



acontis technologies GmbH

SOFTWARE

VxWin

Getting Started Guide

Version 6.1

© Copyright **acontis technologies GmbH**

Neither this document nor excerpts therefrom may be reproduced, transmitted, or conveyed to third parties by any means whatever without the express permission of the publisher. At the time of publication, the functions described in this document and those implemented in the corresponding hardware and/or software were carefully verified; nonetheless, for technical reasons, it cannot be guaranteed that no discrepancies exist. This document will be regularly examined so that corrections can be made in subsequent editions. Note: Although a product may include undocumented features, such features are not considered to be part of the product, and their functionality is therefore not subject to any form of support or guarantee.

Table of Contents

Inhalt

1.	Introduction	4
1.1.	Documentation	4
2.	Getting Started with the System Manager	5
2.1.	Prerequisites	5
2.2.	The Software Development Kit (SDK)	5
2.3.	System Manager	6
3.	Setup the Real-time Extension Runtime.....	7
4.	Run vxWorks	9
5.	Deployment to a runtime PC	13
On the development PC	13	
On the target PC.....	14	
6.	Device Partitioning.....	15
6.1.	Real-time extension configuration (for ex. Network Card).....	15
6.1.1.	Identify your Network Controller	15
7.	Building the Real-time Extension Example Programs (Windows part)	17
1.	Build instructions for Windows XP / Windows 7	17
II.	Appendix.....	18
1.	Adding User settings in a System Manager configuration	18
2.	Technical Background of System Manager	18

1. Introduction

The System Manager for VxWorks is an add-on for the acontis VxWorks-based product line (**VxWin**) of real-time extension for Windows XP 32-Bit / Windows 7 / 8 / 10 32-Bit / 64-Bit.

The acontis **VxWin** enables VxWorks to share a single computer with a Microsoft Windows desktop operating system while maintaining VxWorks hard real-time determinism and performance. Both operating systems operate independent from each other, and, therefore, VxWorks operation is immune to desktop Windows exceptions and faults.

This document gives you an overview about the System Manager for VxWorks add-on. In this document a step by step walkthrough is given how to setup the VxWin real-time extension. In the last chapter we give brief instructions how the shipped SDK examples can be built with Visual Studio.

The real-time extension package comes with the following components:

- Real-time extension runtime
 - Kernel driver for virtual network adapter between Windows and VxWorks
 - Kernel driver to assign hardware resources to the real-time OS (VxWorks)
 - Virtual Machine Framework (Runtime shared by Windows and VxWorks)
 - Ready to Run “VxWorks” OS Image + SDK.
- SDK's for the real-time extension
 - Libraries and header files for developing your own C/C++ applications, using our RTOS framework. E.g. you can develop applications which can communicate between Windows and VxWorks through shared memory and signals.
 - Extensible demo applications and examples (under `SDK/examples/`)
- Tools and Utilities
 - System Manager. This is the primary configuration tool for our real-time extension product family.

1.1. Documentation

Detailed information about the real-time extension (VxWin) can be found at the [VxWin.pdf](#) manual.

The VxWin API is described in the [RTOS-Virtual-Machine-ReferenceManual.pdf](#) document.

2. Getting Started with the System Manager

This manual will guide you through the following steps:

1. Short **overview** over System Manager
2. **Setting up** a real-time configuration for VxWorks
3. **Hardware / Software setup** for a valid configuration
4. **Deployment** of the real-time configuration

2.1. Prerequisites

Please make sure that the following conditions are met:

- Installed .NET Framework Version 4.0 (Client Profile) (For running the System Manager)
- Installed Microsoft Visual Studio 2005 with latest service pack.
- Installed VxWin (requires at least .NET Framework Version 4.0 Client Profile).
- Intel PRO/1000 gigabit network controller hardware installed on the target device. (optional)
- For **Windows 7 64-Bit** edition only: At least two (logically) CPU cores.

2.2. The Software Development Kit (SDK)

The software development kit, located in SDK (below the installation directory), contains libraries and header files to develop software on top of the real-time extension (VxWin based).

2.3. System Manager

The System Manager is used for configuration of the real-time part of real-time extension as well as for managing the whole development process. It provides functionality to manage the memory configuration, the CPU assignment to the RTOS and the assignment of PCI devices. Furthermore it provides functionality to deploy applications to the real-time part and create the Visual Studio projects for the example programs.

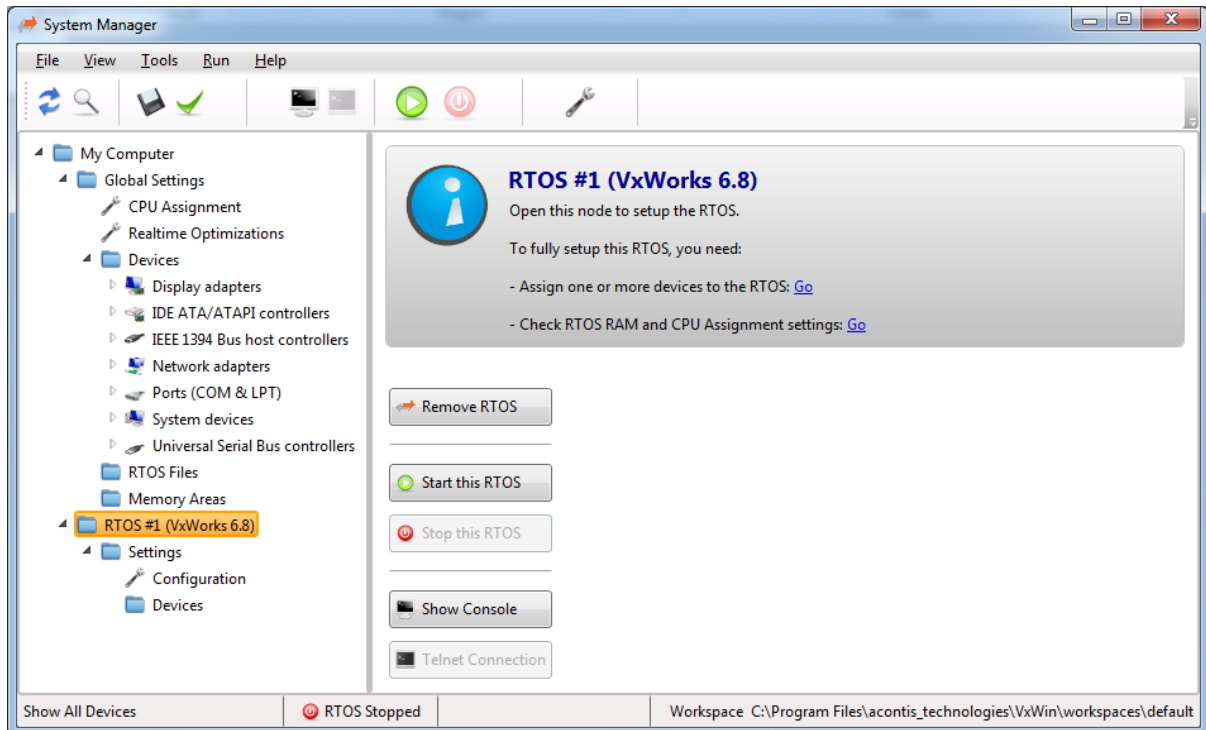


Figure 2-1 (System Manager)

3. Setup the Real-time Extension Runtime

This chapter describes how to setup the basic parameters of the real-time extension via the supplied System Manager tool. First step is to choose a place for your workspace. Selecting the default place is recommended.

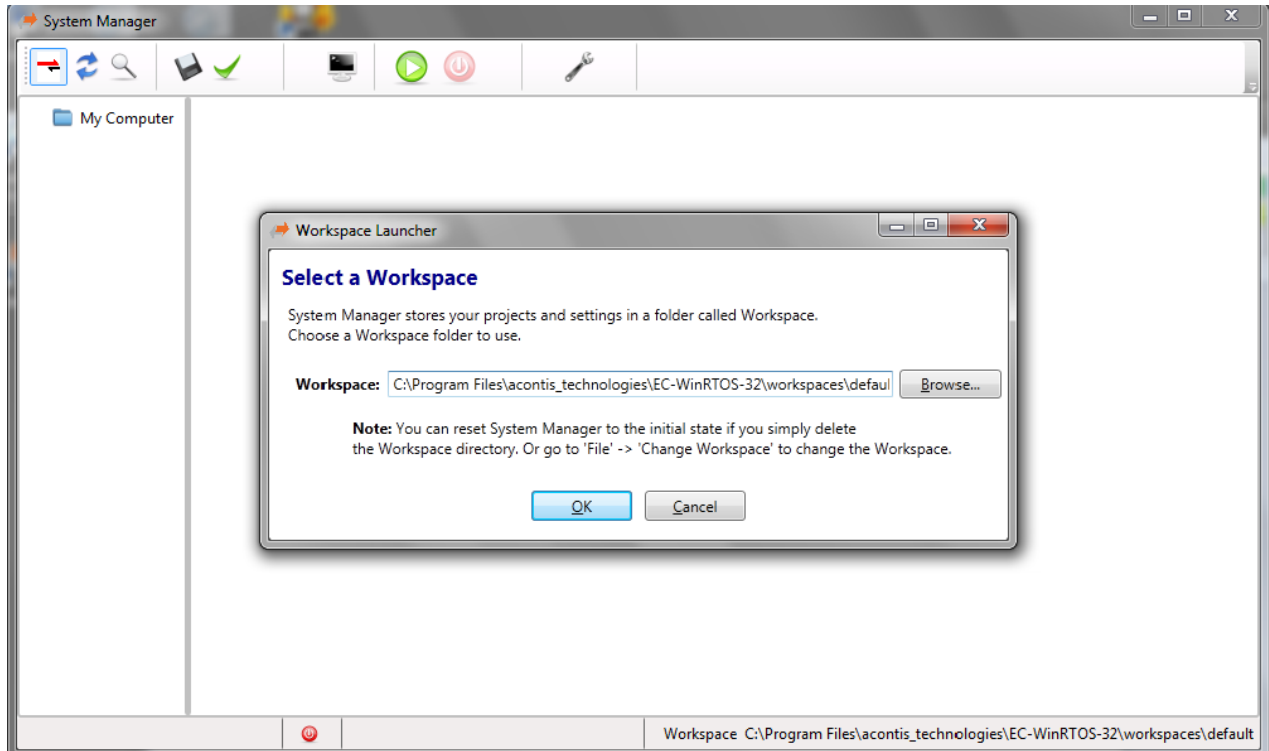


Figure 3-1 (Add Workspace)

Next step is to “Add RTOS”. In this example it’s “VxWorks 5.x or 6.x. After adding RTOS it’s time to go to the “CPU Assignment”

If started first and Windows 7 32BIT is used all available CPUs are assigned to Windows 7 and the first CPU is additionally assigned to VxWorks. **(Figure 3-2)** This CPU Assignment may be valid, however it isn’t recommended. To achieve a recommended “CPU Assignment” assign all CPU’s to Windows, except the last one. The last CPU has to be assigned to “VxWorks”. **(Figure 3-3)**

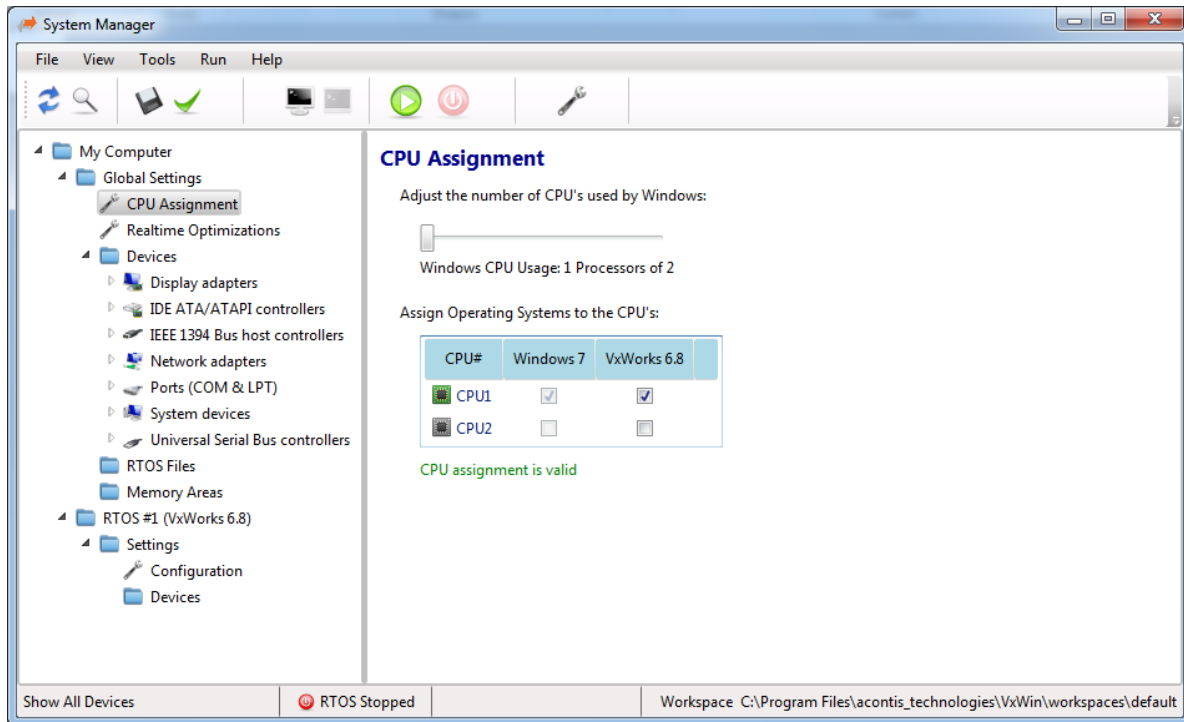


Figure 3-2 (CPU Assignment)

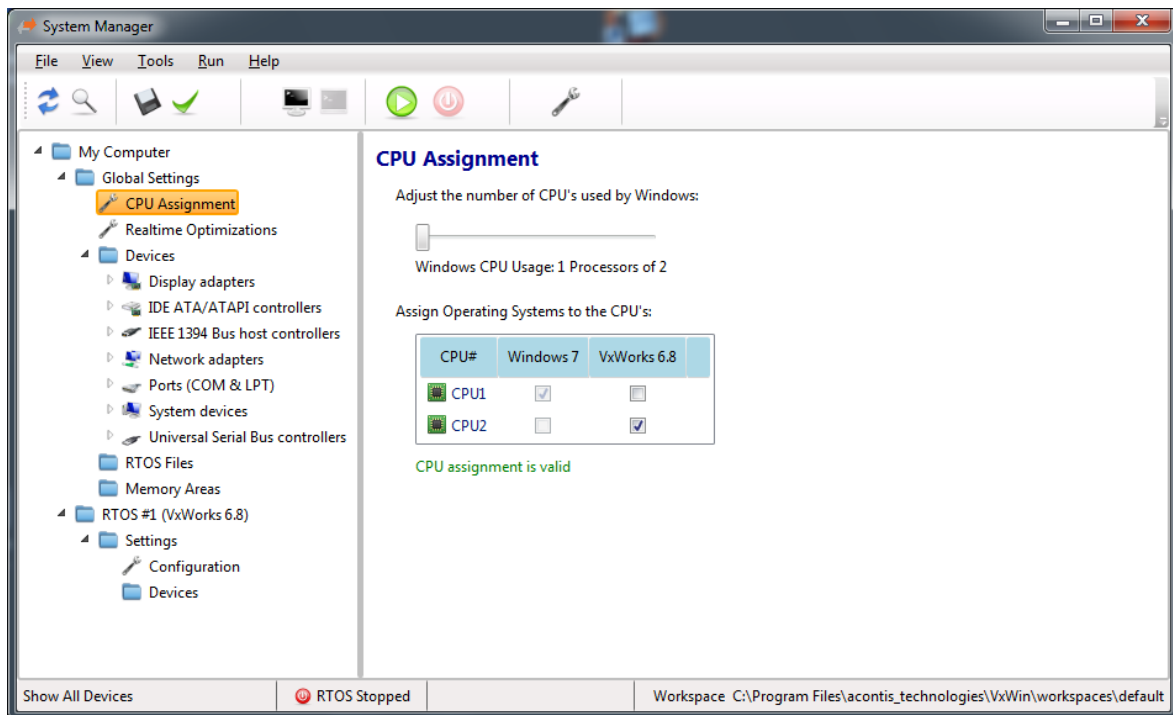


Figure 3-3 (CPU Assignment)

If Windows 7 64Bit is used, the real-time part **must** use at least one CPU exclusively. To achieve this use the “number of processor slider” above the CPU table in such a way that the real-time part has at least one CPU exclusively. After doing that the CPU assignment should be valid.

→ *It's important that the message beyond the CPU table is green and the CPU assignment is valid!*

The System Manager will show a reboot message box, if some changes require a reboot after *save settings* has been pushed.

4. Run vxWorks

Several vxWorks OS images are delivered as pre-compiled binaries, which are ready-to-use. To measure the target PC for real-time capability a RealtimeDemo.bin is supplied. As a starting point a valid System Manager configuration is needed → see previous chapter 3 Setup the Real-time Extension Runtime.

The next step requires starting the “Target Console Window”. This isn’t mandatory, but without the console output nothing of the VxWin output will be shown to the user. To start the Target Console Window the monitor button must be pushed as shown in the following screenshot:

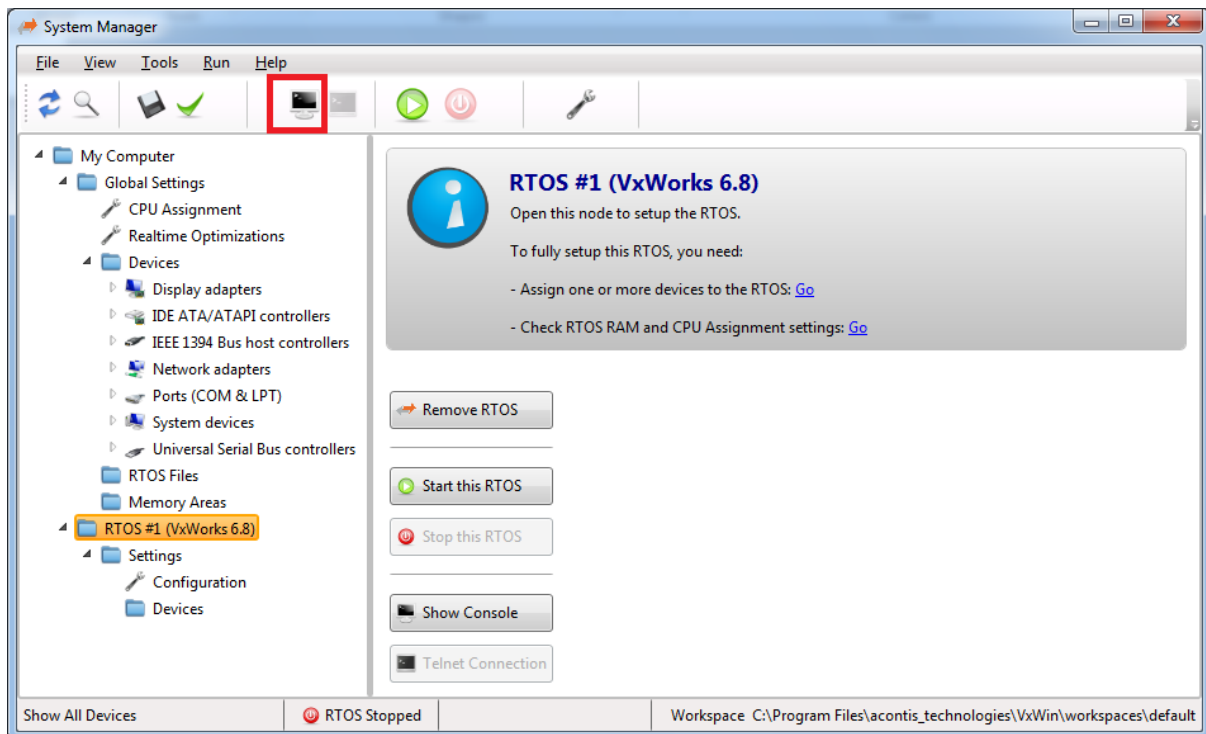


Figure 4-1

The following window appears:

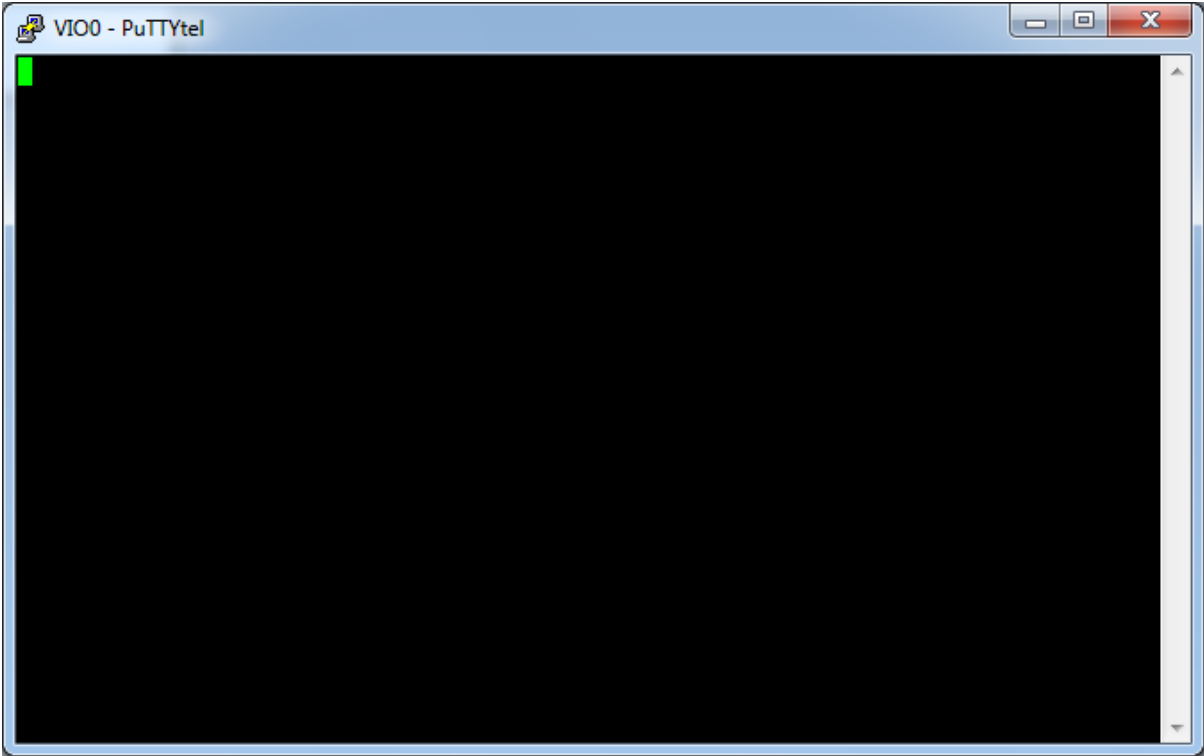


Figure 4-2

The system is now ready to start the desired vxWorks image. To do so, please push the “green” arrow button as shown below:

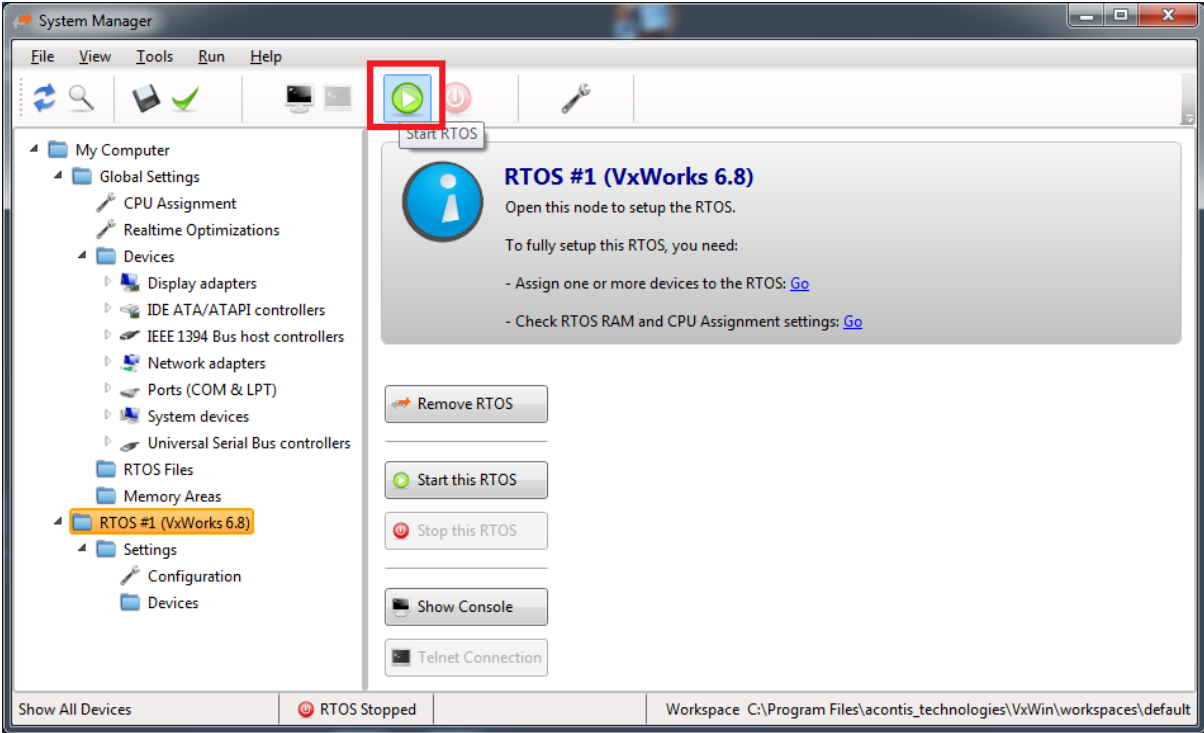


Figure 4-3

After pushing the button, the following message box appears:

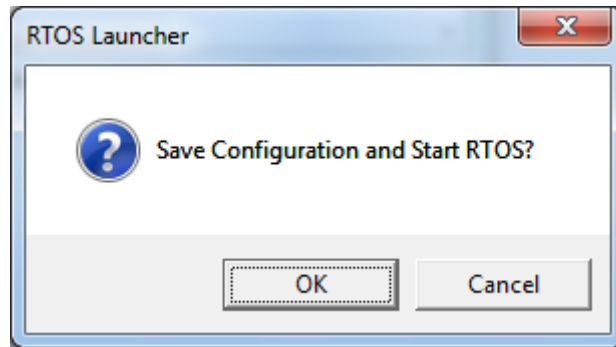


Figure 4-4

After successful starting of the VxWorks presents the following screenshot indicators for a successful start of VxWorks:

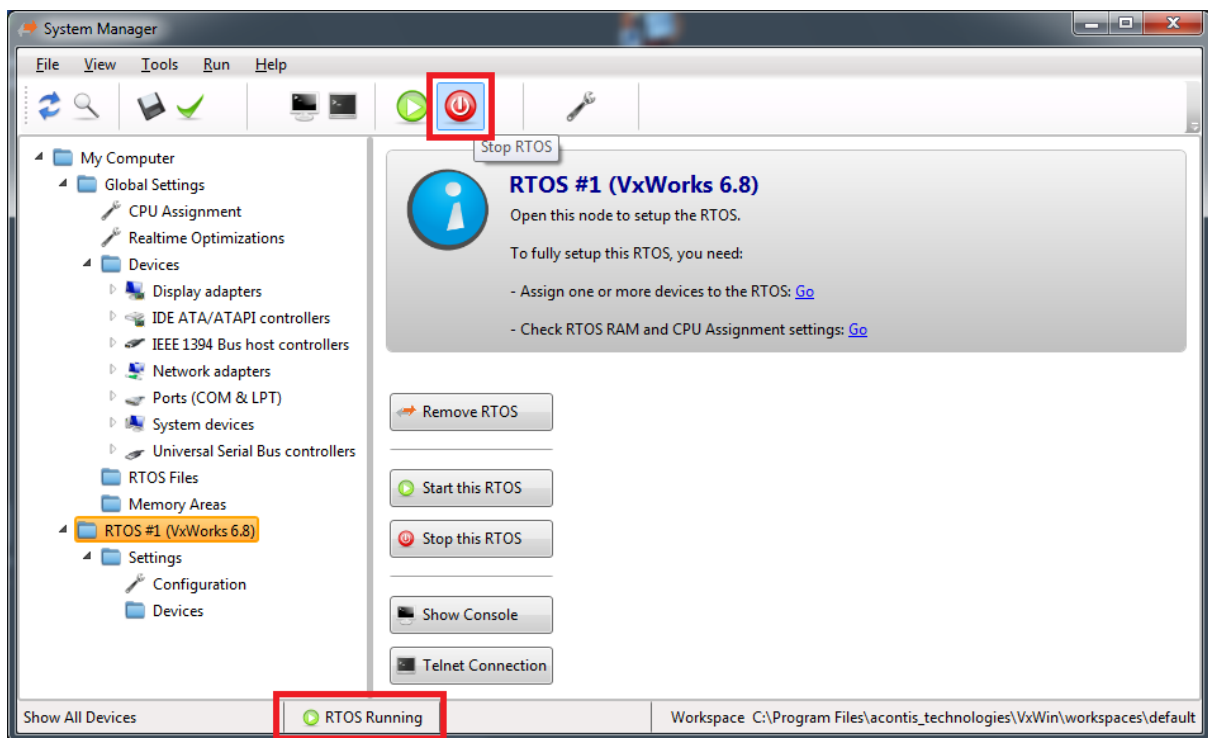


Figure 4-5

A RTOS Running will show in the status bar and the Stop Rtos button (red) will be active. The console output should be like the following:

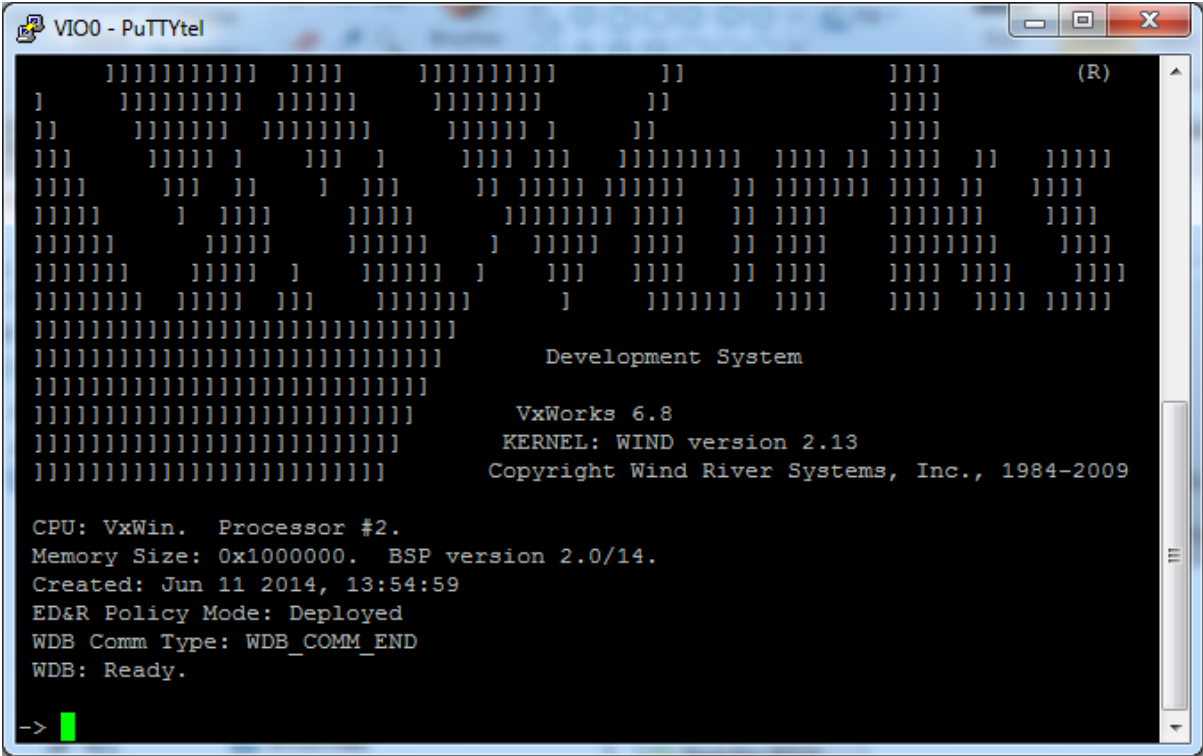


Figure 4-6

5. Deployment to a runtime PC

To deploy the settings, configurations from the development PC to a target runtime PC, there are some preparations on development and target PC needed. Please look at the following subchapter.

On the development PC

- Go to menu **Run|Create Launch Shortcuts** and press <enter>.

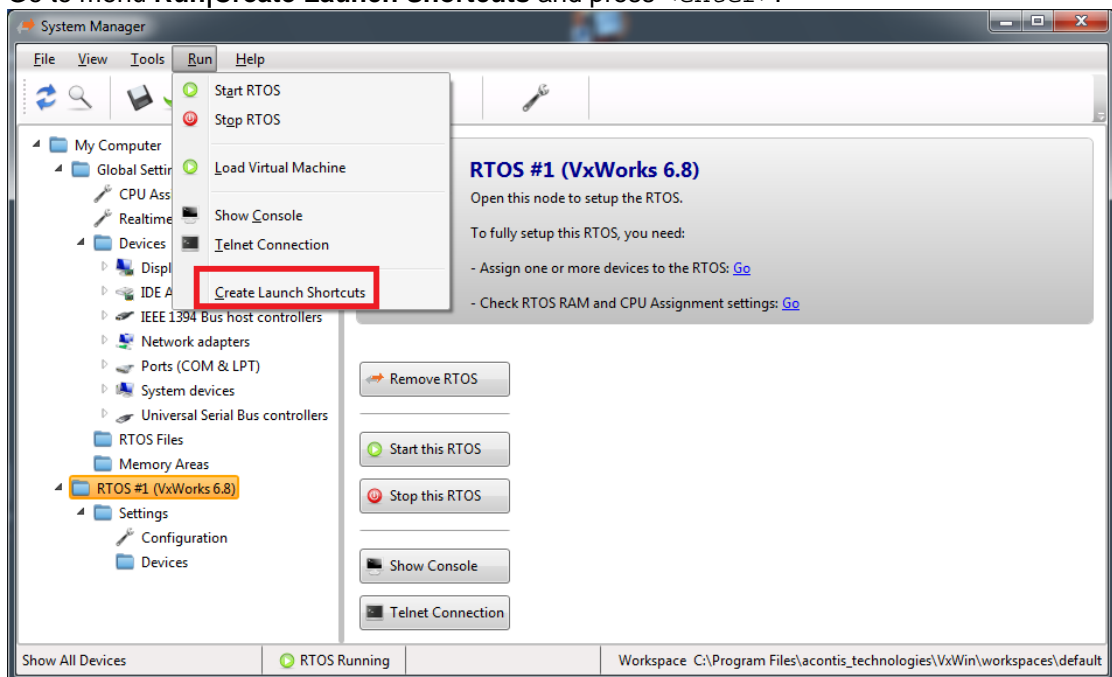


Figure 5-1

- Needed batch files and links will be created into the workspaces directory.

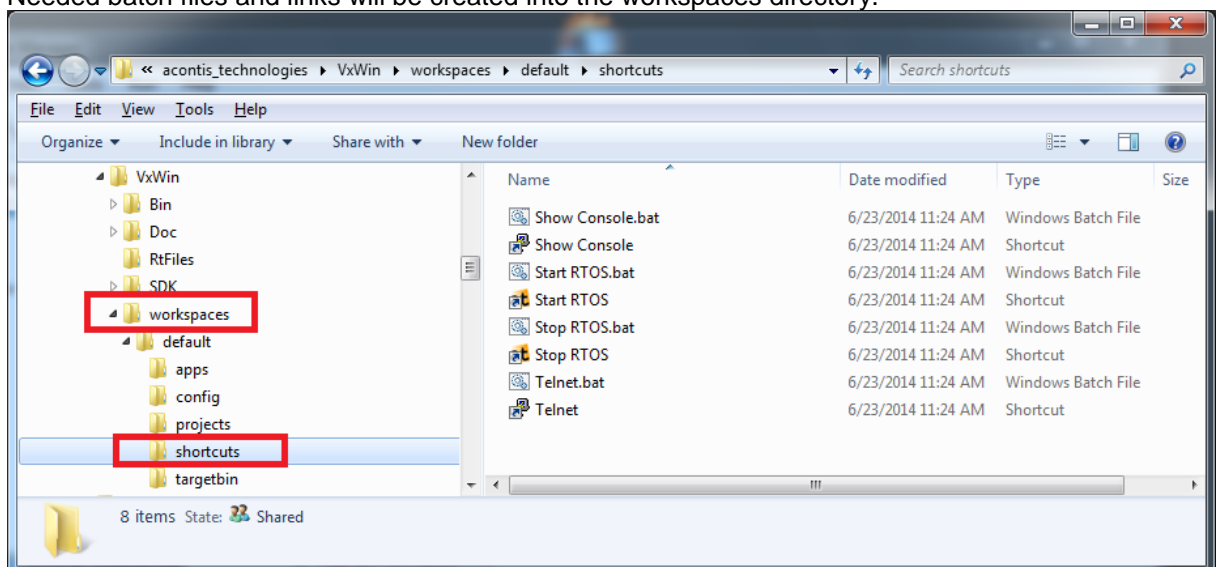


Figure 5-2

- Copy **whole** workspace directory to a location, from which the target PC has access.
→ Maybe compress whole workspace directory to a zip file. (Optional)

On the target PC

- Install VxWin on a target runtime PC
- Copy (decompress) workspace directory to %RTE_ROOT% directory.
- *Optional:* move the shortcuts from %RTE_ROOT%\workspace\default\shortcuts\ to the desktop (if desired).
- To automatically start the real-time extension: add an entry to the autostart with a call of %RTE_ROOT%\workspace\default\shortcuts\Start RTOS.bat.

6. Device Partitioning

If a device should be used exclusively by the real-time OS (RTOS) the desired device must be assigned to the RTOS by the System Manager. Reassignment of hardware in the real-time extension is a two-step process: Isolate the device interrupts and install the RtosPnp device driver. See also the RTOS VM User Manual for more information about real-time device management. Most of the work will be done by the System Manager.

In the following chapter a network card will be assigned to the RTOS for example.

6.1. Real-time extension configuration (for ex. Network Card)

6.1.1. Identify your Network Controller

The System Manager lists an overview over all available/possible network controller and other devices in the current system.

Please apply the following to identify a suitable/compatible network controller:

1. Open the System Manager.
2. Expand the **Global Settings**.
3. Expand the **Devices**.
4. Select the desired network adapter (if a compatible network adapter was found!)
5. Right click of the desired network adapter and select *Assign to RTOS (Interrupt disabled)*.
6. Please see Figure 6-1.

Hint: The default is to show all devices of this PC (graphics adapter card, HDD, etc.)

Assign with Interrupts Disabled:

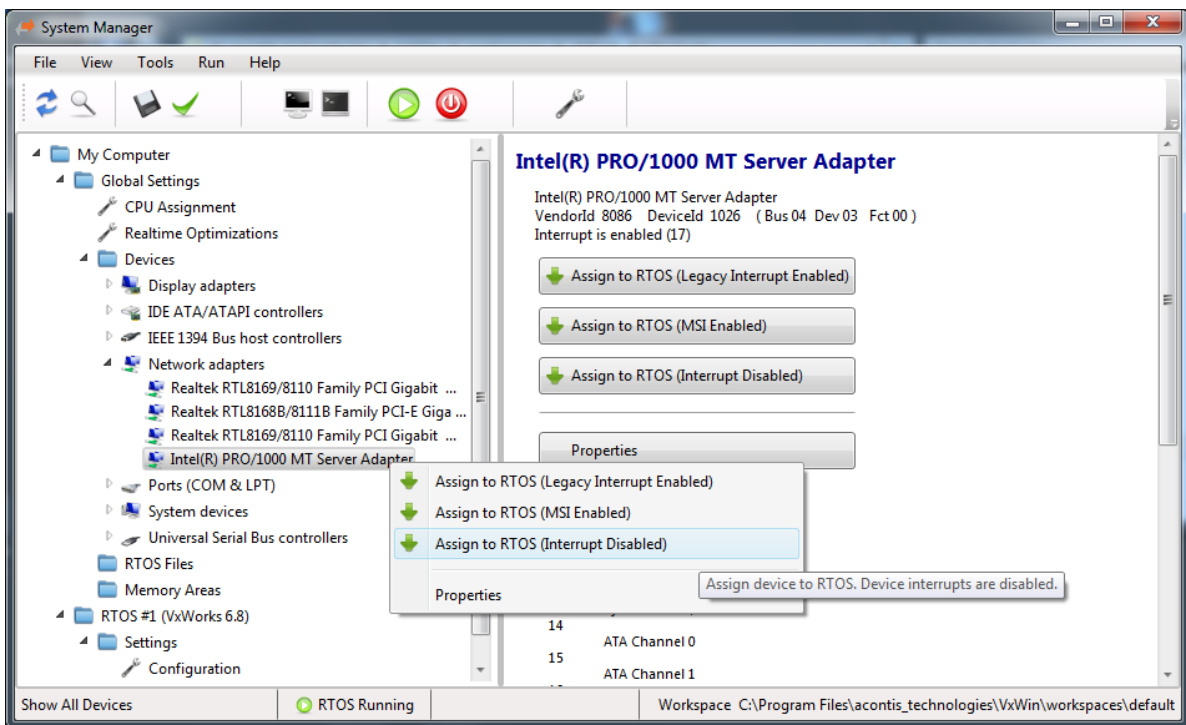


Figure 6-1 (System Manager)

Successfully assigned RtosPnp Device Driver will be listed under the Realtime OS Devices tree node under the node "Device Configuration".

WARNING: DURING THE INSTALLATION OF THE RTOS DRIVER OF THE SELECTED NETWORK CARD, WINDOWS COULD THROW SEVERAL POP-UP WINDOWS. ON WINDOWS XP THE USER MUST IGNORE THESE POP-UP WINDOWS! THEY WILL DISAPPEAR AUTOMATICALLY!

WARNING: DURING INSTALLATION OF THE RTOS DRIVER OF THE SELECTED NETWORK CARD A REBOOT COULD BE NEEDED. IN THAT CASE START THE SYSTEM MANAGER AFTER THE REBOOT AGAIN AND CHECK IF THE DESIRED NETWORK CARD IS ASSIGNED CORRECTLY TO THE RTOS.

7. Building the Real-time Extension Example Programs (Windows part)

1. Build instructions for Windows XP / Windows 7

This section offers building instructions for the shipped example programs in **SDK\Examples\Windows** (below the real-time extension installation directory).

Each directory is an own example program Please look in the appropriate folders under subdirectory **/SDK/Examples/*** of the installation directory.

- Open Visual Studio 2005
- Create a new "Win32 Console Application", select "Console application", "Empty project"
- Add all *.cpp files located in **SDK\Examples\Windows\<Demo-Name>** below the real-time extension installation directory to the project.
- Go to Project "Properties" -> "C/C++" -> "General" -> "Additional Include Directories"
Add **SDK\inc** and **SDK\inc\Windows** (below the real-time extension installation directory) as additional include directories.
- Go to Project "Properties" -> "Linker" -> "General" -> "Additional Library Directories"
Add **SDK\LIB\Windows\X86** (below the real-time extension installation directory)
- Go to Project "Properties" -> "Linker" -> "Input" -> "Additional Dependencies" -> "..."
Add **"RtosLib32.lib"**
- Go to Project "Properties" -> "C/C++" -> "Language" -> "Treat wchar_t as Built-in Type" select "No"
- Go to Project "Properties" -> "C/C++" -> "Preprocessor"
Add **RTOS32WIN** and **_CRT_SECURE_NO_DEPRECATED** to section "Preprocessor Definitions".
- Build the Example.
- Run the Example.

*Note that for **EventDemo**, **InterlockDemo** and **ShmDemo** a counterpart executable on the **VxWorks** side must be started (identically named; see source code below **SDK\Examples\VxWorks**) for proper operation!*

II. Appendix

1. Adding User settings in a System Manager configuration

In a System Manager controlled environment, all user settings shall be stored in the file `systemmgr_user.config` which is located below the workspace's config sub directory.

2. Technical Background of System Manager

There are some technical notes, which the user should know about the System Manager.

The most important fact in this direction is the use of a workspace directory. All modified, added, etc files, data, etc are located in the workspace. At first start of the System Manager the workspace directory will be created (default: `%RTE_ROOT%\Workspace` or user-defined location) and all configuration files, etc will be copied therein. Thus the System Manager operates **only** on copied files and will **never** change the original files. *As a side effect of this, after deleting of the workspace directory, the start of System Manager will be like the first step before.*

A further important notice is the fact, that the System Manager uses the newest available Visual Studio, which is installed at the development PC. This means, if ex. Visual Studio 2005 and Visual Studio 2012 are installed, Visual Studio 2012 will be preferred and automatically selected. To change the preferred Visual Studio, a manual change in `usersettings.xml` is needed. This `usersettings.xml` will be found at `%RTE_ROOT%\Bin\Windows\x64\SystemManagerV2` or `%RTE_ROOT%\Bin\Windows\x86\SystemManagerV2` on a 32bit Windows system.