

# KUKA



## **Add Windows® XP functionality to VxWorks® on single or multicore CPUs**

**KUKA Roboter GmbH  
Global Sales Center  
Hery-Park 3000  
D-86368 Gersthofen  
Germany**

**Tel.: +49 821 4533-3768  
Fax: +49 751 4533-2129  
Email: [RainerGallus@KUKA-Roboter.de](mailto:RainerGallus@KUKA-Roboter.de)  
Web: [www.kuka-rtos.com](http://www.kuka-rtos.com)**

**Edition: April 2008**

## Introduction

### Wind River VxWorks

Wind River VxWorks is a widespread real-time operating system (RTOS), produced by Wind River. Several versions of VxWorks are available for different processor architectures (Motorola 68k, Intel 80x86, Sparc, AMD 29k and others). VxWorks traditionally runs standalone on a single processor and uses the RAM, ROM, I/O hardware resources, and so on, exclusively.

### Microsoft Windows XP

Microsoft Windows XP is the dominate operating system in use today on x86 desktop PCs worldwide. It has a very familiar graphical user interface, but due to the absence of real-time ability, it is not suitable for most industrial applications. The real advantages of Windows XP, however, are to be found in its widespread application base and from both its acceptance and familiarity by users. To make Windows XP useful for industrial real-time applications, such as industrial automation, motion control, test and measurement, medical devices and defense systems, KUKA Roboter has developed a solution which eliminates the disadvantages of Windows XP without giving up its advantages.

### VxWin: The Windows XP extension for VxWorks

KUKA Roboters VxWin® extends the VxWorks real-time operating system with Windows functionality; allowing both operating systems to run concurrently on the same x86 based computer. The requirement for a separate dedicated computer running a Windows-based user interface is eliminated. VxWin does not make proprietary modifications to either OS and maintains complete binary compatibility with both standalone VxWorks and Windows XP. Your application gets the best of both worlds.

To make this possible, KUKA Roboter has developed a software only, real-time extension technology. KUKA's technology guarantees deterministic response on interrupts which are targeted for VxWorks. These interrupts can be generated by any PCI-, PCIe or ISA- PC plug-in board controlled by VxWorks. The real-time interrupts always have a higher priority than anything occurring in Windows.

#### Single Core Systems:

As long as at least one VxWorks task is active, the processor's execution time is available exclusively for VxWorks. Windows XP runs only if all VxWorks tasks are idle. Windows XP effectively operates instead of the VxWorks idle task. Virtual machine technology or time slicing is not utilized by VxWin. During all real-time processing, VxWorks has exclusive use of the processor. Conversely, as long as no real-time interrupts are being received, Windows XP has exclusive control. No proprietary modifications are made to either OS which insures binary compatibility with standalone versions of each OS.

#### Dualcore or Multicore Systems:

When VxWin is operating in exclusive mode on a dualcore or multicore system, VxWorks usually runs on the last processor core while Windows XP runs on all other. Both operating systems operate completely concurrently and independent from each other. In this operation mode, latencies and jitter is dramatically reduced compared with a singlecore system.

### **Target Market**

- Real-Time applications requiring high-end user-interface and connectivity
- PC-based industrial automation; including PLC, motion control, robotics, CNC
- Real-time test and measurement/data recording
- Medical devices and equipment
- Military and Aerospace
- PC and HMI add-on for existing VxWorks applications
- VxWorks target simulation with real-time behavior and hardware access

### **Technical features**

- Full use of the VxWorks RTOS concurrent with Windows XP
  - Multitasking, intertask communication
  - TCP/IP Stack
  - Realtime processes
  - Powerful shell and optional products
- APIC and multiprocessor support
- Exception handling (Bluescreen)
- Software-only solution, no additional hardware necessary
- Complete binary compatibility with standalone VxWorks
- Communication between Windows XP and VxWorks via shared memory
- Mature development environments from Wind River and Microsoft
- Graphical real-time analysis tools
- Two high-resolution real-time timers
- Hardware access to ISA, PCI and PCIe plug-in boards
- Access to powerful Windows XP API
- Deterministic Performance / Task Scheduling
- Full debug support (incl. remote debugging)
- Time synchronization between VxWorks and XP
- VxWin can start as a service – no user login to Windows XP is required
- Auxiliary system clock with frequencies up to 25kHz
- Most common Microsoft-supplied HALs are compatible
- Support for VxWorks versions 5.4, 5.5 and 6.x

### **Commercial Features**

- No need for separate dedicated system running a Windows-based user interface
  - Reduction of hardware costs; eliminates a second CPU
  - Less hardware components: increased reliability (MTBF)
  - More compact control systems possible
- Nonproprietary ~ built on COTS operating systems
- Full support from a company with worldwide locations
- Short learning curve: known development tools (Visual Studio, Tornado and Workbench)
- Binary compatibility allows reuse of Windows and VxWorks software and knowledge

### **Application Development**

A separate host is not necessary for application development - the host and target system are identical. Both Wind River Workbench / Tornado and Microsoft Visual Studio can reside on the VxWin development PC. A VxWin development seat provides a BSP in the form of a Wind River Tornado or Workbench project. The VxWin SDK contains all the necessary header and library files needed to build VxWin applications. Components for integration into Windows XP Embedded images are also included.

The VxWorks part of the application is edited, compiled and debugged using the tools offered by Wind River Tornado or Workbench. Additional tools such as WindView and ScopeTools Suite are also supported. The Windows XP part is developed using Microsoft Developer Studio. Develop, test and debug the actual target, as opposed to a simulated target, on the same PC. VxWin is started and stopped using the Uploader tool. This tool can be used stand-alone or integrated directly into a VxWin application using the supplied DLL.

## **Communication between VxWorks and Windows XP**

Since Version 3.0, VxWin applications use either a direct shared memory interface or the TCP/IP-protocol for communication between VxWorks and Windows XP. For a direct shared memory interface, applications allocate memory from a shared memory pool. The shared memory is read from or written to by both the VxWorks and Windows XP parts of the application. A shared event mechanism is provided to coordinate reading and writing of the shared memory.

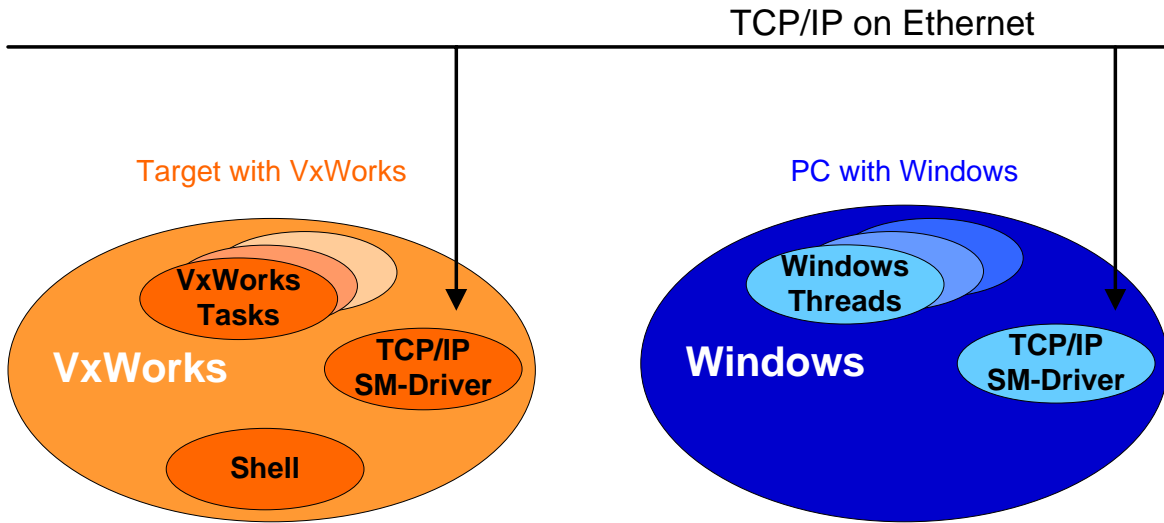
The TCP/IP network operates over an internal, dedicated shared memory area. For this purpose two corresponding network drivers have been developed; one for each operating system. Both systems can exchange data over the commonly accessible shared memory area as they would via an Ethernet line.

The network communication architecture has the advantage that both the Windows XP and VxWorks parts of the application can be separated and connected by a physical Ethernet line without having to modify the original application. This makes sense if the real-time target system is not only PC-based but also an embedded system not requiring Windows XP functionality. Conversely, existing VxWorks applications communicating with a Windows XP based front-end can be easily ported to VxWin.

Using the Windows XP built-in IP-Forwarding feature, VxWorks can access other systems connected to the Windows XP TCP/IP network. Also, an Ethernet card can be configured so that it is controlled directly by VxWorks. This card may be used to facilitate a direct TCP/IP connection between VxWorks and systems external to the VxWin PC. Any Ethernet card supported by VxWorks may be used.

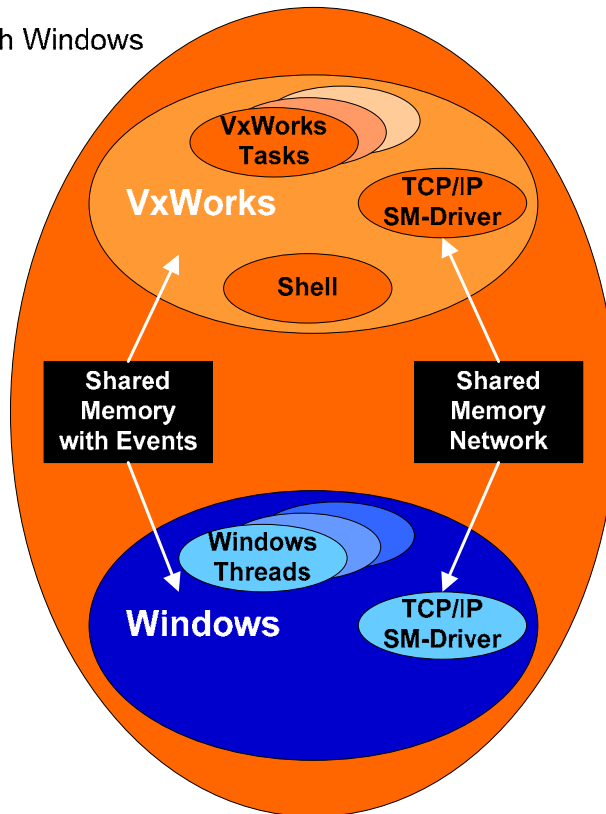
Other benefits of using the standard TCP/IP protocol include the ability to make use of all the network-based VxWorks tools, e.g., WindView, ScopeTools Suite, etc. The runtime system can make use of TCP/IP sockets for implementing proprietary communication protocols. From the point of view of VxWin applications utilizing the VxWin shared memory network, there is no difference between running on the same PC or running on two separate systems.

**Traditional Solution:**

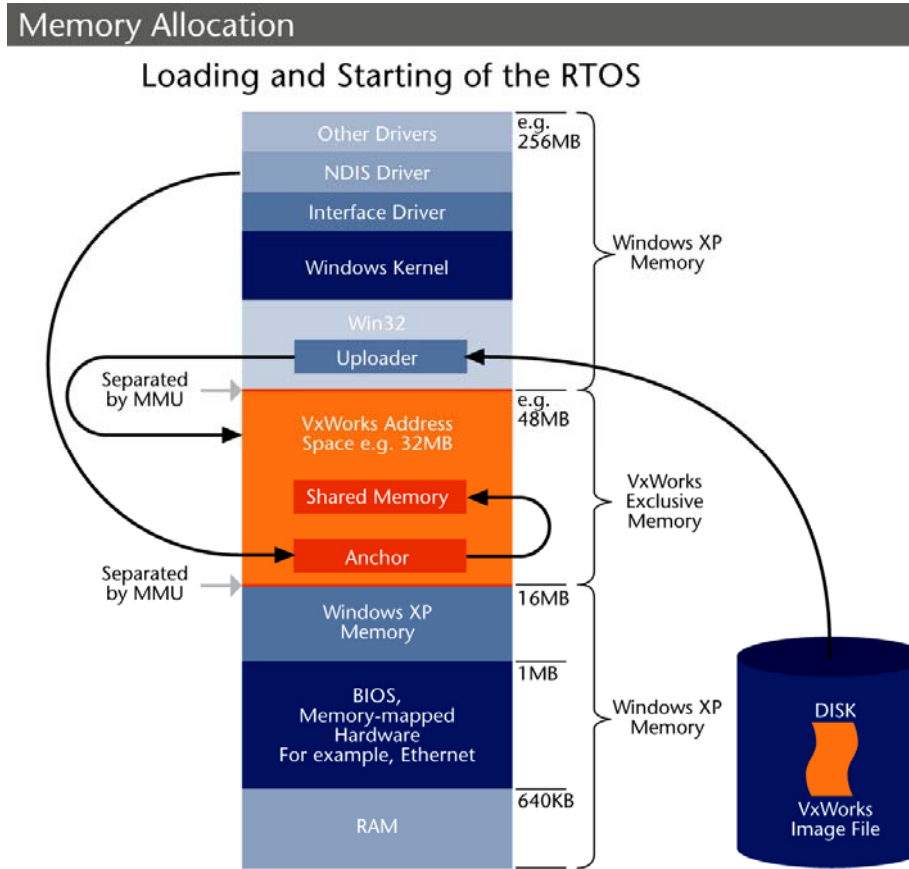


**VxWin Solution:**

PC with Windows



## Start-Up Sequence of VxWin

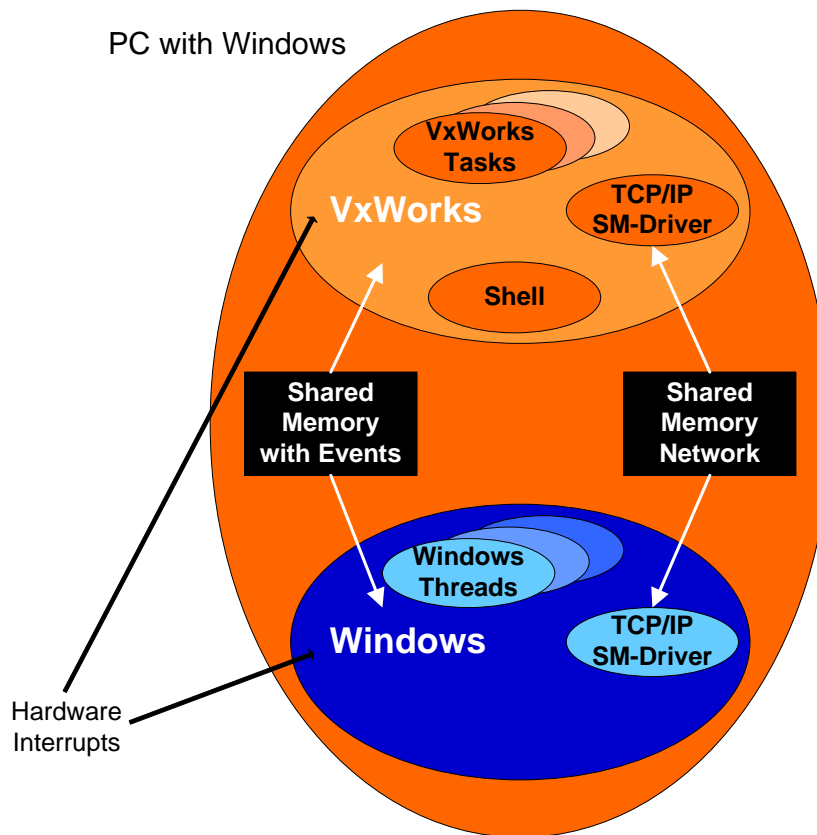


While Windows XP is booting, a section of memory is partitioned that is reserved for VxWorks. Both the size and location of the partitioned memory is configurable. Consistency checks are performed to insure the validity of the configuration. The partitioned memory is guaranteed to be fixed and cannot be swapped to disk. The code running in this memory executes with the highest privilege level (Ring 0). Additionally, the partitioned memory is protected by the hardware memory protection facilities of the x86 MMU. Windows XP is not aware that the memory exists.

After Windows XP is up and running, VxWin is started using the Uploader tool. The Uploader is a Win32 application, supplied with VxWin, which can be easily integrated into the user's application. The Uploader communicates with the VxWin interface driver loaded during the boot sequence of Windows XP. The Uploader calls the interface driver to allocate the memory, partitioned during boot process of Windows XP, to be used for VxWorks code and data.

The next step of the Uploader is to read the VxWorks image from the local disk and copy it to the allocated memory on behalf of the driver. After the code and data are relocated to the loading address, VxWorks is started by calling its `init()` function. As soon as all initialization tasks are idle, the `init()` function returns and the Uploader terminates. From this point forward, VxWorks will be activated only by real-time interrupts followed by task scheduling.

## Singlecore Systems: Activation of VxWin by interrupts

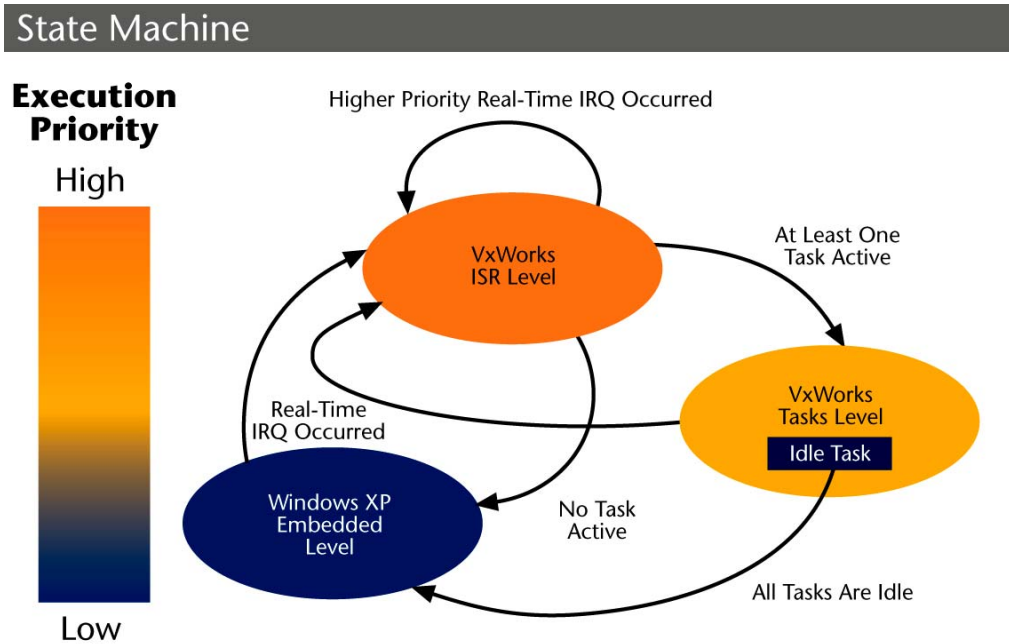


All incoming interrupts for VxWorks, as well as for Windows XP, are received by the VxWin interface driver directly from the PC's interrupt controller. KUKA Roboters real-time extension technology assures that the interrupts assigned to VxWorks always have higher priority than any event occurring in Windows XP including Windows XP interrupts.

KUKA Roboters VxWin technology does not utilize virtual machine technology or OS time slicing in any form. Whenever a real-time interrupt occurs, VxWorks runs exclusively until all real-time processing is complete. Conversely, Windows XP has the full resources of the CPU until a real-time interrupt occurs.



## VxWin State Machine



Real-time interrupts will interrupt either Windows XP or VxWorks tasks within a few microseconds. The interrupt signals the corresponding VxWorks interrupt service routine (ISR). After the ISR returns, VxWorks checks for any tasks ready to run. If a task or tasks are ready, VxWorks will activate them accordingly. Each task runs until it becomes suspended. After all tasks are suspended, VxWin transfers control back to Windows XP.

Under VxWin, it is not possible for Windows XP to run unless all VxWorks tasks are idle. However, a situation could exist where Windows XP must be allowed to run before all VxWorks tasks complete. Therefore, a VxWorks function is supplied within VxWin that forces the execution of the VxWorks idle task, effectively transferring execution back to Windows XP.

### Exception Handling of VxWorks

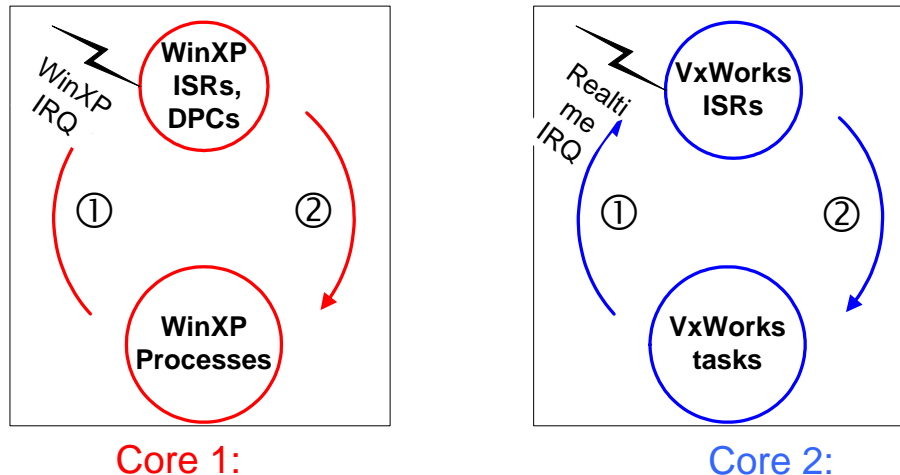
Whenever VxWorks is activated via the ISRs, a new exception table is loaded. This is accomplished by a single assembler instruction. Therefore, exception handling for page faults, debugging, and so on is exactly the same under VxWin as it is for standalone VxWorks. The Windows XP exception table is reloaded before falling back to Windows XP.

The arithmetic co-processor may be used within VxWin without any restrictions. Since it is possible that the interrupted Windows XP application is using the arithmetic co-processor, the co-processor state is saved if VxWorks requires its use and restored before returning to Windows XP.

## Dualcore and Multicore Systems: Exclusive Mode Operation

When VxWin operates in exclusive mode (only available on multiprocessor/multicore systems) the last CPU core in the system is dedicated to VxWorks. All remaining CPUs are used by Windows XP.

The following diagram, illustrates the flow of control on a dual core system:



### Operating states of VxWin in exclusive mode

- ① Exception-handling or a higher priority interrupt becomes outstanding.
- ② Interrupt Service Routine optionally starts a new task and then finishes.

Note: When running VxWin in exclusive mode Windows XP will never be interrupted. Application and interrupt processing run concurrently and independently on both CPU cores. There is no need in VxWorks to become idle.

## Hardware Requirements

- Minimum 200MHz Pentium® II processor
- Minimum 128MB RAM
- 16MB available hard disk space

## Operating System Requirements

- Microsoft Windows® XP SP2 <or>
- Microsoft Windows® XP Embedded SP2
- Wind River VxWorks® 5.4, 5.5 or 6.x

## **Licensing**

The VxWin product license from KUKA Roboter GmbH only consists of the coupling mechanisms allowing both operating systems to run concurrently on the same PC. It does not include the required licenses for Microsoft Windows XP and Wind River VxWorks. Those must be supplied by Microsoft and Wind River distributors. VxWin is installed only after licenses for both Windows XP and VxWorks have been received.

### *About KUKA Roboter GmbH*

*KUKA Roboter GmbH is the number one European industrial robot manufacturer and a part of KUKA AG Group. KUKA Roboter GmbH specializes in real-time software, operating systems, motion control, graphical user interfaces (GUI), handhelds, dynamic algorithms, PC-based hardware development and application programming. KUKA Roboter GmbH globally markets real-time extension technology for automation and robotics as well as test and measurement instrumentation, and medical devices and equipment. For more information on KUKA Roboter GmbH, contact Mr. Rainer Gallus, Sales and Marketing – Real-Time Products. Phone: +49 821 4533-3768 or RainerGallus@KUKA-Roboter.de. Online information can be viewed at [www.kuka-rtos.com](http://www.kuka-rtos.com).*

© 2008 KUKA Roboter GmbH all rights reserved

Information contained herein supersedes previously published specifications and is subject to change without notice. Trademarks are property of their respective owners.