website)
- VGA Driver issue?

System Manager: Real-time analyzation (with optimization)



- Results
 - CPU clock: no throttling
 - Timer: short task level latency
 - No SMIs

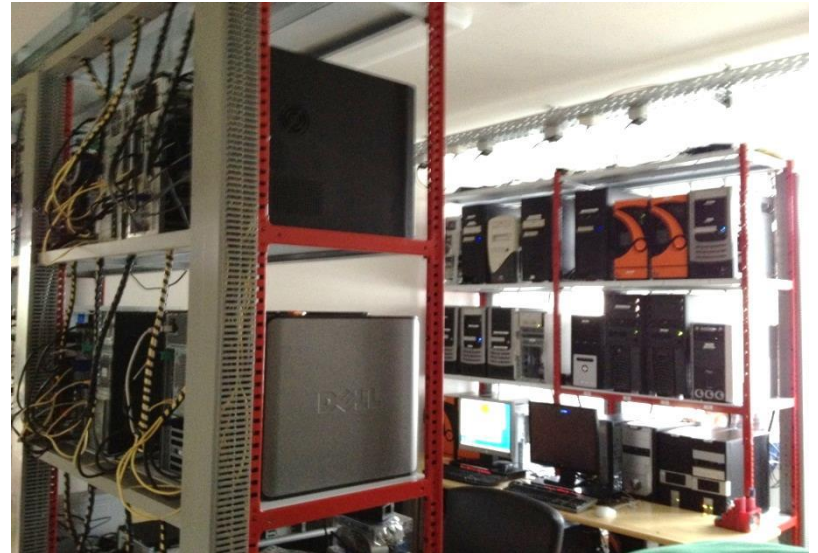
After applying one or multiple optimizations
→ most PCs will be able to run real-time applications!

- Fully integrated EtherCAT real-time solution
 - one vendor, one support contact
 - Acontis has expertise for Windows real-time extensions back to 1994 and is leading provider for EtherCAT software since 2005
 - System Manager tool: One-stop tool for the whole product
- Key Features
 - Win32 real-time platform with Microsoft® Visual Studio® support for the non-real-time and real-time part of the software
 - High speed: No interrupt latency impact
 - Class A EtherCAT Master Stack
 - High Performance real-time Ethernet Drivers
- All runtime components included: No additional license required
 - License for EtherCAT Class A Master Stack
 - License for Basic Windows Real-time Platform
 - License for On Time RTOS-32 real-time operating system

EC  ***Win***

Quality Assurance

- Test lab to test products before a new version gets released
- More than 50 different PC's
 - some supplied by customers as reference systems
- A wide range of Intel and AMD processors
 - AMD: Athlon, Duron, Fusion etc.
 - Intel: Celeron/Pentium, Core i5/7, XEON, Atom etc.
- Different chipsets
 - Intel, Nvidia, VIA, SiS etc.
- Automated test scripts (e.g. ½ million start/stop cycles in one single test)



KUKA Robot Controller: TUV certified Safety inside!



- TUV certified safety PLC
- Based on latest acontis real-time technology
- EtherCAT master from acontis as well
- Robot controller and Safety PLC within one single controller PC
- 15.000 controllers per year
- ➔ Quality proven technology!