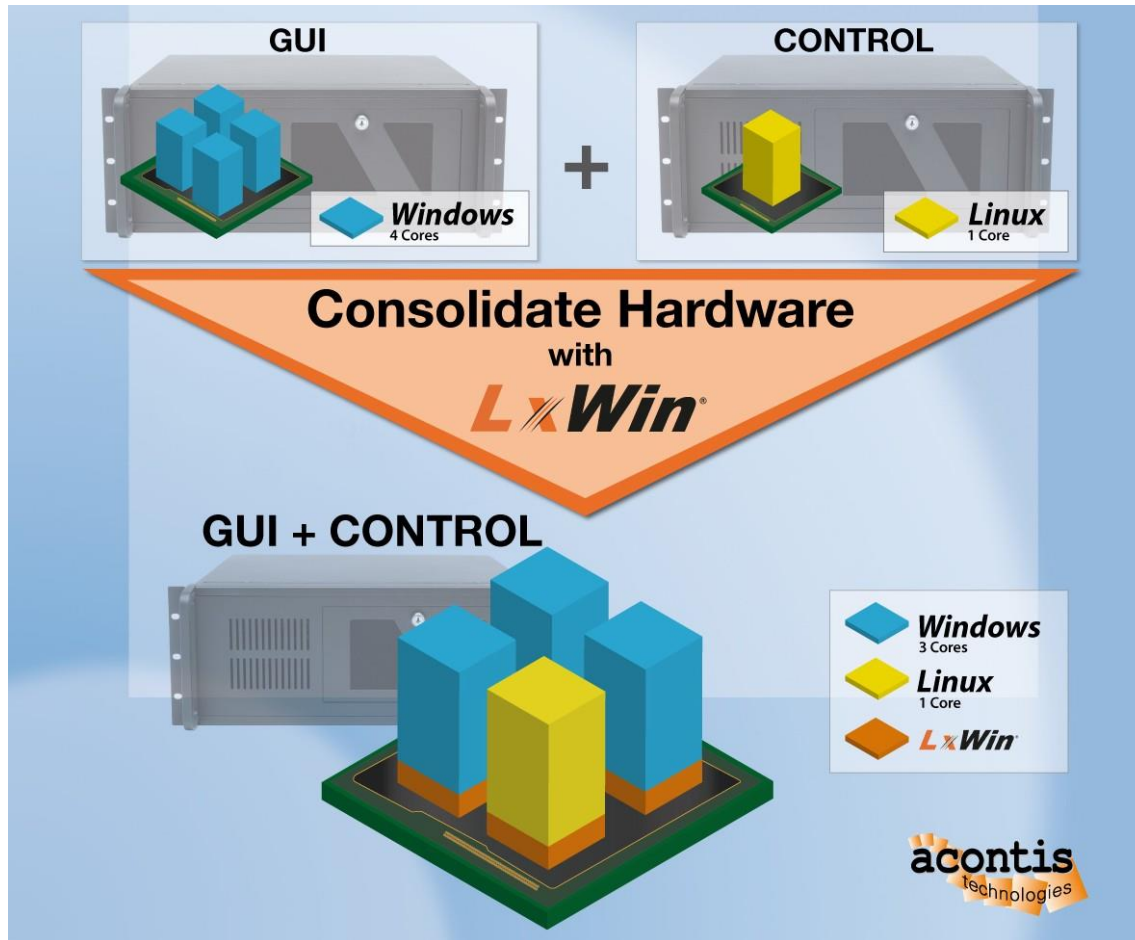


LxWin[®]

Windows Real-time Hypervisor

August 2018

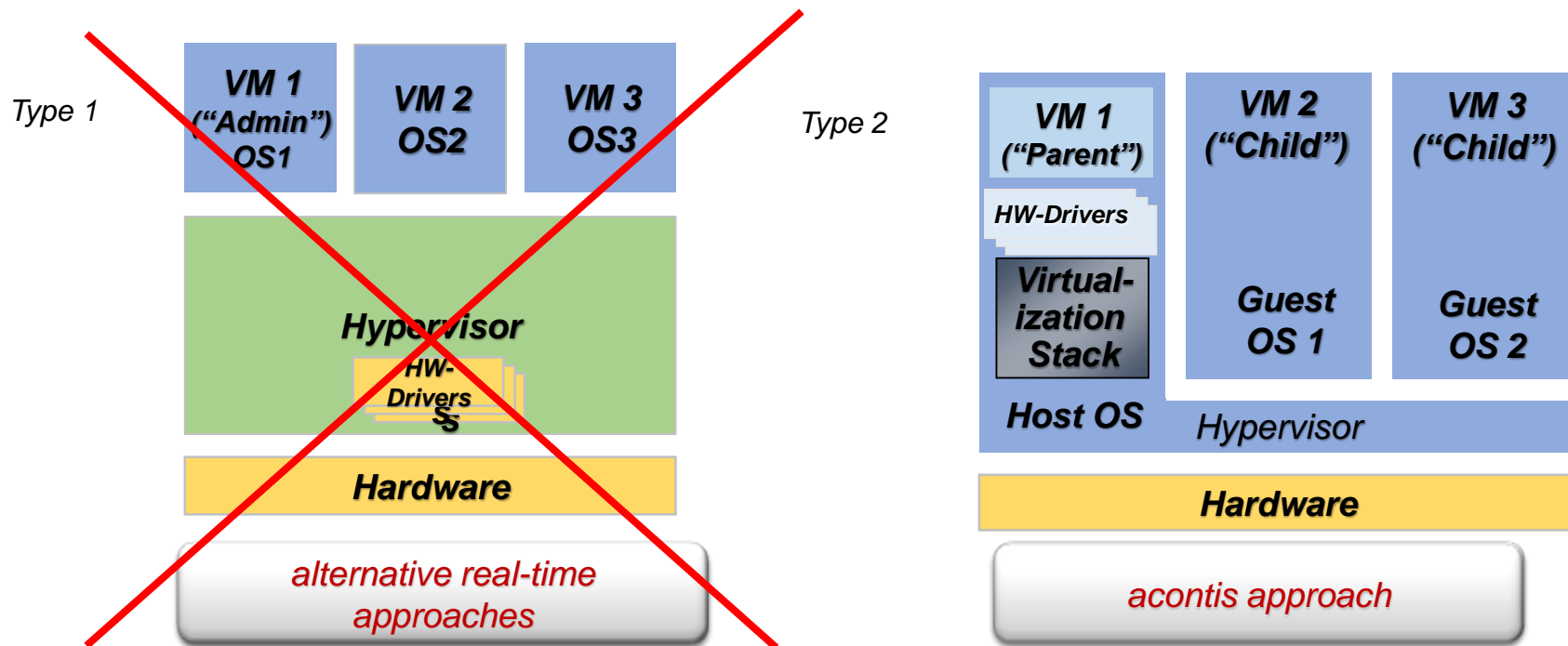
Run Real-time Linux alongside Windows on one PC

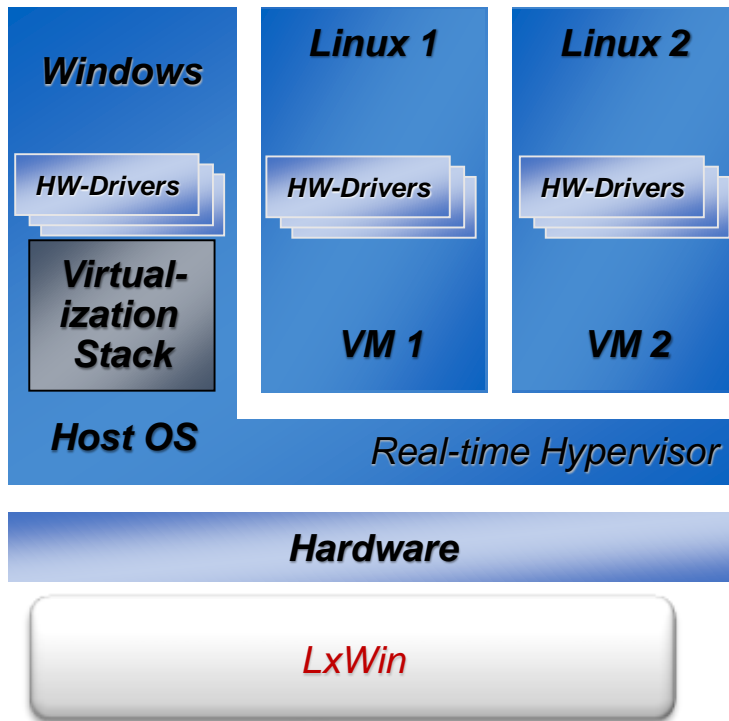


- Save hardware costs
- Reduce space:
Compact controllers can be built
- Improve reliability:
Increase the MTBF as less hardware components are used
- Scalability:
Use the same Linux application in an embedded system

General Hypervisor Overview (from Wikipedia and Microsoft)

- A hypervisor is a piece of a platform-virtualization software that allows multiple operating systems to run on a host computer concurrently
- Type 1 (or native, bare-metal) hypervisors are software systems that run directly on the host's hardware to control the hardware and to monitor guest operating-systems
- Type 2 (or hosted) hypervisors are software applications running within a conventional operating-system environment (Host OS)





- Hypervisor Host OS = Windows
- Hypervisor Host OS and Guest OS are the same
- Windows boots first and loads the real-time Hypervisor
- Real-time Hypervisor boots and controls Linux
- Windows installed first, Hypervisor installed later
 - use unmodified original Windows OS
- Runs on all hardware where Windows runs
 - Intel or AMD processors
 - single or multi core processors
 - No HW VT support needed in the processor
 - HW VT support optional (e.g. for higher reliability, Shared Core Support)

LxWin[®]

Real-Time Linux

The Linux real-time operating system

- The most popular real-time OS
 - Used by customers all over the world
 - Most powerful real-time OS available (API richness, OS capabilities, ...)
 - Thousands of drivers available, hundreds of third party products available
- RT Preempt Patch
 - Adds hard real-time to Linux, proven in thousands of industrial applications
 - LxWin: ready to use headless real-time image included
 - Yocto support to create custom real-time Linux kernels
- Scalable Solutions can be built
 - Level 1: Embedded Controller without GUI: use native, embedded Linux
 - Level 2: Embedded Controller with GUI: use native Linux with GUI (e.g.Qt)
 - Level 3: High End Controller with powerful Windows GUI: use LxWin

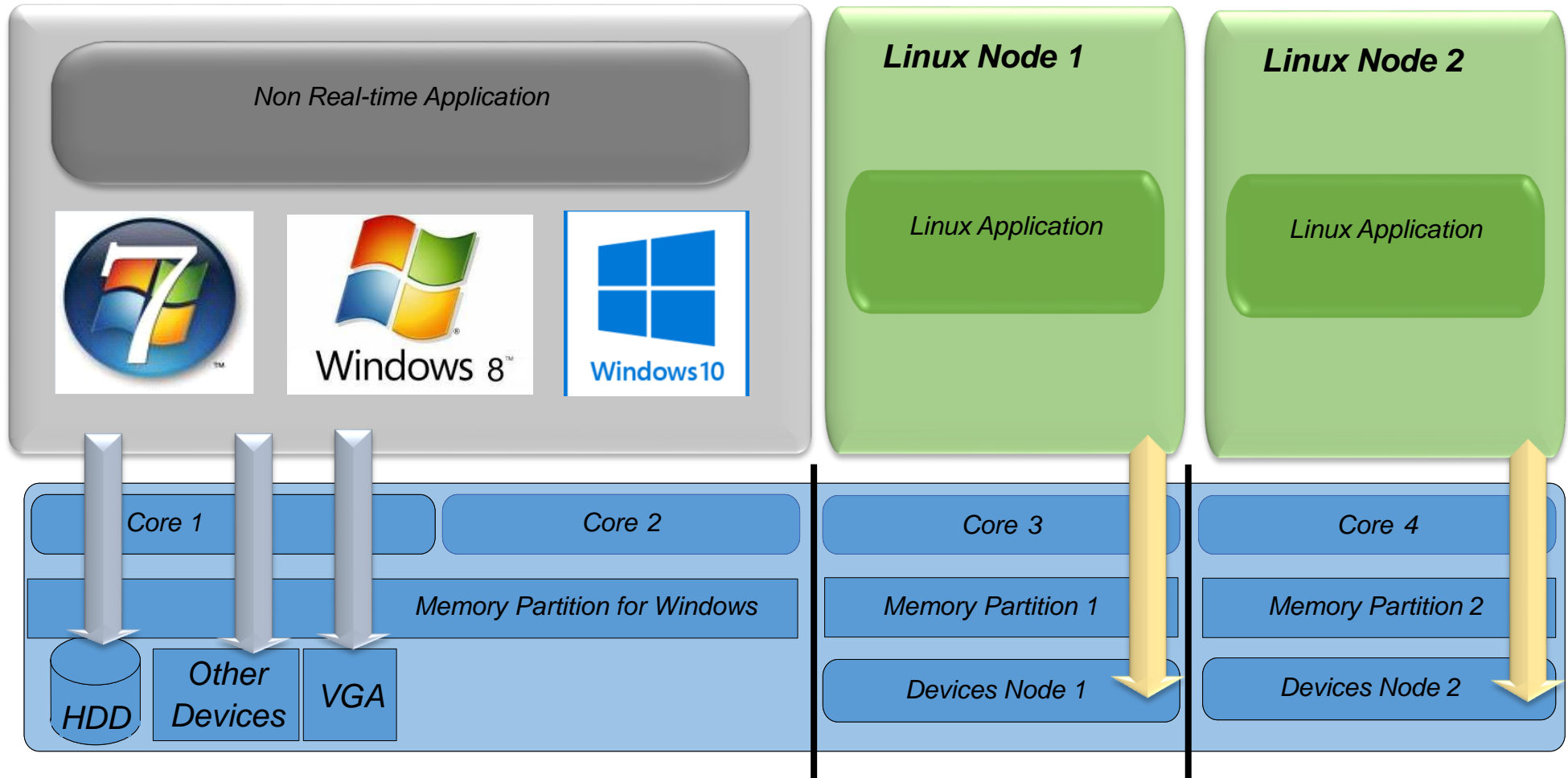
- Hard real-time performance (extremely short latencies).
- 64 Bit and 32 Bit support
- Symmetric multiprocessing (utilize multiple cores)
- Powerful OS
 - Separate applications from Kernel
 - Powerful communication means
 - Posix compliant
 - Great development tools

LxWin[®]

Real-time Environment
Hardware Partitioning

Hardware Partitioning

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



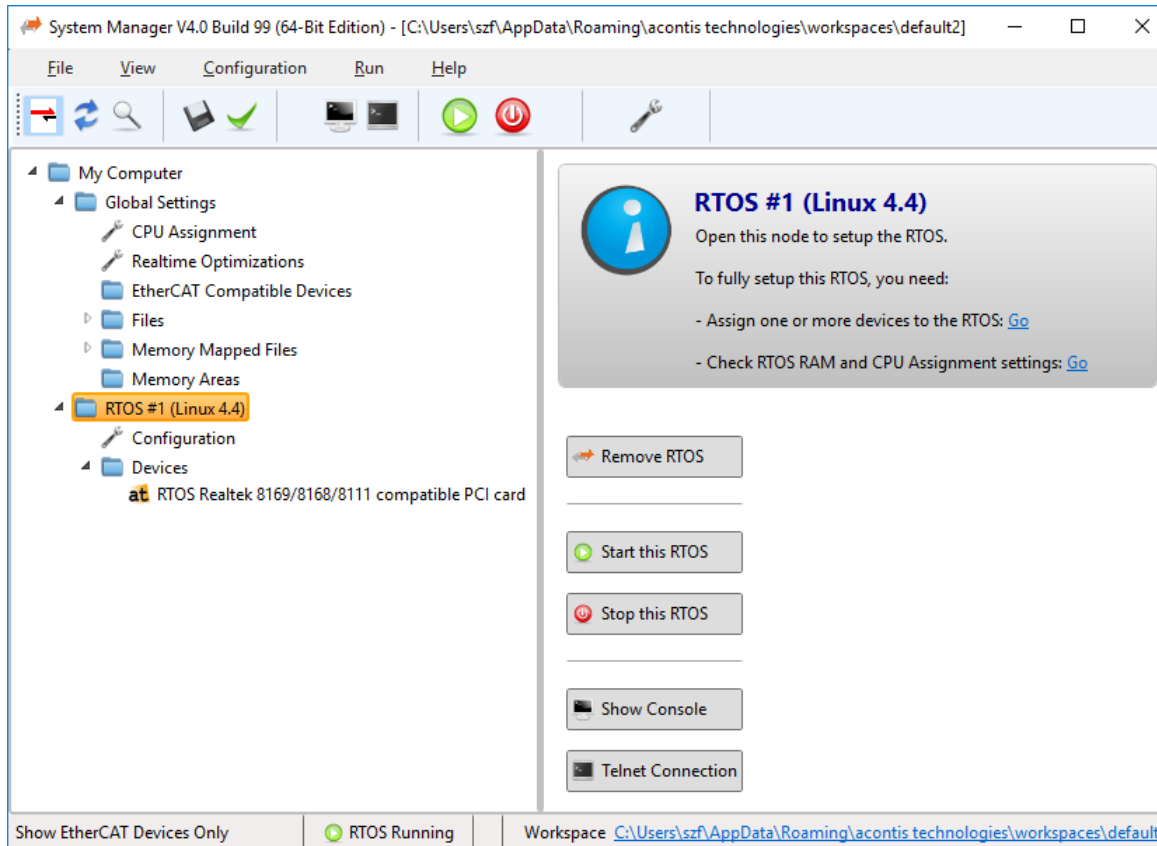


SYStem Manager Tool

Configuration and Development Management Console

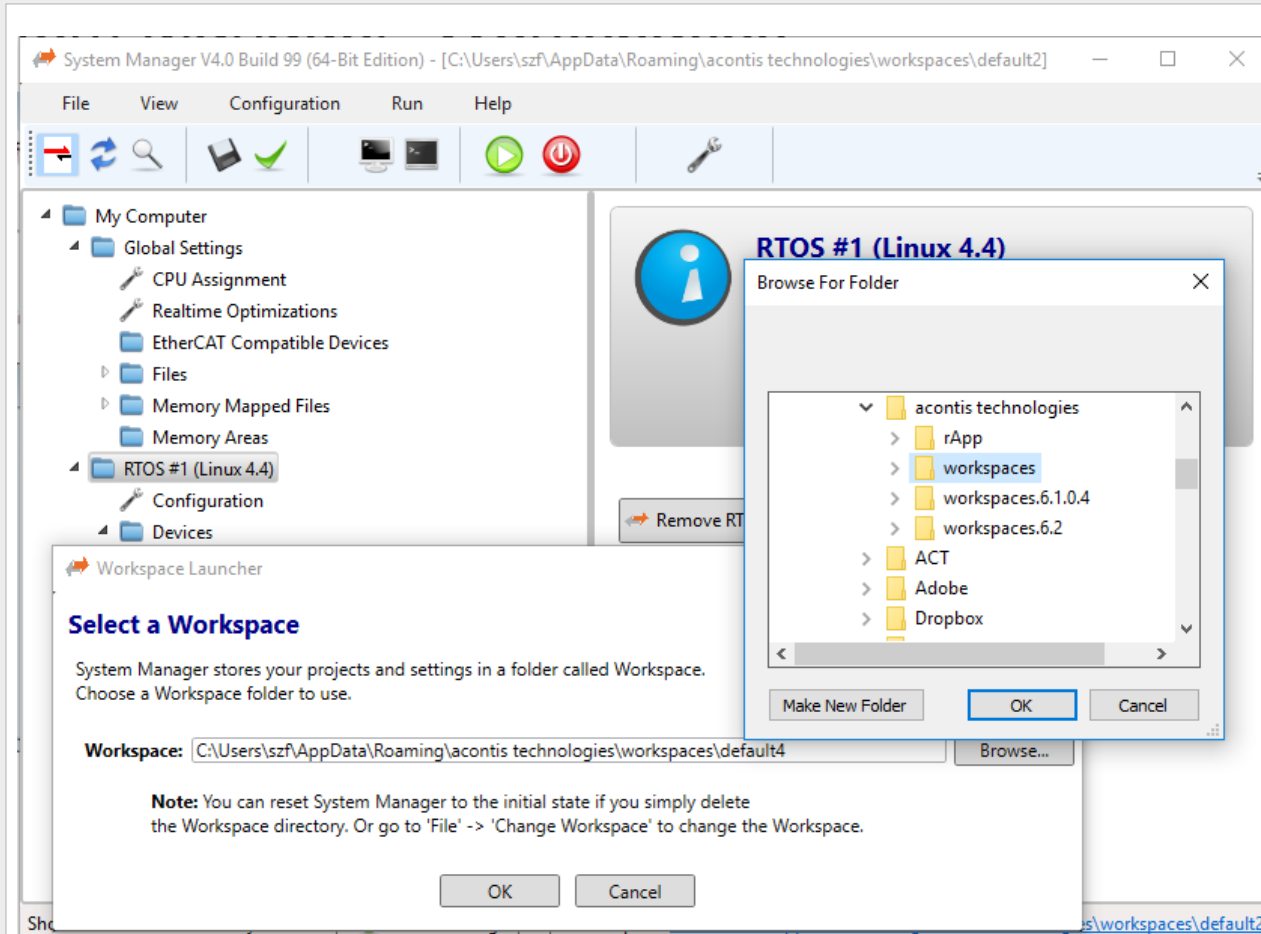
System Manager Tool: Management Console

One GUI for all major development tasks



- Hardware Partitioning
- System Configuration (e.g. RAM size for Linux)
- Launch Panel (e.g. start/stop Linux)
- Select and run pre-compiled demos
- Visual Studio Integration
 - create VS projects based on demo applications

System Manager: Workspaces



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

System Manager V4.0 Build 99 (64-Bit Edition) - [C:\Users\szf\AppData\Roaming\acontis technologies\workspaces\default2]

File View Configuration Run Help

My Computer

- Global Settings
 - CPU Assignment**
 - Realtime Optimizations
 - EtherCAT Compatible Devices
- Files
- Memory Mapped Files
- Memory Areas
- RTOS #1 (Linux 4.4)
 - Configuration
- Devices
 - RTOS Realtek 8169/8168/8111 compatible PCI card

CPU Assignment

Assign Operating Systems to the CPU's:

CPU#	Windows 10	Linux 4.4
CPU1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CPU2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CPU3	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CPU4	<input type="checkbox"/>	<input checked="" type="checkbox"/>

CPU assignment is valid

Reset Recommended

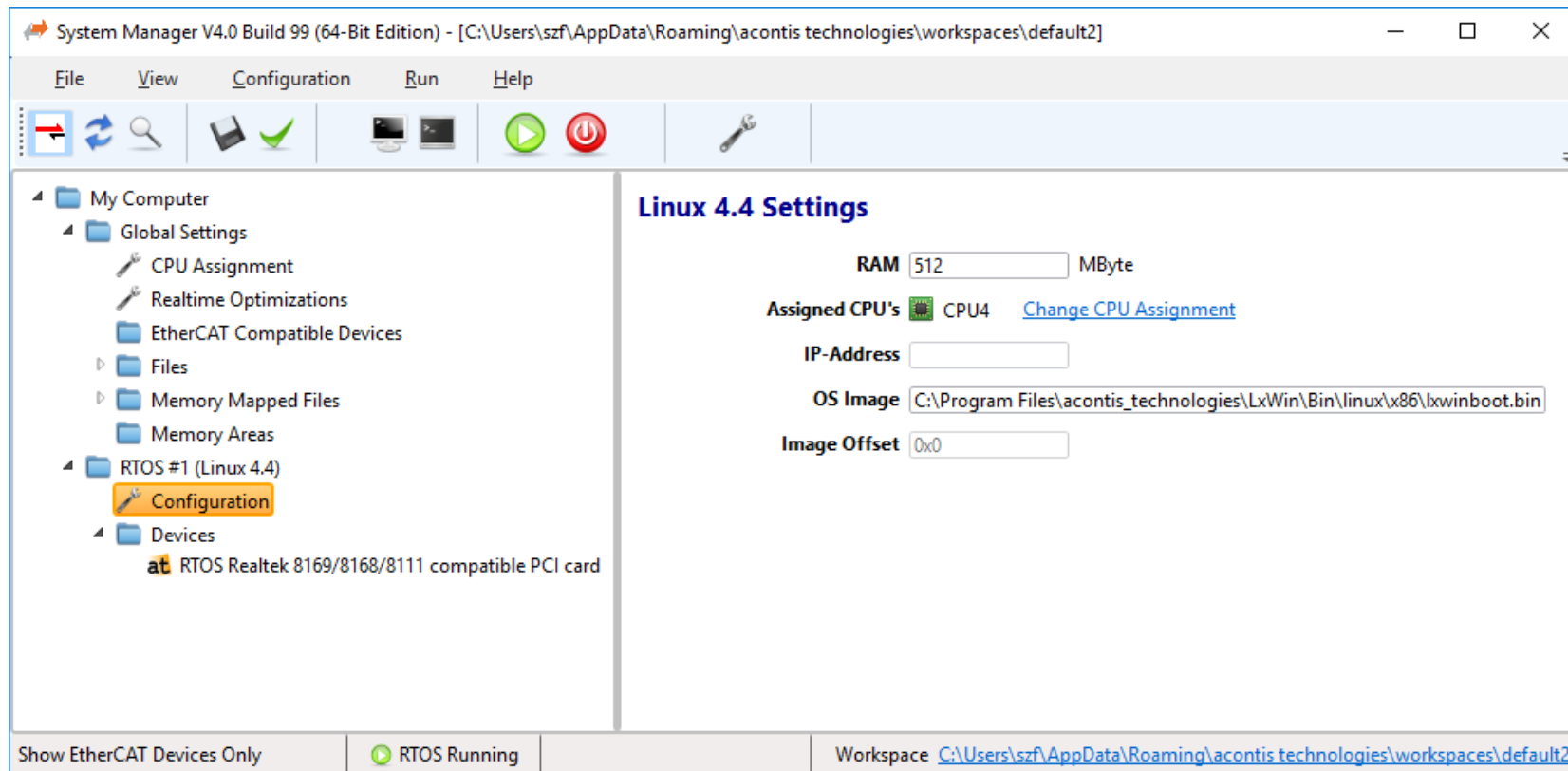
Recommended:
The recommended CPU configuration for this system will be created/restored. *After saving the changes a reboot maybe required!*

Reset:
All changes are reverted back. Exclusively reserved CPU(s) for the RTOS(es) will give(n) back to Windows. *After saving the changes a reboot maybe required!*

Show EtherCAT Devices Only | RTOS Running | Workspace: C:\Users\szf\AppData\Roaming\acontis technologies\workspaces\default2

- CPU Assignment
 - Select number of CPUs to be used by Windows
 - Select CPU(s) to be used by Linux

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



System Manager Tool: Hardware Partitioning



The screenshot shows the System Manager interface with the following components:

- Left Panel:** A tree view under 'My Computer' > 'Global Settings' > 'Devices' > 'Network adapters'. The 'Realtek PCIe GBE Family Controller' is selected and highlighted in orange.
- Right Panel:** Displays details for the 'Realtek PCIe GBE Family Controller'. It includes three assignment buttons: 'Assign to RTOS (Legacy Interrupt Enabled)', 'Assign to RTOS (MSI Enabled)', and 'Assign to RTOS (Interrupt Disabled)'. Below these is a 'Properties' button.
- Legacy Interrupt Usage Table:** A table listing interrupt numbers and their corresponding devices. The entry for interrupt 18, 'Realtek PCIe GBE Family Controller', is highlighted in green.
- Bottom Bar:** Shows 'Show All Devices', 'RTOS Stopped', and the current workspace path: 'C:\Users\szf\AppData\Roaming\acontis technologies\workspaces\default2'.

• Device Assignment

- Select Device currently controlled by Windows
- Assign Device to Linux

System Manager Tool: Hardware Partitioning

Device ready to use for Linux

The screenshot shows the System Manager V4.0 Build 99 (64-Bit Edition) interface. The window title is "System Manager V4.0 Build 99 (64-Bit Edition) - [C:\Users\szf\AppData\Roaming\acontis technologies\workspaces\default2]". The menu bar includes File, View, Configuration, Run, and Help. The toolbar contains icons for navigation, search, and device management. The left pane shows a tree view under "My Computer" with "RTOS #1 (Linux 4.4)" selected, and "Devices" expanded to show "at RTOS Realtek 8169/8168/8111 compatible PCI card". The right pane displays the device's configuration options:

- RTOS Realtek 8169/8168/8111 compatible PCI card**
- RTOS Realtek 8169/8168/8111 compatible PCI card
- VendorId 10EC DeviceId 8168 (Bus 03 Dev 00 Fct 00)
- Interrupt is enabled (MSI)
- Assign to Windows
- IRQ Property: Legacy
- IRQ Property: MSI
- IRQ Property: None
- Rename Device
- Properties

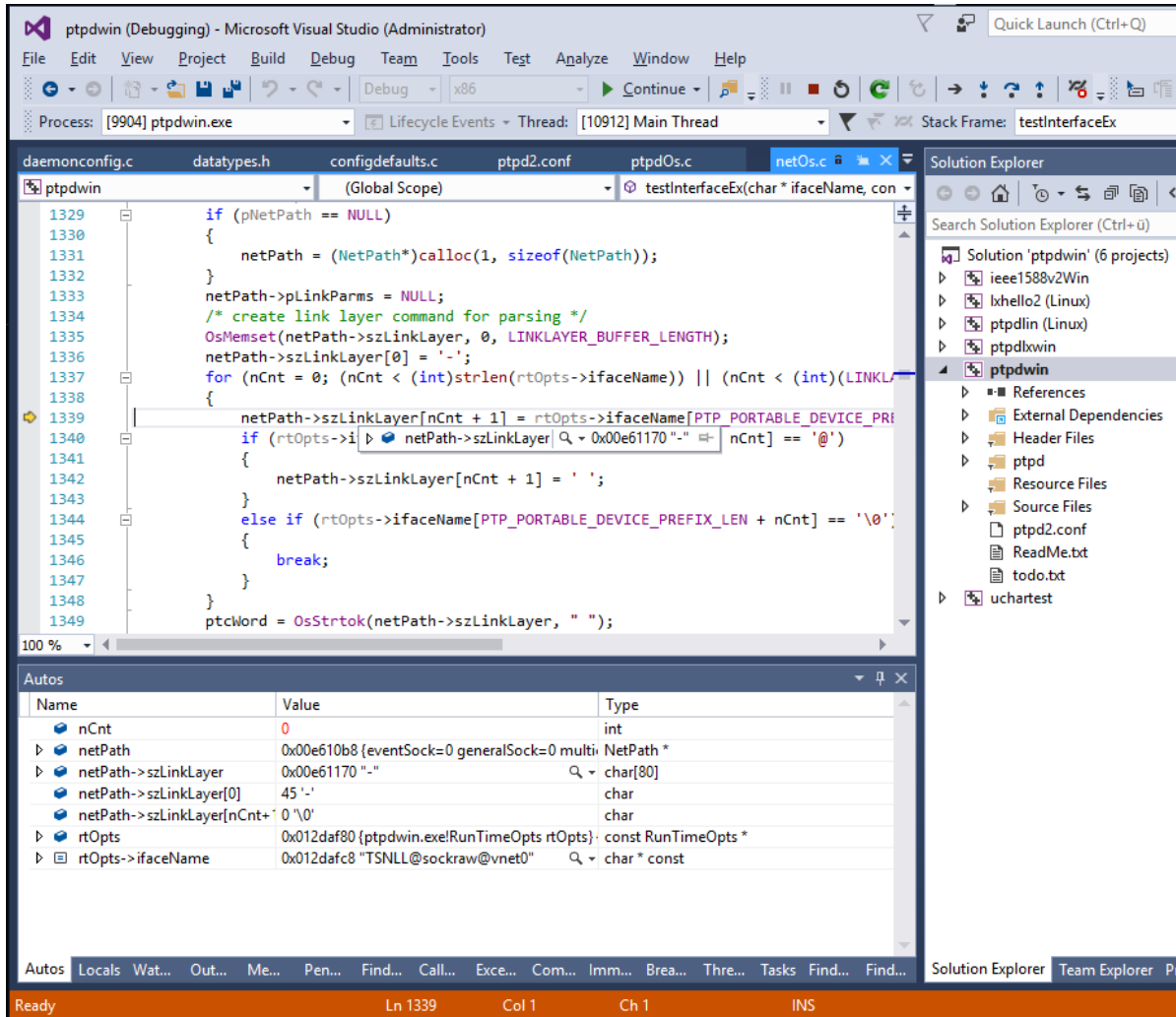
The status bar at the bottom shows "Show EtherCAT Devices Only", "RTOS Stopped", and "Workspace C:\Users\szf\AppData\Roaming\acontis technologies\workspaces\default2".

LxWin[®]

SOFTWARE Development

Run Real-time applications on Windows

VisualGDB: MS Visual Studio Support for developing real-time applications



- Powerful VisualGDB plugin for real-time application development
- Project Wizards
 - Automatically Create new real-time applications
- Powerful Graphical Debugger
 - As convenient as debugging a standard Windows application
- Analysis tools
 - Valgrind based dynamic analysis (e.g. memory leak or race condition detection)
 - Code Coverage reports

Fast and deterministic timer: Windows with Real-Time

Shortest possible cycle time: 1 millisecond

High jitter, huge delays

Code snippet for 1 msec cycle controller:

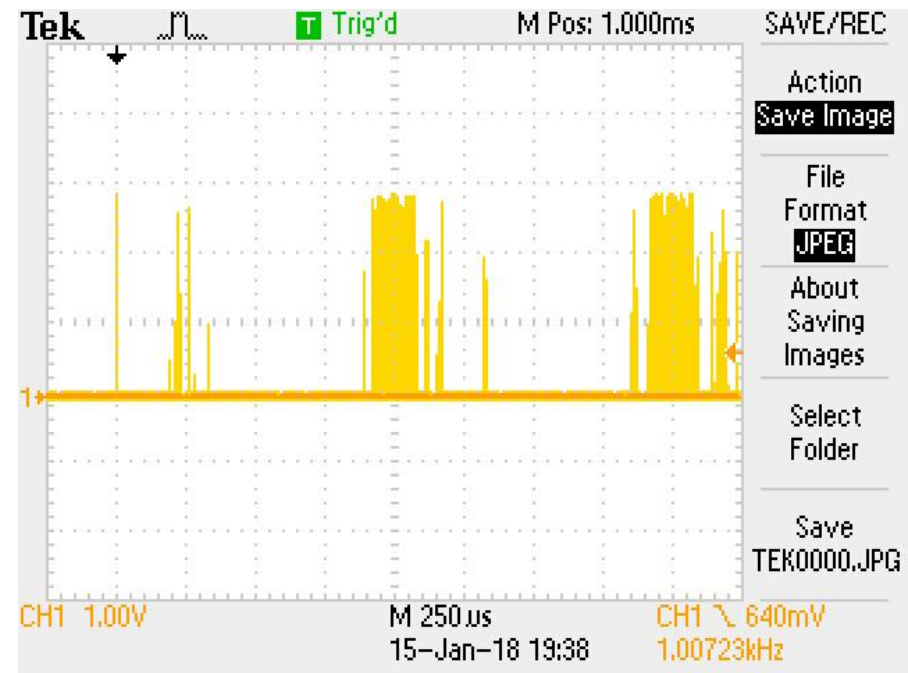
```
timeBeginPeriod(1);  
for (;;)   
{  
    Sleep(1);  
    RunController();  
}
```

Screenshot

1 msec cycle

high average Ethernet send jitter

Sometimes huge delays > 100 msec!

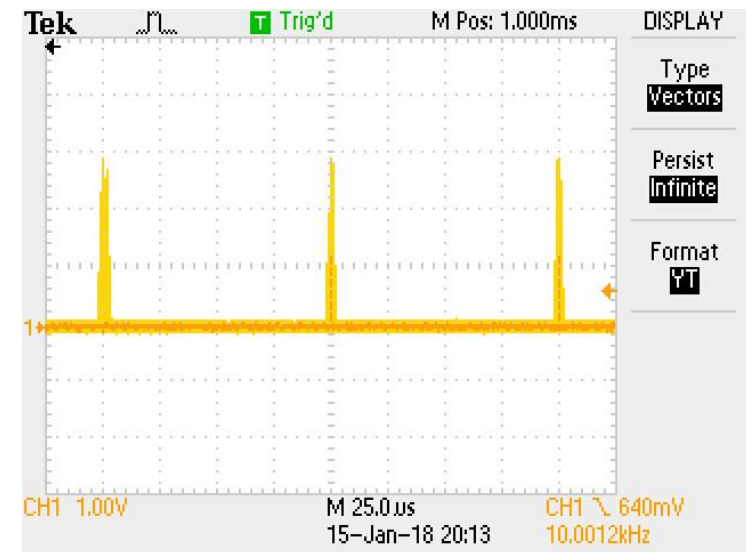
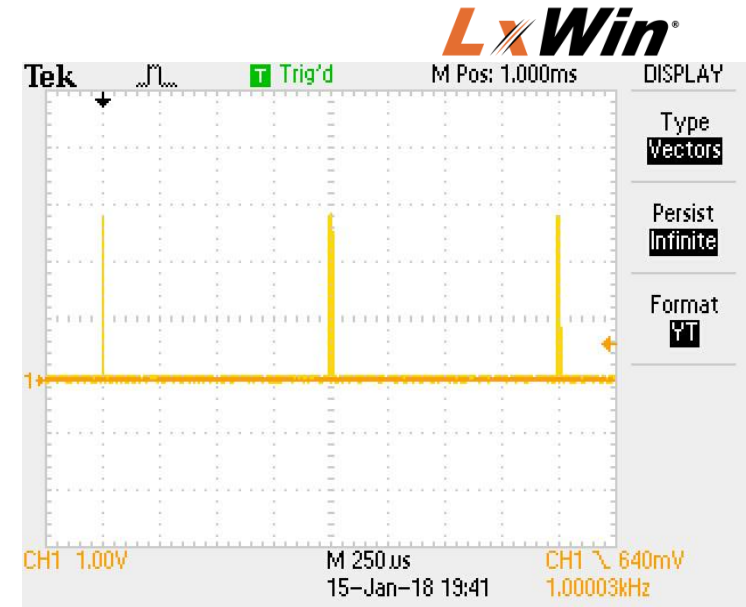


Fast and deterministic timer: Windows with LxWin

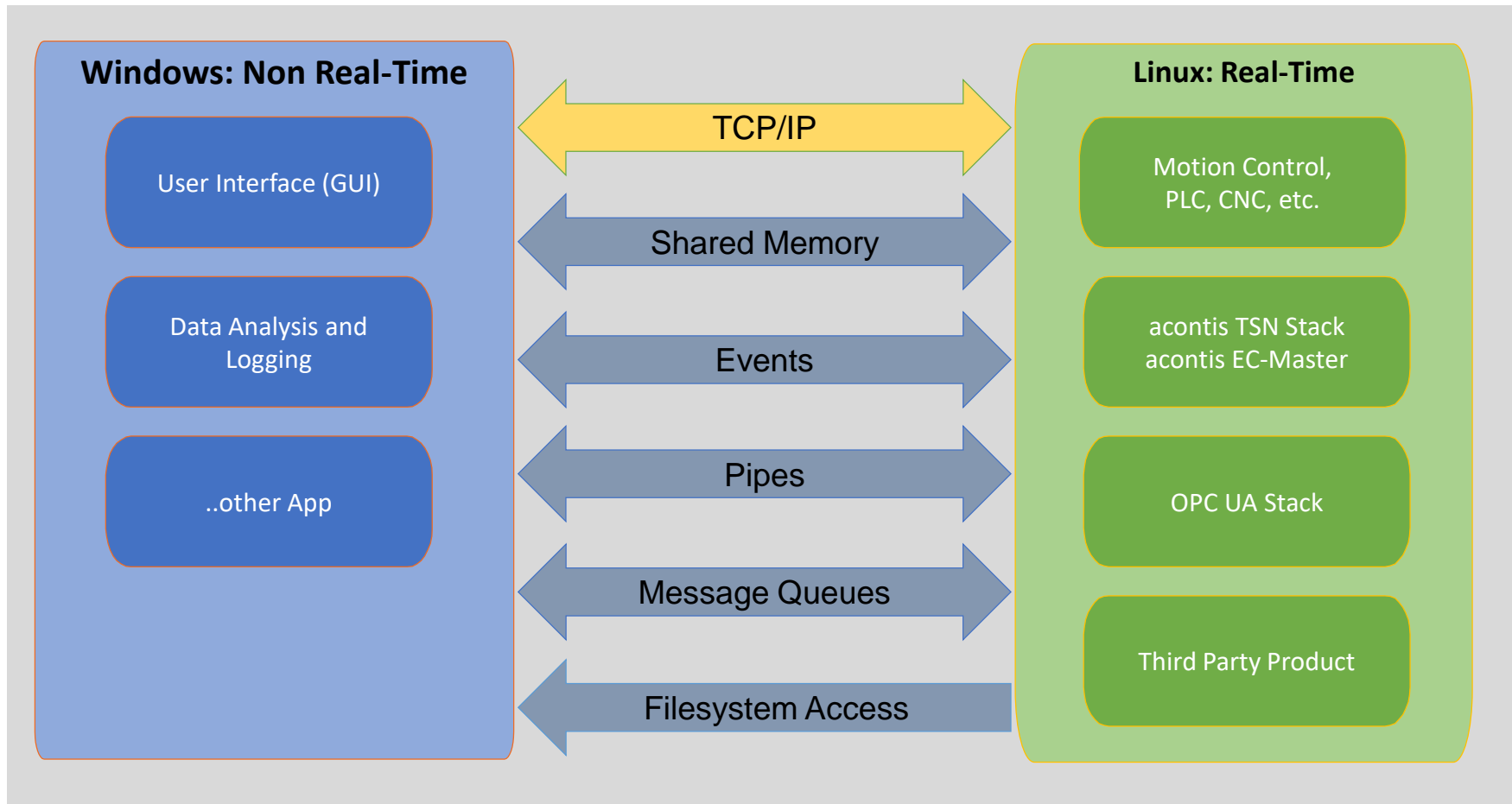
- Up to 50 usec cycle time
- Low Jitter
- Simple to use
- Code snippet for 100 usec cycle controller:

```
struct timespec ts;  
ts.tv_sec=0; ts.tv_nsec=100000;  
for(;;)  
{  
    clock_nanosleep(CLOCK_MONOTONIC,0,&ts,NULL);  
    RunController();  
}
```

- Screenshots
 - Top: 1 msec cycle – Bottom: 100 usec cycle
 - Low Ethernet send jitter
 - Deterministic!
 - Very short cycle times up to 50 usec



Run Real-time applications on Windows

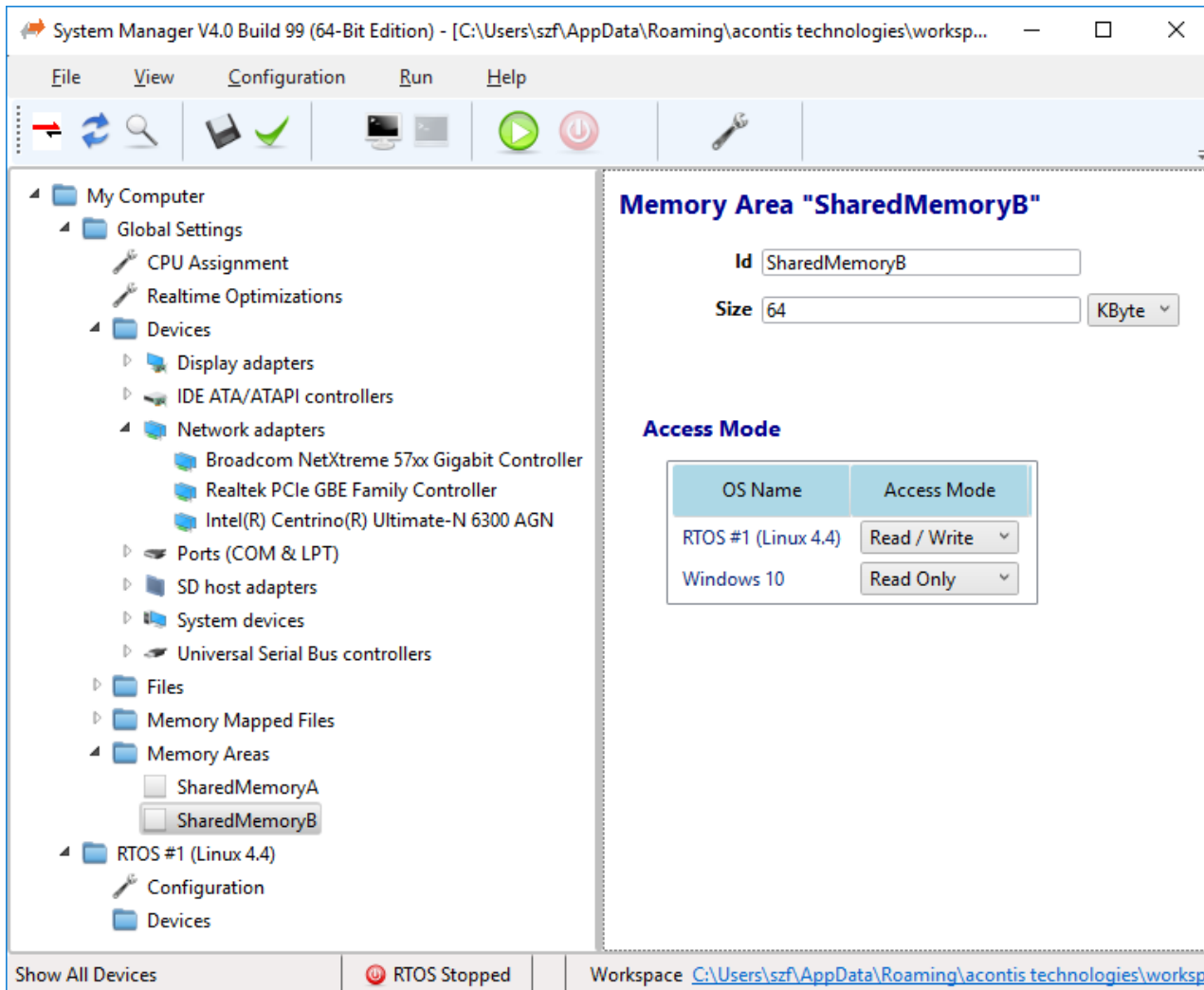




Communication

Data and information exchange between Windows and Linux

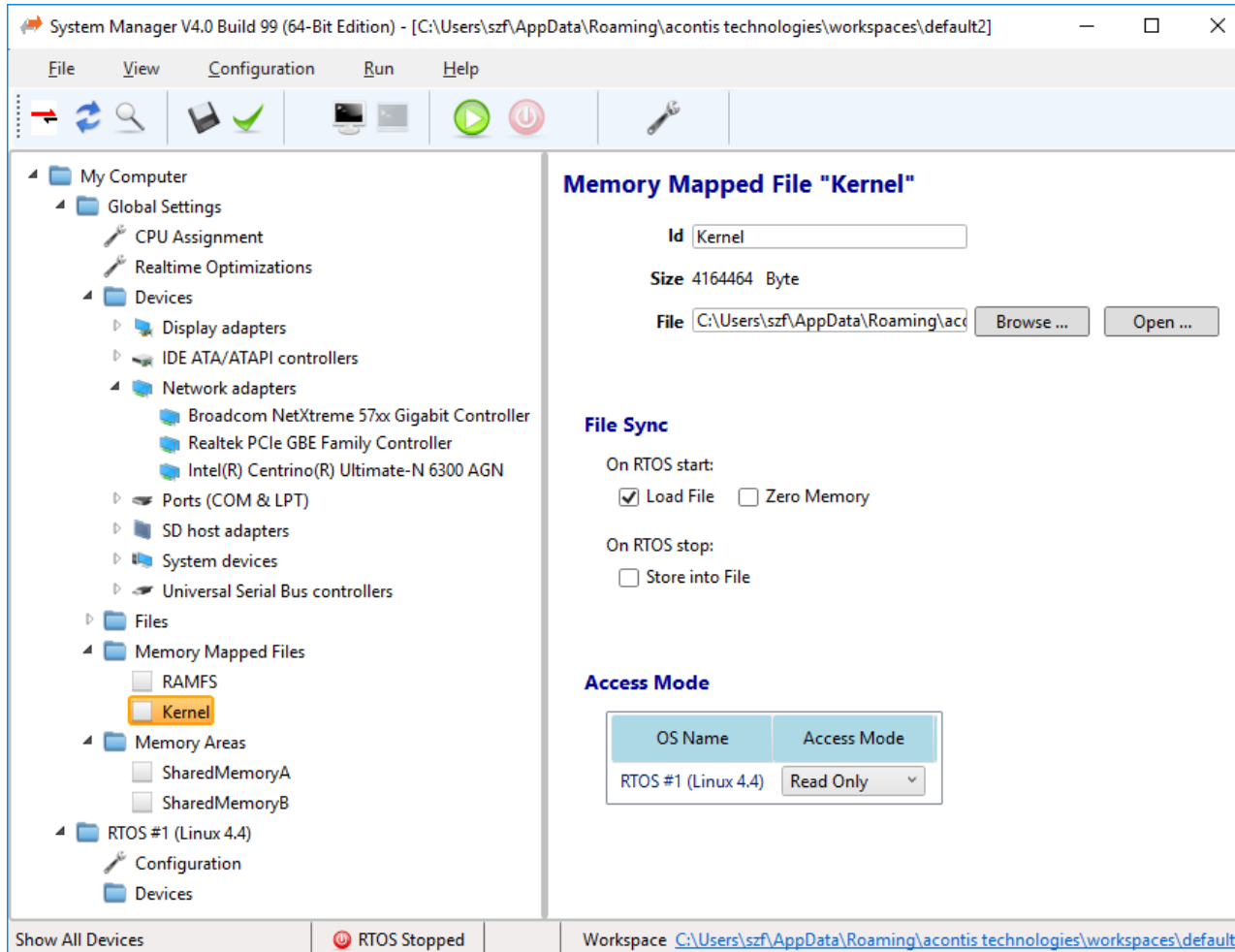
Low level communication: Shared Memory Areas + Events



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

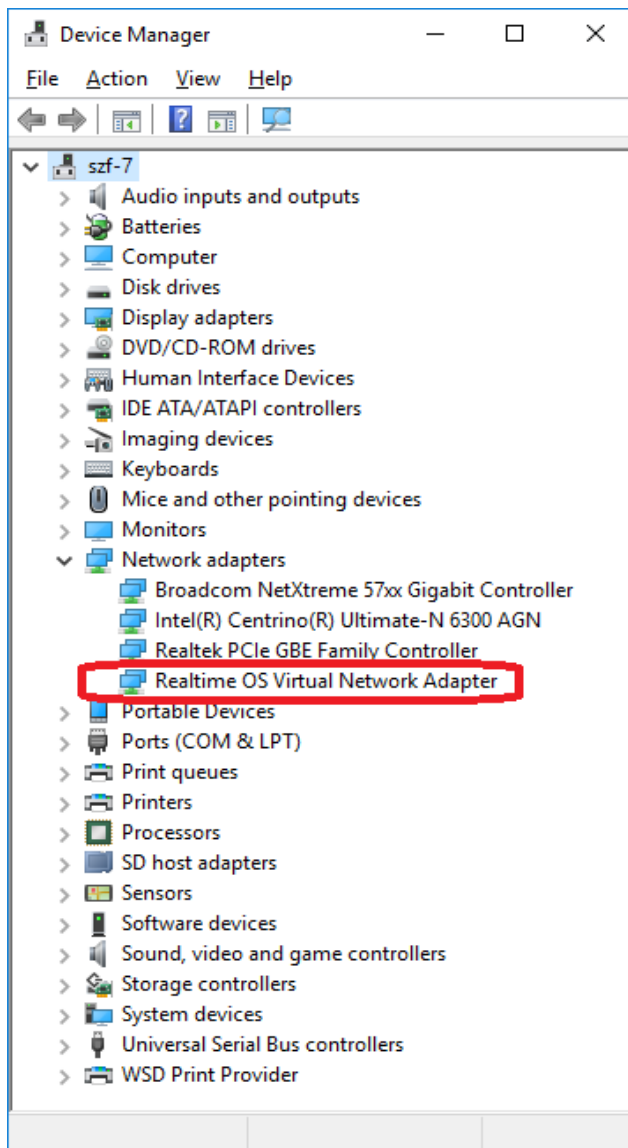
- Named message queues and pipes
 - Telegram or stream oriented communication
- Real-time Socket API
 - Create applications which can run locally using Shared Memory or remote using TCP/IP
- Remote TCP Gateway
 - Routing TCP to Shared Memory
 - Connect external tools with Windows IP address
 - automatic forwarding to acontis EC-Master EtherCAT 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, ...)

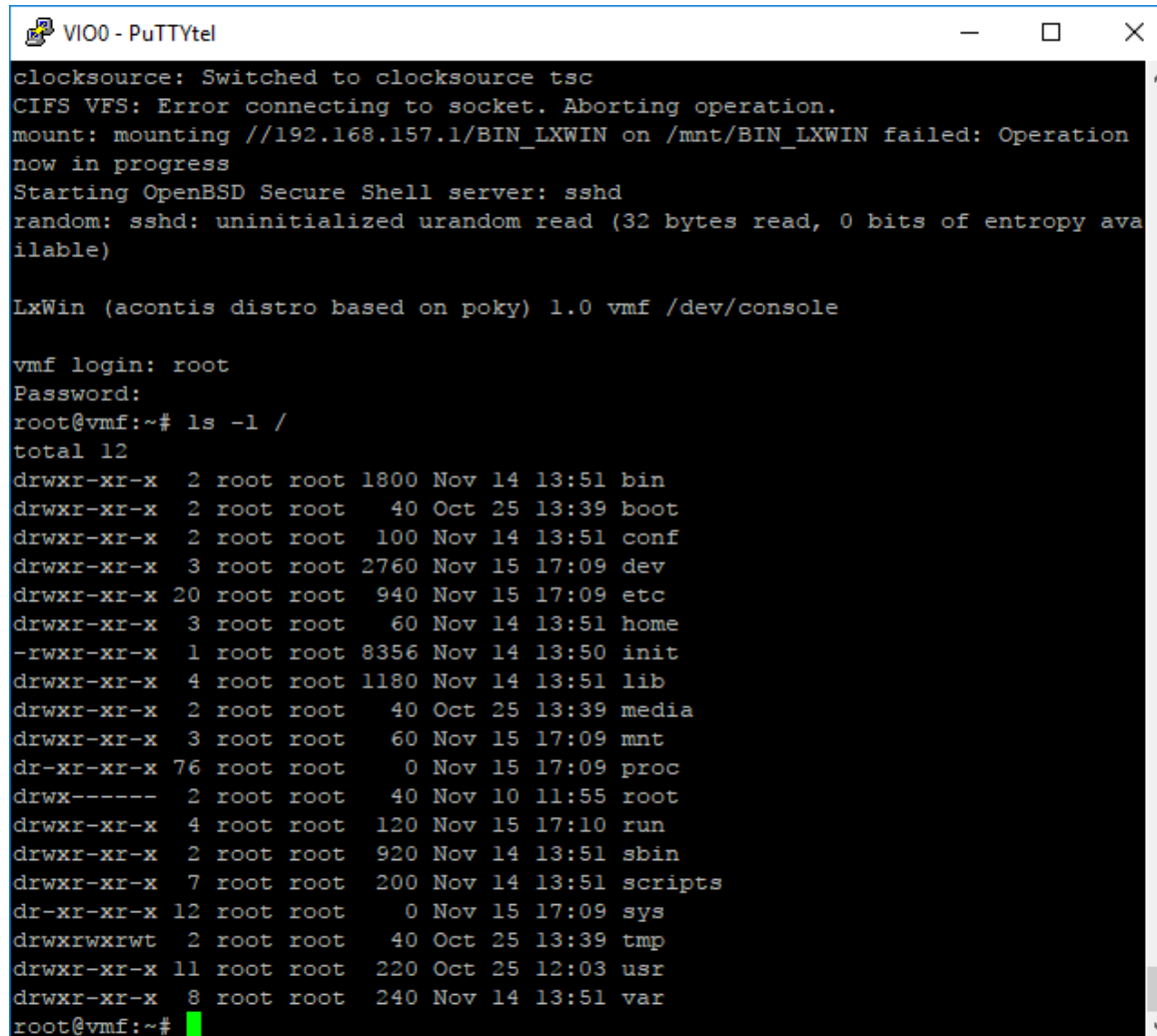
Communication: Virtual Network



- Used for application debugging
- TCP/IP
 - Useful for scalable solutions
 - e.g. same application to communicate with external Windows tools

Virtual Console

- Can be used for Linux Shell



```
clocksource: Switched to clocksource tsc
CIFS VFS: Error connecting to socket. Aborting operation.
mount: mounting //192.168.157.1/BIN_LXWIN on /mnt/BIN_LXWIN failed: Operation
now in progress
Starting OpenBSD Secure Shell server: sshd
random: sshd: uninitialized urandom read (32 bytes read, 0 bits of entropy ava
ilable)

LxWin (acontis distro based on poky) 1.0 vmf /dev/console

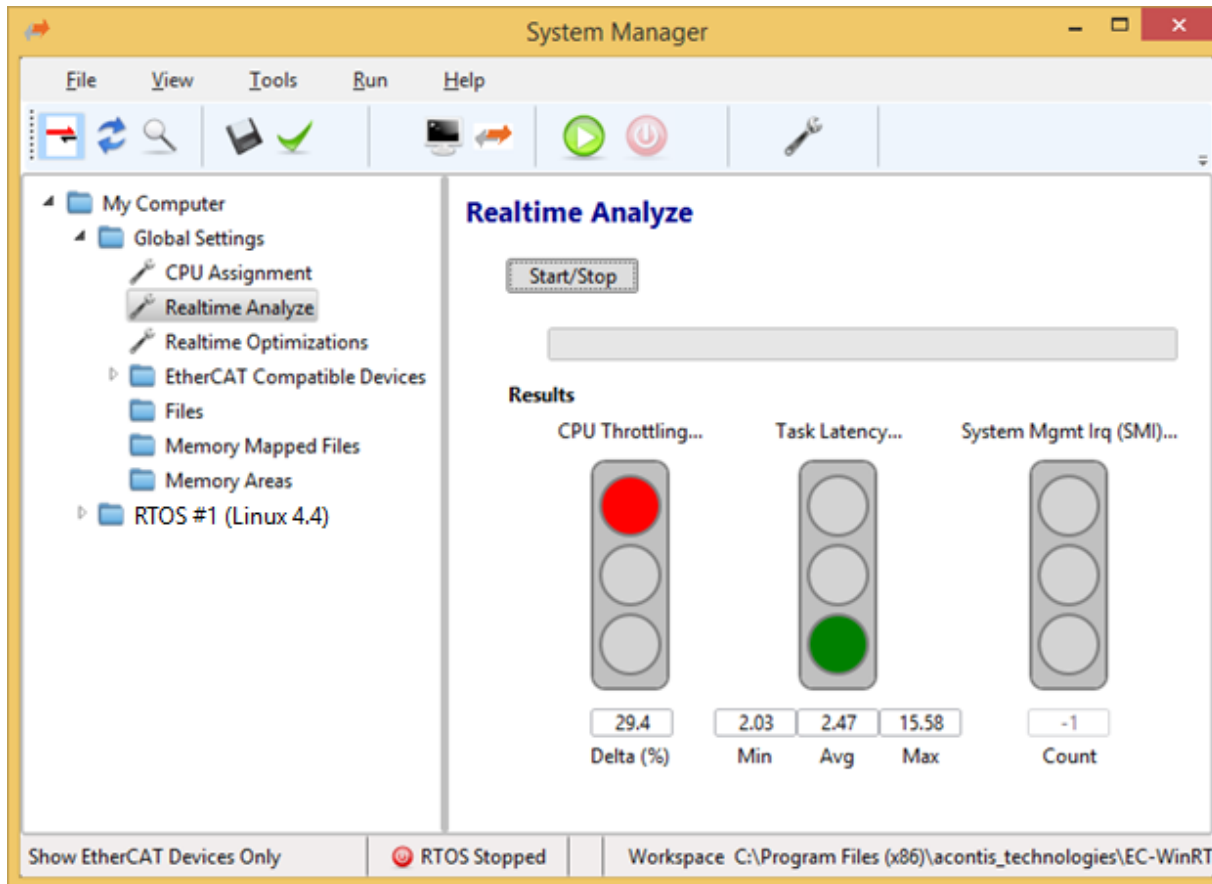
vmf login: root
Password:
root@vmf:~# ls -l /
total 12
drwxr-xr-x  2 root root 1800 Nov 14 13:51 bin
drwxr-xr-x  2 root root  40 Oct 25 13:39 boot
drwxr-xr-x  2 root root  100 Nov 14 13:51 conf
drwxr-xr-x  3 root root 2760 Nov 15 17:09 dev
drwxr-xr-x 20 root root  940 Nov 15 17:09 etc
drwxr-xr-x  3 root root  60 Nov 14 13:51 home
-rwxr-xr-x  1 root root 8356 Nov 14 13:50 init
drwxr-xr-x  4 root root 1180 Nov 14 13:51 lib
drwxr-xr-x  2 root root  40 Oct 25 13:39 media
drwxr-xr-x  3 root root  60 Nov 15 17:09 mnt
dr-xr-xr-x 76 root root   0 Nov 15 17:09 proc
drwx----- 2 root root  40 Nov 10 11:55 root
drwxr-xr-x  4 root root  120 Nov 15 17:10 run
drwxr-xr-x  2 root root  920 Nov 14 13:51/sbin
drwxr-xr-x  7 root root  200 Nov 14 13:51/scripts
dr-xr-xr-x 12 root root   0 Nov 15 17:09 sys
drwxrwxrwt  2 root root  40 Oct 25 13:39 tmp
drwxr-xr-x 11 root root  220 Oct 25 12:03 usr
drwxr-xr-x  8 root root  240 Nov 14 13:51 var
root@vmf:~#
```

LxWin[®]

Real-time on Windows

Real-time analysis and optimization

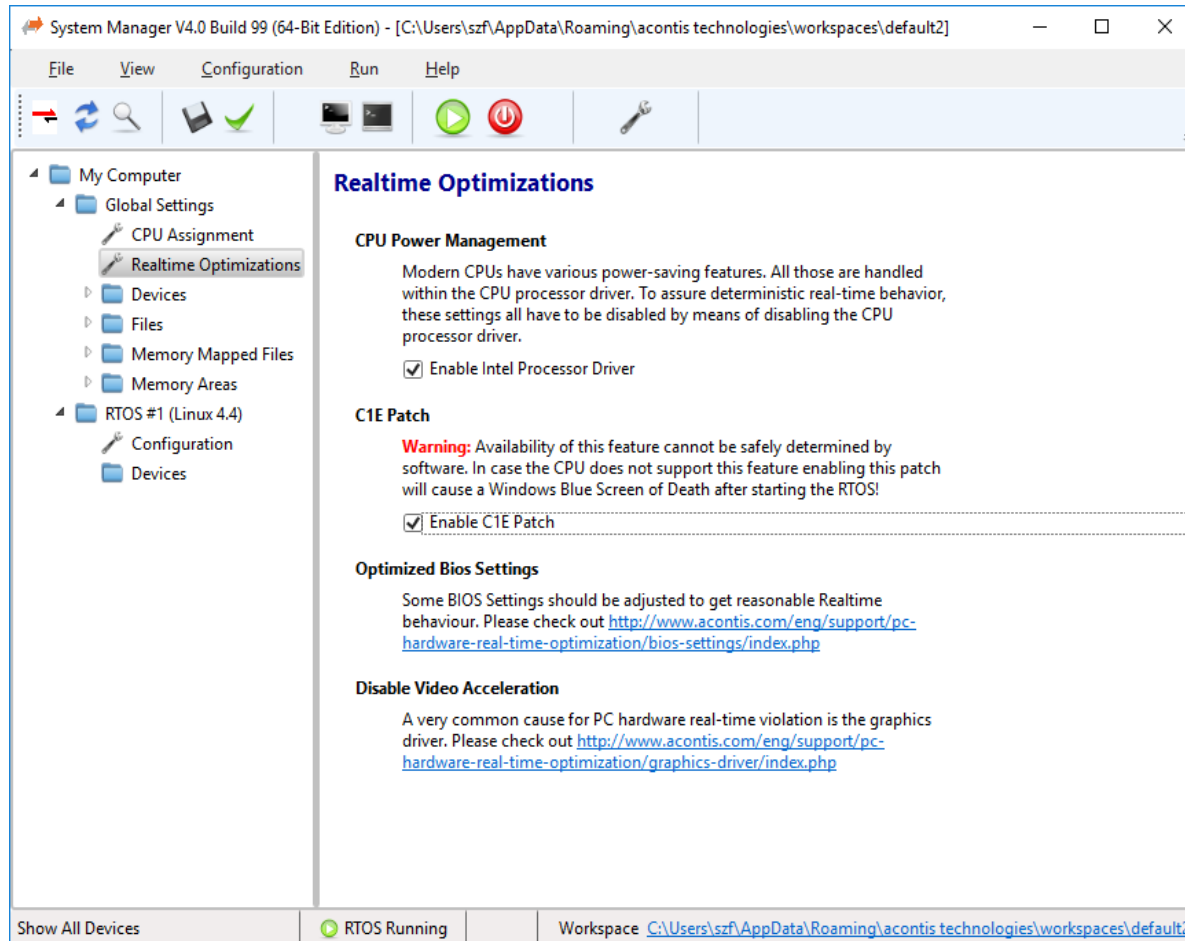
System Manager: Real-time analyzation (no optimization)



• Results

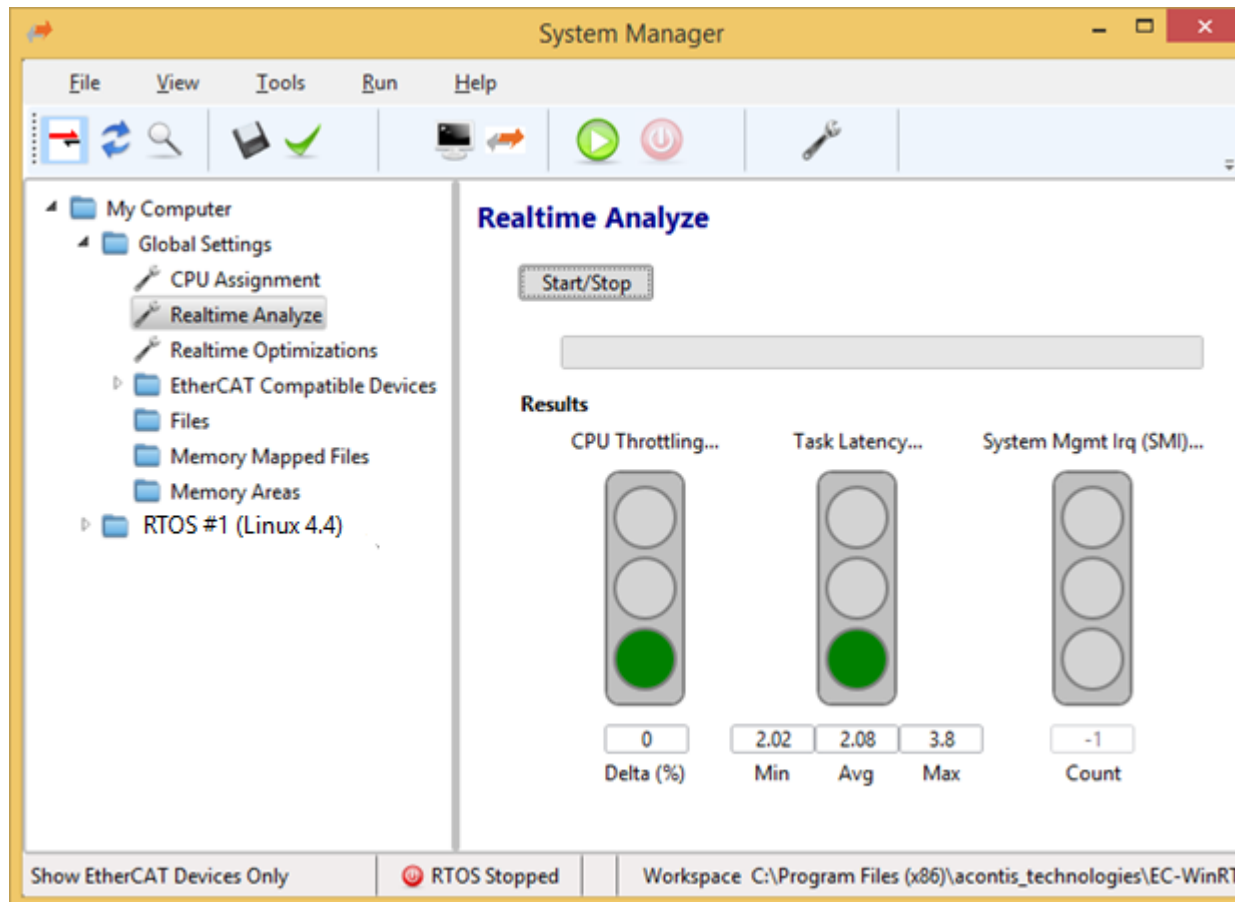
- CPU clock: throttling active?
- Timer: task level latency
- System Management Interrupts?

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



- 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!

- Hypervisor based real-time Linux solution for Windows
 - Run hard Real-time Linux applications alongside Windows
 - RT Preempt patch support out of the box
 - Secure separation of Windows and Real-time Linux
- Communication (Windows \leftrightarrow Real-time Linux)
- Linux development and debugging tools supported
 - Eclipse
 - Visual Studio 2015/2017
- Development Console: System Manager

LxWin[®]

Intel VT support

- Better isolation: Fatal Linux kernel crash does not violate Windows
 - Exit into Virtual Monitor instead of system reboot
- Utilize more memory for Linux
 - Physical memory > 4GB can be used, even for 32 Bit Linux
 - Non-contiguous Windows Memory can be used
 - Example: 4 times 500 Mbyte Windows memory = 1 time 2 Gbyte Linux memory
- Support Shared Mode for Windows 64 Bit
 - Run Windows and Linux on the same physical core
- Activate via configuration setting
 - Linux image and applications do not need to be changed

LxWin[®]

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)

