SCALEXIO Hypervisor Extension

Guide

Release 2021-A – May 2021

dSPACE

How to Contact dSPACE

Mail:	dSPACE GmbH Rathenaustraße 26 33102 Paderborn Germany
Tel.:	+49 5251 1638-0
Fax:	+49 5251 16198-0
E-mail:	info@dspace.de
Web:	http://www.dspace.com

How to Contact dSPACE Support

If you encounter a problem when using dSPACE products, contact your local dSPACE representative:

- Local dSPACE companies and distributors: http://www.dspace.com/go/locations
- For countries not listed, contact dSPACE GmbH in Paderborn, Germany.
  Tel.: +49 5251 1638-941 or e-mail: support@dspace.de

You can also use the support request form: http://www.dspace.com/go/supportrequest. If you are logged on to mydSPACE, you are automatically identified and do not need to add your contact details manually.

If possible, always provide the relevant dSPACE License ID or the serial number of the CmContainer in your support request.

Software Updates and Patches

dSPACE strongly recommends that you download and install the most recent patches for your current dSPACE installation. Visit http://www.dspace.com/go/patches for software updates and patches.

Important Notice

This publication contains proprietary information that is protected by copyright. All rights are reserved. The publication may be printed for personal or internal use provided all the proprietary markings are retained on all printed copies. In all other cases, the publication must not be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of dSPACE GmbH.

© 2020 - 2021 by:
dSPACE GmbH
Rathenaustraße 26
33102 Paderborn
Germany

This publication and the contents hereof are subject to change without notice.

AUTERA, ConfigurationDesk, ControlDesk, MicroAutoBox, MicroLabBox, SCALEXIO, SIMPHERA, SYNECT, SystemDesk, TargetLink and VEOS are registered trademarks of dSPACE GmbH in the United States or other countries, or both. Other brand names or product names are trademarks or registered trademarks of their respective companies or organizations.

Contents

About This Guide 7

Safety Precautions 11

General Warning 11
Safety Precautions for Working with the SCALEXIO Hypervisor Extension 12

Introduction to the SCALEXIO Hypervisor Extension 13

Overview of the SCALEXIO Hypervisor Extension 13
Features of the SCALEXIO Hypervisor Extension 16
Requirements of the SCALEXIO Hypervisor Extension 17
Introduction to Real-Time Linux 19

Accessing a SCALEXIO System with Hypervisor Extension 21

How to Access the Hypervisor Extension File System by Windows File Explorer 22
How to Access the SCALEXIO Web Interface and Virtual Machines by an Internet Browser 23
How to Access a Virtual Machine on the SCALEXIO System via SSH 24
How to Access a Virtual Machine on the SCALEXIO System via VNC 26
The Hypervisor Page of the SCALEXIO Web Interface 28

Installing the SCALEXIO Hypervisor Extension 31

Installation Workflow 31
How to Install the SCALEXIO Hypervisor Extension Boot Medium 32
How to Create a Virtual Machine 36
How to Install a Linux Distribution on a Virtual Machine 37
How to Add an SSH Server to a Linux Distribution 39
How to Add Hypervisor Tools to a Linux Distribution 40
How to Add NFS share to a Linux Distribution 42

Starting a SCALEXIO System with Hypervisor Extension 45

How to Power Up and Boot a SCALEXIO System with Hypervisor Extension ... 45
How to Start a Virtual Machine ... 47
Booting a Virtual Machine ... 48

Shutting Down a SCALEXIO System with Hypervisor Extension 51

Workflow of the System Shutdown Procedure ... 52
How to Shut Down a Virtual Machine ... 52
How to Switch Off a SCALEXIO System with Hypervisor Extension ... 54

Integrating Linux Custom Software Components 57

Basics on Integrating Linux Custom Software Components ... 57
How to Integrate Hypervisor API Functions Into a Custom Linux Application ... 58
How to Integrate Hypervisor API Functions Into a SCALEXIO Real-Time Application ... 59

Communication Between Operating Systems 63

Communication via Virtual Ethernet Connection ... 63
Communication via Shared Memory ... 65
Communication via Interrupts ... 67
Communication via a Shared File System ... 68

Configuring the Hypervisor and Virtual Machines 69

How to Edit a Configuration File ... 69
Syntax of Configuration Files ... 70
Configuring a Virtual Machine ... 72
Configuring the Hypervisor ... 80

Backing Up and Restoring Virtual Machines 83

Procedures for Backing Up and Restoring a Virtual Machine ... 83

Modifying or Replacing a Linux Distribution 85

Procedures for Modifying or Replacing a Linux Distribution ... 85

Appendix 87

SCALEXIO System Connectors for External Hardware..... 87
SCALEXIO System Mains Controls..... 92
Virtual Machine Configuration File..... 93
Hypervisor Configuration File..... 96

Index 97

About This Guide

Content

The SCALEXIO Hypervisor Extension is based on a kernel-based virtual machine (KVM). It operates the SCALEXIO real-time operating system and virtual machines with any Linux distribution simultaneously. This allows for integrating Linux real-time and non-real-time applications into the environment of SCALEXIO processing hardware.

The guide provides the following information:

- Installing and configuring the SCALEXIO Hypervisor Extension
- Working with the SCALEXIO Hypervisor Extension
- Integrating custom software components
- Methods of communication between operating systems
- Modifying the system configuration

Required knowledge

To work with a SCALEXIO system with Hypervisor Extension, you must be familiar with the following:

- SCALEXIO hardware, especially processing hardware configuration
- Basic Ethernet and network communication
- Real-time operating systems, especially Linux
- Working with and managing Linux distributions
- C programming language

Symbols

dSPACE user documentation uses the following symbols:

Symbol	Description
DANGER	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
WARNING	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
CAUTION	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.

Symbol	Description
NOTICE	Indicates a hazard that, if not avoided, could result in property damage.
Note	Indicates important information that you should take into account to avoid malfunctions.
Tip	Indicates tips that can make your work easier.
	Indicates a link that refers to a definition in the glossary, which you can find at the end of the document unless stated otherwise.
	Precedes the document title in a link that refers to another document.

Naming conventions

dSPACE user documentation uses the following naming conventions:

%name% Names enclosed in percent signs refer to environment variables for file and path names.

< > Angle brackets contain wildcard characters or placeholders for variable file and path names, etc.

Special folders

Some software products use the following special folders:

Common Program Data folder A standard folder for application-specific configuration data that is used by all users.

```
%PROGRAMDATA%\dSPACE\<InstallationGUID>\<ProductName>
```

or

```
%PROGRAMDATA%\dSPACE\<ProductName>\<VersionNumber>
```

Documents folder A standard folder for user-specific documents.

```
%USERPROFILE%\Documents\dSPACE\<ProductName>\
<VersionNumber>
```

Local Program Data folder A standard folder for application-specific configuration data that is used by the current, non-roaming user.

```
%USERPROFILE%\AppData\Local\dSPACE\<InstallationGUID>\
<ProductName>
```

Accessing dSPACE Help and PDF Files

After you install and decrypt dSPACE software, the documentation for the installed products is available in dSPACE Help and as PDF files.

dSPACE Help (local) You can open your local installation of dSPACE Help:

- On its home page via Windows Start Menu
- On specific content using context-sensitive help via **F1**

dSPACE Help (Web) You can access the Web version of dSPACE Help at www.dspace.com/go/help.

To access the Web version, you must have a *mydSPACE* account.

PDF files You can access PDF files via the icon in dSPACE Help. The PDF opens on the first page.

Safety Precautions

Where to go from here

Information in this section

General Warning 11

Safety Precautions for Working with the SCALEXIO Hypervisor Extension..... 12

General Warning

Danger potential

Using dSPACE software can be dangerous. You must observe the following safety instructions and the relevant instructions in the user documentation.

Improper or negligent use can result in serious personal injury and/or property damage.

Using the SCALEXIO Hypervisor Extension can have a direct effect on dSPACE systems and technical (electrical, hydraulic, mechanical) systems connected to them.

- Only persons who are qualified to use dSPACE software, and who have been informed of the above dangers and possible consequences, are permitted to use it.
- The user must examine all applications where malfunctions or operating errors involve the danger of injury or death must be examined for potential hazards. If necessary, the user must take additional measures for protection (for example, an emergency off switch).

Liability

It is your responsibility to adhere to instructions and warnings. Any unskilled operation or other improper use of this product in violation of the respective safety instructions, warnings, or other instructions contained in the user

documentation constitutes contributory negligence, which may lead to a limitation of liability by dSPACE GmbH, its representatives, agents and regional dSPACE companies, to the point of total exclusion, as the case may be. Any exclusion or limitation of liability according to other applicable regulations, individual agreements, and applicable general terms and conditions remain unaffected.

Safety Precautions for Working with the SCALEXIO Hypervisor Extension

Installing the hypervisor boot medium

Do not insert the boot medium, for example, a CompactFlash card, when the SCALEXIO Processing Unit is running. This might damage the boot medium.

- Ensure that the SCALEXIO Processing Unit is powered down before you insert the boot medium.

SCALEXIO system shutdown

Do not simply switch off a SCALEXIO system if a Hypervisor Extension is installed. Incorrectly shutting down a SCALEXIO system with Hypervisor Extension might result in data loss and rendering the files of virtual machines unusable.

- Shut down a SCALEXIO system properly. Refer to Shutting Down a SCALEXIO System with Hypervisor Extension on page 51.

Delayed process step execution

It is possible to run applications you created before starting to work with the SCALEXIO Hypervisor Extension. These are applications you were able to use without issues on a regular SCALEXIO system with only the SCALEXIO real-time operating system.

However, with the SCALEXIO Hypervisor Extension, executing real-time applications slows down and might damage connected external devices, especially ECUs and mechanical actuators.

- Ensure that existing real-time applications cannot damage connected external devices when they are executed under the SCALEXIO Hypervisor Extension.

Downloading real-time applications when shared memory is used

When shared memory is used for communication between the SCALEXIO real-time operating system and a virtual machine, downloading real-time applications when the virtual machine is shut down might result in erratic system behavior.

- First start the virtual machines, then download SCALEXIO real-time applications.

Storing user data on the boot medium

With the SCALEXIO Hypervisor Extension, the boot medium can be used to store user data.

- Regularly back up the user data from the boot medium to minimize the risks of data loss. Refer to Backing Up and Restoring Virtual Machines on page 83.

Introduction to the SCALEXIO Hypervisor Extension

Where to go from here

Information in this section

Overview of the SCALEXIO Hypervisor Extension..... 13
Features of the SCALEXIO Hypervisor Extension..... 16
Requirements of the SCALEXIO Hypervisor Extension..... 17
Introduction to Real-Time Linux..... 19

Overview of the SCALEXIO Hypervisor Extension

Use case of the SCALEXIO Hypervisor Extension

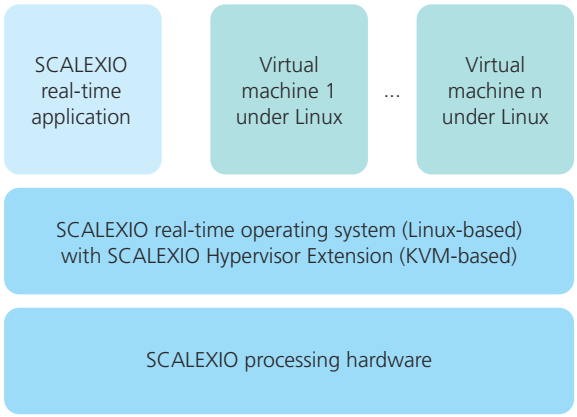
The SCALEXIO Hypervisor Extension lets you integrate Linux software components, for example, modeling and simulation software, into the SCALEXIO real-time application environment.

SCALEXIO processing hardware with multicore processors and the Linux-based SCALEXIO real-time operating system allow for parallel execution of real-time applications.

If the SCALEXIO Hypervisor Extension is installed, the SCALEXIO real-time operating system can host one or more virtual machines running under real-time-capable Linux. The simultaneous execution of the SCALEXIO real-time operating system and the virtual machines enables parallel and synchronized execution of SCALEXIO real-time applications and Linux real-time applications as well as non-real-time applications. Therefore, the hypervisor lets you almost freely manage computing power by assigning CPU cores to the operating systems as required by the applications.

Working principle of the hypervisor

The following illustration shows how hardware and software components of a SCALEXIO system with Hypervisor Extension are arranged. The hypervisor is part of the SCALEXIO real-time operating system.



The hypervisor is KVM-based and adds almost no latencies to execution times. Thus, it preserves the hard real-time properties of the SCALEXIO real-time operating system. The virtual machines run independently and are isolated by means of the Intel VTx technology. This isolation ensures the undisturbed execution of SCALEXIO real-time applications. The hypervisor provides a low latency and high bandwidth virtual network interface. Additionally, it provides means to transmit and receive interrupts and data. This can be used, for example, to synchronize the execution of tasks that are distributed across the real-time operating system and virtual machines.

Obtaining the SCALEXIO Hypervisor Extension

The SCALEXIO Firmware that includes the Hypervisor Extension is provided on a boot medium, such as a CompactFlash card. The boot medium with the Hypervisor Extension replaces the original boot medium and serves as new boot, installation, and storage medium of the SCALEXIO processing hardware. All the files of the Hypervisor Extension are located on the boot medium if not specified differently in the configuration file.

Compatibility with earlier dSPACE Releases

As of Release 2020-B, the SCALEXIO Hypervisor Extension uses the Linux-based operating system.

Note

If you already use the SCALEXIO Hypervisor Extension in your projects with dSPACE Release 2020-A or earlier, you cannot use the files of these Releases with dSPACE Release 2020-B.

You cannot use any real-time applications, model containers (i.e., BSC, FMU, SIC, V-ECU), S-functions, or custom I/O functions that contain binary libraries built for dSPACE Release 2020-A and earlier on this operating system. You must rebuild these files from source code with dSPACE Release 2020-B. Source code is expected to be reusable in most cases. Cases that might require an adaption of C code are related to custom code that uses special features, e.g., OS-specific functions, created by your company or third-party suppliers

If a migration to dSPACE Release 2020-B is required, contact dSPACE Support (www.dspace.com/go/supportrequest).

Installing the SCALEXIO Hypervisor Extension in existing SCALEXIO systems

Note

Installing the SCALEXIO Hypervisor Extension in an existing SCALEXIO system might change the assignment of connectors for external hardware.

Compare the connector assignment of the existing SCALEXIO system with the default assignment of the SCALEXIO Hypervisor Extension. Refer to SCALEXIO System Connectors for External Hardware on page 87.

If the SCALEXIO Hypervisor Extension has to be reconfigured, contact dSPACE Support (www.dspace.com/go/supportrequest).

Impact on the behavior of real-time applications

NOTICE

Executing real-time applications slows down with the SCALEXIO Hypervisor Extension.

Risk of damage of connected external devices (especially ECUs and mechanical actuators).

- Ensure that using the SCALEXIO Hypervisor Extension cannot result in damaging connected external devices.

Even though the two operating systems are isolated from each other by the hypervisor and the hypervisor itself is guaranteed not to cause any additional latencies, when both operating systems are running in **privileged mode**, the execution times of real-time applications might vary compared to those on SCALEXIO processing hardware running with regular SCALEXIO Firmware.

This is due to the fact that certain resources that influence the execution times of a real-time application are still shared between the two operating systems, for example the processor's level 3 cache. Also, access to the boot medium is shared between the two operating systems and, therefore, read/write operations from/to the medium are slower compared to regular SCALEXIO processing hardware.

Related topics

Basics

Features of the SCALEXIO Hypervisor Extension..... 16
Introduction to Real-Time Linux..... 19
Requirements of the SCALEXIO Hypervisor Extension..... 17

Features of the SCALEXIO Hypervisor Extension

Parallel execution of applications

Applications can be executed in parallel and synchronized:

- Under the SCALEXIO real-time operating system: Real-time applications generated with the dSPACE tool chain.
- On virtual machines under Linux: Real-time or non-real-time applications or services.

Supported Linux distributions

The SCALEXIO Hypervisor Extension supports various Linux distributions, such as Ubuntu, Debian, CentOS 7, etc. on virtual machines. You can install your preferred operating system. However, some features require support inside the Linux kernel of your distribution:

- For using the virtual Ethernet interface, your Linux kernel must support the virtio_net para-virtualization drivers. This is enabled via `CONFIG_VIRTIO_NET` inside the kernel. Most Linux distributions have this feature enabled.
- The shared memory interface requires a special Linux kernel module that is shipped with this extension and must be built against the Linux kernel you use. Build and installation is only supported by Ubuntu 18.04 using the dSPACE hypervisor tools package.
- KVM can run guest virtual machines fully virtualized. However, performance is best when your Linux kernel supports to be run as a KVM guest. This is provided when the kernel has the `CONFIG_KVM_GUEST` config options set.

Tip

The SCALEXIO Hypervisor Extension is tested with Ubuntu 18.04, which is therefore the recommended Linux distribution.

Operating system communication

Low latency, high bandwidth data transfer between operating systems is enabled by the following methods:

- Virtual Ethernet connections
- Shared memories
- Software interrupts
- Shared files

Support of core isolation

Virtual machines use their own isolated cores. For more information on core isolation, refer to Introduction to Real-Time Linux on page 19.

Flexible resource distribution

The following resources can be assigned to operating systems:

- CPU cores
- System memory
- Ethernet channels
- PCI devices
- USB controllers

Related topics

Basics

Introduction to Real-Time Linux..... 19
Overview of the SCALEXIO Hypervisor Extension..... 13
Requirements of the SCALEXIO Hypervisor Extension..... 17

Requirements of the SCALEXIO Hypervisor Extension

Required components for operating the SCALEXIO Hypervisor Extension

Hardware The Hypervisor Extension requires the following hardware components:

- One of the following SCALEXIO processing hardware:

Processing Hardware	Boot Medium
SCALEXIO Processing Unit with E5-2640v3 processor	CompactFlash card
SCALEXIO Processing Unit with E3-1275v6 processor[1)]	
SCALEXIO Processing Unit with Intel 6208U processor	SSD

Processing Hardware	Boot Medium
DS6001 Processor Board in a SCALEXIO AutoBox or a SCALEXIO LabBox[2)]	CFast card

[1)] If upgraded to the Hypervisor Extension from an existing system, the SCALEXIO Processing Unit with E3-1275v6 processor might require a BIOS update. For more information, contact dSPACE Support (www.dspace.com/go/supportrequest).

[2)] An upgrade to the Hypervisor Extension from an existing system has to be performed exclusively by dSPACE. For more information, contact dSPACE Support (www.dspace.com/go/supportrequest).

- The boot medium, which is delivered with the SCALEXIO Hypervisor Extension.

Software The Hypervisor Extension requires the following software:

- ConfigurationDesk for implementing and building real-time applications running in the SCALEXIO real-time environment.

  Tip

  ConfigurationDesk is part of the regular dSPACE Release and is installed by selecting the **ConfigurationDesk** product set during setup.

- MATLAB/Simulink for modeling a behavior model, which can then be connected to the I/O functionality in ConfigurationDesk.

  For using MATLAB/Simulink directly with ConfigurationDesk, the **Model Interface Package for Simulink** must be additionally installed on your PC.

  The **Model Interface Package for Simulink** is part of the regular dSPACE Release and is installed by selecting the **Model Implementation Package for Simulink** product set during setup.

  Tip

  As an alternative, you can connect container files (for example SIC files, FMUs) to ConfigurationDesk. In this case a MATLAB/Simulink installation is not required on your PC.

- An Internet browser for accessing the SCALEXIO processing hardware Web interface to run virtual machines (Google Chrome or Microsoft Edge Chromium are recommended).
- A file explorer to manage the following files:
  - Setup files for the hypervisor and virtual machines
  - Linux applications and files
- A Linux distribution, such as Ubuntu (Ubuntu 18.04.5 is tested and recommended), Debian, CentOS 7, etc.

  The Linux distribution has to be installed on every virtual machine. Refer to How to Install a Linux Distribution on a Virtual Machine on page 37 .

Optional software You can create Linux applications in a virtual machine on the SCALEXIO system or on a stand-alone development system. The stand-alone development system has to be based on one of the following Linux 64-bit distributions:

- Ubuntu 18.04 LTS
- Debian (Stretch)
- CentOS 7

Related topics

Basics

Features of the SCALEXIO Hypervisor Extension..... 16
Introduction to Real-Time Linux..... 19
Overview of the SCALEXIO Hypervisor Extension..... 13

Introduction to Real-Time Linux

What is Real-Time Linux?

Real-Time Linux is a synonym for the RT_PREEMPT patch set available for the Linux kernel. It adds preemptability to the Linux kernel so it enables low latency execution, which is required by real-time applications.

Core isolation

The Linux kernel supports the isolation of individual CPU cores so that real-time applications running on an isolated core are not interfered by the operating system's background services or any applications running on non-isolated cores. All interrupts are by default handled by the non-isolated cores. Only the most basic kernel services are running on the isolated cores and if an interrupt is to be serviced on an isolated core, it must be routed to that core manually. The core isolation feature enables the execution of an application under hard real-time conditions when running on an isolated CPU core.

Interrupt affinity

The interrupt routing has to be set up manually whenever an interrupt is to be serviced on an isolated CPU core. For Linux, this is done by setting the interrupt affinity. A precise documentation on this topic is provided in the documentation directory of the Linux kernel source package. The file with information about interrupt affinities is called `IRQ-affinity.txt` and can be downloaded from several sources in the Internet or is supplied by dSPACE Support on request. Refer to this document if you want to service an interrupt on an isolated CPU core.

Scheduling classes

Linux provides five different scheduling classes:

- SCHED_BATCH
- SCHED_IDLE
- SCHED_RR
- SCHED_FIFO
- SCHED_OTHER

SCHED_RR and SCHED_FIFO are the relevant classes for building real-time applications. Both classes use static priorities from 1 … 99 (higher numbers equal higher priorities). The other scheduling classes use **nice values** that range from -20 … +19. Threads associated with the SCHED_RR or SCHED_FIFO scheduling classes are always prioritized over threads associated with the other scheduling classes.

Limits for executing priority tasks

The execution of real-time threads is limited as follows:

- Soft limit: Maximum priority value = 49.
- Hard limit: Maximum priority value = 99.

The soft limit normally prevents the execution of real-time threads with a priority value greater than 49. Nevertheless, the user can modify the soft limit by using the command `ulimit -r PRIO`, where `PRIO` is a value from 50 … 99. The predefined soft limit prevents the execution of real-time threads at priority levels exceeding those of the Linux kernel services, because this setup might result in a non-responsive system.

POSIX.1b real-time extension library

The GNU C library adheres to the POSIX.1-2008 standard and provides most of the POSIX.1b real-time extensions which are required to create real-time applications. On request, dSPACE Support provides information regarding existing limitations in the current POSIX.1b real-time extension library.

Related topics

Basics

Features of the SCALEXIO Hypervisor Extension 16
Overview of the SCALEXIO Hypervisor Extension 13
Requirements of the SCALEXIO Hypervisor Extension 17

Accessing a SCALEXIO System with Hypervisor Extension

Tasks and accessing methods

You can access a SCALEXIO system with Hypervisor Extension in different ways to perform different tasks:

Task	Access by …
Working with third-party tools under Linux	VNC (refer to How to Access a Virtual Machine on the SCALEXIO System via VNC on page 26)
	SSH (refer to How to Access a Virtual Machine on the SCALEXIO System via SSH on page 24)
Managing files	Windows File Explorer (refer to How to Access the Hypervisor Extension File System by Windows File Explorer on page 22)
	VNC (refer to How to Access a Virtual Machine on the SCALEXIO System via VNC on page 26)
	SSH (refer to How to Access a Virtual Machine on the SCALEXIO System via SSH on page 24)
Via the SCALEXIO processing hardware Web interface: ▪ Configuring the SCALEXIO processing hardware ▪ Starting virtual machines ▪ Stopping virtual machines (only as an exception)	Internet browser (refer to How to Access the SCALEXIO Web Interface and Virtual Machines by an Internet Browser on page 23)

Where to go from here

Information in this section

How to Access the Hypervisor Extension File System by Windows File Explorer 22

How to Access the SCALEXIO Web Interface and Virtual Machines by an Internet Browser 23

How to Access a Virtual Machine on the SCALEXIO System via SSH 24

How to Access a Virtual Machine on the SCALEXIO System via VNC 26

The Hypervisor Page of the SCALEXIO Web Interface 28

How to Access the Hypervisor Extension File System by Windows File Explorer

Objective

To manage files on the boot medium of the SCALEXIO processing hardware, for example, to copy configuration files from there to the host PC and vice versa.

Precondition

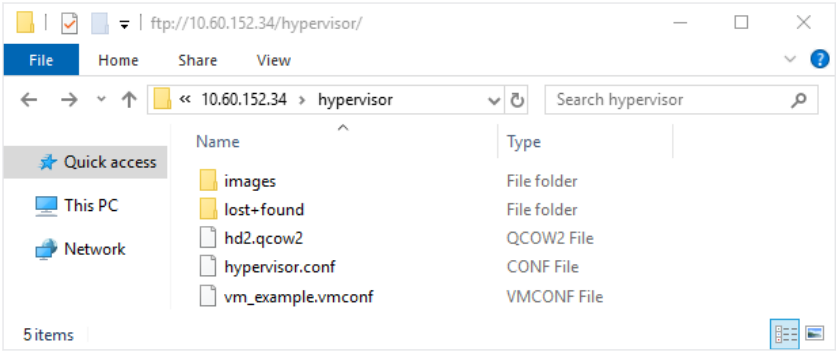
The host PC is connected to the SCALEXIO system via Ethernet.

Method

To access the Hypervisor Extension file system by Windows File Explorer

1 Start Windows File Explorer on the PC.

2 Enter `ftp://<IP address of the SCALEXIO system>/hypervisor` in the path edit field.



3 Perform your planned tasks.

Result

You accessed the Hypervisor Extension file system by Windows File Explorer.

Related topics

Basics

The Hypervisor Page of the SCALEXIO Web Interface.....28

HowTos

How to Access a Virtual Machine on the SCALEXIO System via SSH.....24
How to Access a Virtual Machine on the SCALEXIO System via VNC.....26
How to Access the SCALEXIO Web Interface and Virtual Machines by an Internet Browser.....23

How to Access the SCALEXIO Web Interface and Virtual Machines by an Internet Browser

Objective

To connect the SCALEXIO processing hardware for the following purposes:

- For getting an overview on virtual machines.
- For starting (and stopping) virtual machines.
- For installing Linux distributions on virtual machines.
- Optional: For working with virtual machines.

Recommended software tools

You are recommended to use the Google Chrome or Microsoft Edge Chromium Internet browser to access the SCALEXIO processing hardware Web interface.

Precondition

The host PC is connected to the SCALEXIO system via Ethernet.

Method

To access the SCALEXIO processing hardware Web interface by an Internet browser

1. Start an Internet browser on the PC.
2. Enter the IP address of the SCALEXIO system in the address edit field of the Internet browser.

The Web interface of the SCALEXIO processing hardware opens.

3
- For an overview of installed virtual machines and for starting and stopping them, go to the **Hypervisor** page of the Web interface. For more information, refer to The Hypervisor Page of the SCALEXIO Web Interface on page 28.
- For instructions on configuring the SCALEXIO processing hardware via the Web interface, refer to the documentation of the SCALEXIO processing hardware.

Result

You accessed the SCALEXIO processing hardware Web interface by an Internet browser.

Related topics

Basics

The Hypervisor Page of the SCALEXIO Web Interface.....28

HowTos

How to Access a Virtual Machine on the SCALEXIO System via SSH.....24
How to Access a Virtual Machine on the SCALEXIO System via VNC.....26
How to Access the Hypervisor Extension File System by Windows File Explorer.....22

How to Access a Virtual Machine on the SCALEXIO System via SSH

Objective

To establish an SSH connection, for example, to manage and/or run third-party Linux tools or other Linux programs on virtual machines on the SCALEXIO system.

Software tool

To establish the SSH connection, you can use the SSH and telnet client PuTTY, which you can download from various sources on the Internet.

Precondition

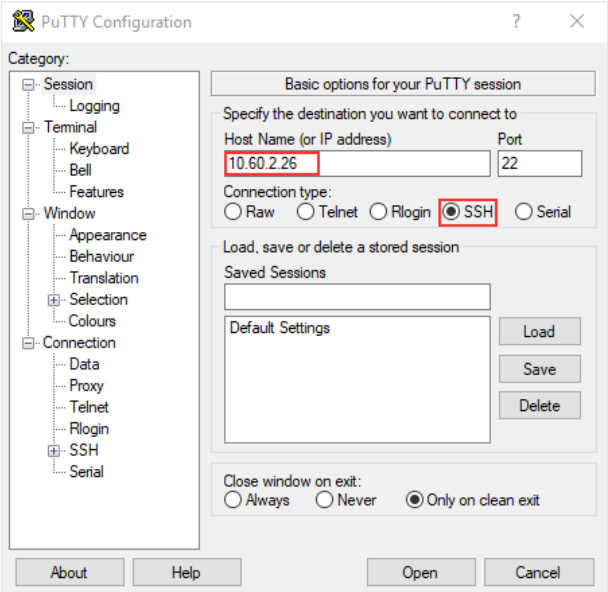
- The host PC is connected to the SCALEXIO system via Ethernet.
- An SSH and telnet client, such as PuTTY, is available on the host PC.
- An SSH server is installed on the virtual machine. Refer to How to Add an SSH Server to a Linux Distribution on page 39.
- The virtual machine that you want to access is running.

Method

To access Linux on the SCALEXIO system via an SSH connection

1 Start PuTTY on the host PC.

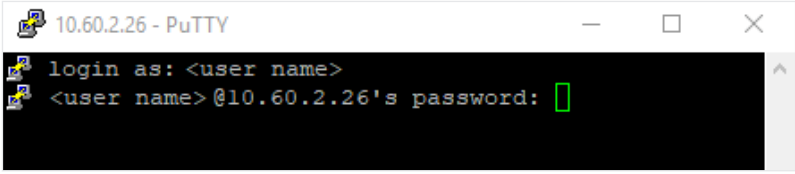
2 In the PuTTY Configuration window, enter the IP address or the name of the installed Linux distribution (Linux computer IP or name) in the related edit field and set the connection type to SSH.



3 Click Open.

The remote session window opens and displays the Linux terminal screen.

4 Log on with the user name and password you specified during the installation of the Linux distribution.



You are logged on and can start performing your tasks.

Result

You accessed a virtual machine under Linux on the SCALEXIO system via an SSH connection.

Terminating the SSH connection

To close the SSH connection, enter **`exit`** in the command line.

The SSH connection and the remote session window are closed.

Related topics

Basics

The Hypervisor Page of the SCALEXIO Web Interface..... 28

HowTos

How to Access a Virtual Machine on the SCALEXIO System via VNC..... 26
How to Access the Hypervisor Extension File System by Windows File Explorer..... 22
How to Access the SCALEXIO Web Interface and Virtual Machines by an Internet Browser..... 23

How to Access a Virtual Machine on the SCALEXIO System via VNC

Objective

To establish a Virtual Network Control (VNC) connection for remotely controlling a virtual machine on the SCALEXIO system.

Software tool

To remotely control the virtual machine, you can use a VNC viewer, for example, UltraVNC Viewer, which you can download from various sources on the Internet.

A VNC *server* is started by default on the SCALEXIO system if specified in the configuration file of the virtual machine.

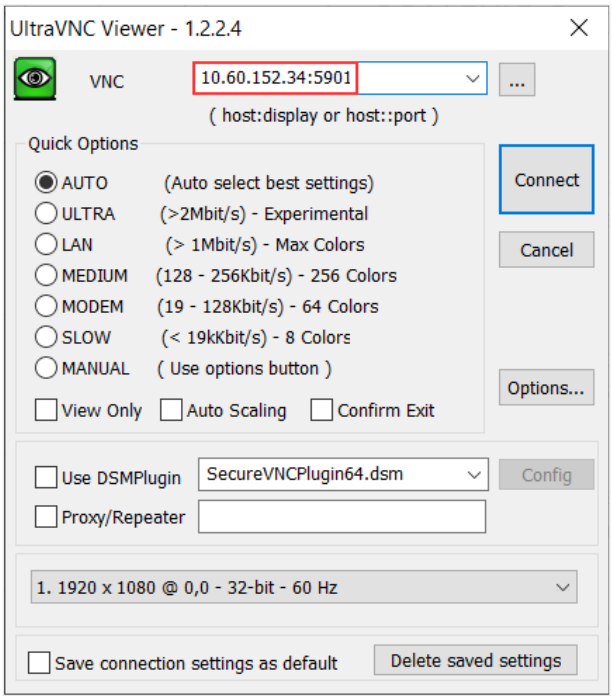
Preconditions

- The host PC is connected to the SCALEXIO system via Ethernet.
- A VNC viewer is available on the host PC.
- The virtual machine that you want to access is running.

Method

To access a virtual machine on the SCALEXIO system via VNC

1. Start a VNC viewer, for example, UltraVNC Viewer, on the host PC.
2. Enter `<IP address of the SCALEXIO system>:<port number>` in the VNC edit field. The port number is calculated by adding the ID of the virtual machine to the basic VNC port number 5900. For example, the port number for the virtual machine with the ID 1 is 5901.



3 Click Connect.

The remote session window opens and displays the screen of the accessed virtual machine. You can start performing your tasks.

Result

You accessed a virtual machine on the SCALEXIO system via VNC.

Terminating the VNC connection

To close the VNC connection, close the remote session window.

Related topics

Basics

The Hypervisor Page of the SCALEXIO Web Interface 28

HowTos

How to Access a Virtual Machine on the SCALEXIO System via SSH 24
How to Access the Hypervisor Extension File System by Windows File Explorer 22
How to Access the SCALEXIO Web Interface and Virtual Machines by an Internet Browser 23

The Hypervisor Page of the SCALEXIO Web Interface

Introduction

The Hypervisor page of the SCALEXIO processing hardware Web interface provides an overview and the control of virtual machines available on the SCALEXIO system with Hypervisor Extension and access to running virtual machines.

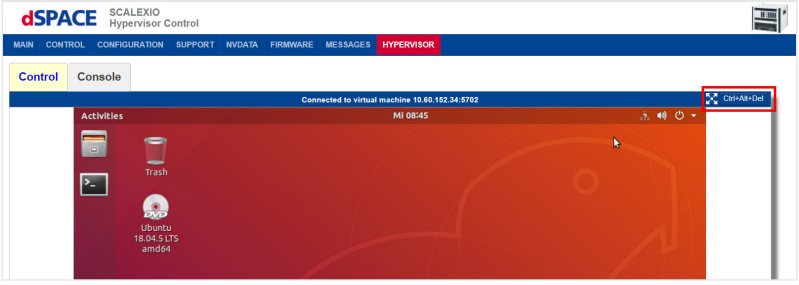
Elements of the Hypervisor page

The Hypervisor page provides a Control page and a Console page.

- The Hypervisor Control page shows all virtual machines on the SCALEXIO system. Their basic characteristics are displayed and control buttons according to the current operation state are provided for each virtual machine.

You can start, stop, and pause/resume the virtual machines. To access the console of a virtual machine, click the related Console button and go to the Console page.

- The Hypervisor Console page displays the screen of the selected virtual machine, such as the Ubuntu screen shown in the following illustration.



You can access the virtual machine/Linux distribution. Two additional buttons are provided in the top right corner of the Console page: The button on the left lets you change the display of the Console page to full-screen mode and the button on the right lets you send Ctrl + Alt + Del to the virtual machine.

Related topics

HowTos

How to Access a Virtual Machine on the SCALEXIO System via SSH..... 24
How to Access a Virtual Machine on the SCALEXIO System via VNC..... 26
How to Access the Hypervisor Extension File System by Windows File Explorer..... 22
How to Access the SCALEXIO Web Interface and Virtual Machines by an Internet Browser..... 23

Installing the SCALEXIO Hypervisor Extension

Where to go from here

Information in this section

Installation Workflow.....31
How to Install the SCALEXIO Hypervisor Extension Boot Medium.....32
How to Create a Virtual Machine.....36
How to Install a Linux Distribution on a Virtual Machine.....37
How to Add an SSH Server to a Linux Distribution.....39
How to Add Hypervisor Tools to a Linux Distribution.....40
How to Add NFS share to a Linux Distribution.....42

Installation Workflow

Steps to be performed

Observe the following sequence, because the installation methods are based on it.

1. Install the SCALEXIO Hypervisor Extension boot medium in the processing hardware. Refer to How to Install the SCALEXIO Hypervisor Extension Boot Medium on page 32.
2. Configure a virtual machine on the SCALEXIO system. You can either use the preinstalled example or create a new one. Refer to:
   - How to Create a Virtual Machine on page 36.
   - Configuring the Hypervisor and Virtual Machines on page 69.
3. Install a Linux distribution on the virtual machine. Refer to How to Install a Linux Distribution on a Virtual Machine on page 37.

4. Optional: Add software or packages to the Linux installation, for example:
   - Add an SSH server package so that you can access the virtual machine via an SSH connection. Refer to How to Add an SSH Server to a Linux Distribution on page 39.
   - Add the dSPACE hypervisor tool package so that you can use shared memory. Refer to How to Add Hypervisor Tools to a Linux Distribution on page 40.
   - Add NFS share so that you can use network file sharing. Refer to How to Add NFS share to a Linux Distribution on page 42.

Related topics

HowTos

How to Add an SSH Server to a Linux Distribution.....39
How to Add Hypervisor Tools to a Linux Distribution.....40
How to Add NFS share to a Linux Distribution.....42
How to Create a Virtual Machine.....36
How to Install a Linux Distribution on a Virtual Machine.....37
How to Install the SCALEXIO Hypervisor Extension Boot Medium.....32

How to Install the SCALEXIO Hypervisor Extension Boot Medium

Objective

To install the boot medium of the SCALEXIO Hypervisor Extension in a SCALEXIO Processing Unit.

Restriction

Note

- The installation of the SCALEXIO Hypervisor Extension boot medium can be performed by the user only for SCALEXIO Processing Units.
- For a DS6001 Processor Board, the installation of the SCALEXIO Hypervisor Extension boot medium has to be performed exclusively by dSPACE. For more information, contact dSPACE Support (www.dspace.com/go/supportrequest).

Precondition

The SCALEXIO Processing Unit is powered down.

NOTICE

Inserting a boot medium when the SCALEXIO Processing Unit is running.

Risk of damaging the boot medium.

- Ensure that the SCALEXIO Processing Unit is powered down before you insert the boot medium.

Possible methods

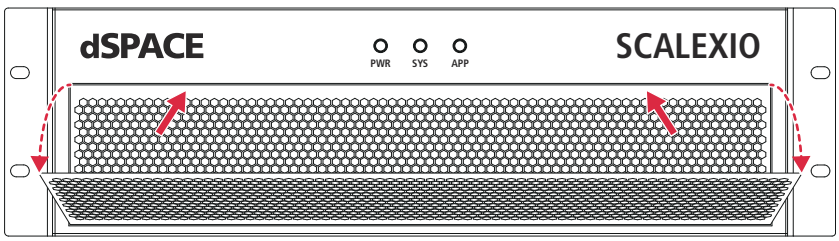
Depending on the SCALEXIO Processing Unit and the boot medium, you have to proceed as follows:

- For installing a CompactFlash card in a SCALEXIO Processing Unit with E5-2640v3 processor or a SCALEXIO Processing Unit with E3-1275v6 processor, refer to Method 1 on page 33.
- For installing an SSD in a SCALEXIO Processing Unit with Intel 6208U processor, refer to Method 2 on page 34.

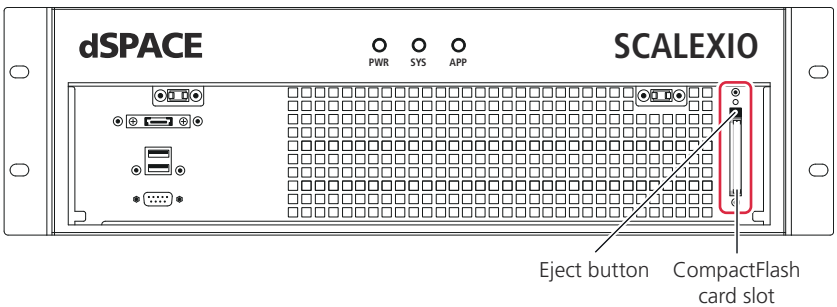
Method 1

To install the SCALEXIO Hypervisor Extension CompactFlash card in a SCALEXIO Processing Unit

1 Press the Processing Unit's front flap at the upper edge and fold it down as shown in the following illustration.



2 Press the eject button of the card slot.



3 Remove the CompactFlash card with the original SCALEXIO Firmware from the card slot.

4 Insert the CompactFlash card with the SCALEXIO Hypervisor Extension into the card slot.

5 Close the front flap of the Processing Unit.

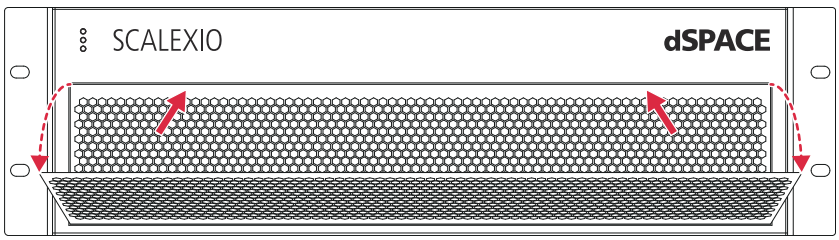
6 Store the CompactFlash card with the original SCALEXIO Firmware in a safe place so you can use it again if required.

7 Perform a firmware update to ensure that all the components of your SCALEXIO system work with the correct firmware version. For information on updating the firmware, refer to https://www.dspace.com/go/firmware.

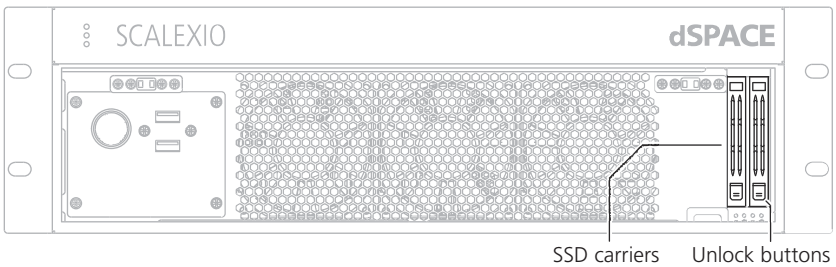
Method 2

To install the SCALEXIO Hypervisor Extension SSD in a SCALEXIO Processing Unit

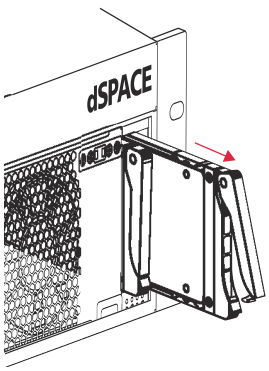
1 Press the Processing Unit's front flap at the upper edge and fold it down as shown in the following illustration.



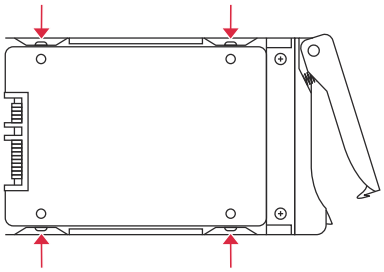
2 Press the unlock button of the carrier with the original SCALEXIO Firmware SSD.



3 Remove the carrier from the carrier slot.



4 Remove the four screws holding the SSD and remove the SSD with the original SCALEXIO Firmware from the carrier.



5 Install the SSD with the SCALEXIO Hypervisor Extension in the carrier by using the four screws you previously removed.

6 Insert the carrier into the carrier slot again.

7 Close the front flap of the Processing Unit.

8 Store the SSD with the original SCALEXIO Firmware in a safe place so you can use it again if required.

9 Perform a firmware update to ensure that all the components of your SCALEXIO system work with the correct firmware version. For information on updating the firmware, refer to https://www.dspace.com/go/firmware.

Result

You installed the boot medium of the SCALEXIO Hypervisor Extension in the SCALEXIO Processing Unit.

Next step

Configure the hypervisor and the preinstalled virtual machine or create a new one. Refer to

- Configuring the Hypervisor and Virtual Machines on page 69.
- How to Create a Virtual Machine on page 36.

Related topics

Basics

Installation Workflow.....31

HowTos

How to Add an SSH Server to a Linux Distribution.....39
How to Add Hypervisor Tools to a Linux Distribution.....40
How to Add NFS share to a Linux Distribution.....42
How to Create a Virtual Machine.....36
How to Install a Linux Distribution on a Virtual Machine.....37

How to Create a Virtual Machine

Objective

To create a virtual machine for the SCALEXIO system with Hypervisor Extension.

Overview

Virtual machines for the SCALEXIO system with Hypervisor Extension are specified by configuration files that must be located in the `hypervisor` folder of the SCALEXIO Hypervisor boot medium. Each virtual machine requires an individual configuration file.

Preconditions

- The SCALEXIO Hypervisor boot medium is installed in the SCALEXIO processing hardware. Refer to How to Install the SCALEXIO Hypervisor Extension Boot Medium on page 32.
- The host PC is connected to the SCALEXIO system via Ethernet.

Method

To create a virtual machine

1. Copy an existing configuration file of a virtual machine (for example, template.vmconf) from the `hypervisor` folder on the boot medium to your PC. Refer to How to Access the Hypervisor Extension File System by Windows File Explorer on page 22.
2. Rename the file to `<myVM>.vmconf` .
3. Open the file with an editor such as Notepad and modify the configuration for the new virtual machine. Refer to Configuring a Virtual Machine on page 72.
4. Copy the new configuration file to the `hypervisor` folder on the boot medium.

Result

You created a virtual machine for the SCALEXIO system with Hypervisor Extension.

Next step

- Install a Linux distribution on the virtual machine. Refer to How to Install a Linux Distribution on a Virtual Machine on page 37.

Related topics

Basics

Installation Workflow 31

HowTos

How to Add an SSH Server to a Linux Distribution 39
How to Add Hypervisor Tools to a Linux Distribution 40
How to Add NFS share to a Linux Distribution 42

How to Install a Linux Distribution on a Virtual Machine..... 37
How to Install the SCALEXIO Hypervisor Extension Boot Medium..... 32

How to Install a Linux Distribution on a Virtual Machine

Objective

To install a Linux distribution, such as Ubuntu 18.04.5, on a virtual machine that runs on the SCALEXIO system with Hypervisor Extension.

Preconditions

- The SCALEXIO Hypervisor boot medium is installed in the SCALEXIO processing hardware. Refer to How to Install the SCALEXIO Hypervisor Extension Boot Medium on page 32.
- The hypervisor and the virtual machine are configured as required. Refer to Configuring the Hypervisor and Virtual Machines on page 69.
- The host PC is connected to the SCALEXIO system via Ethernet.

Workflow

A Linux distribution is installed on a virtual machine in a three-part process:

- Specifying the Linux distribution (refer to Part 1 on page 37).
- Installing the Linux distribution (refer to Part 2 on page 38).
- Disabling unattended automatic upgrades (refer to Part 3 on page 38).

Part 1

To specify the Linux distribution that has to be installed

1. Copy an ISO image of the Linux distribution to the `hypervisor` folder on the boot medium. Refer to How to Access the Hypervisor Extension File System by Windows File Explorer on page 22.
2. Specify a CD-ROM block device for installing the Linux distribution in the virtual machine's configuration file. Include the following entry in the `block devices` array:

```
{
   "file" : "<iso_image_file name>.iso",
   "type" : "cdrom",
   "virtio" : true,
},
```

Tip

A CD-ROM block device for installing Ubuntu 18.04.5 is specified in the `template.vmconf` example configuration file in the `hypervisor` folder on the boot medium. If you want to install a different Linux distribution, you have to modify the entry accordingly.

Refer to Configuring the Hypervisor and Virtual Machines on page 69.

Part 2

To install the Linux distribution on a virtual machine

1 Start the virtual machine on which you want to install the Linux distribution. Refer to How to Start a Virtual Machine on page 47.

2 Go to the **Hypervisor Console** page of the virtual machine.

The screen of the virtual machine is displayed and shows the Linux screen after the virtual machine booted completely.

3 Start and perform the installation when prompted. The installation process depends on the particular Linux distribution.

> **Note**
>
> During the installation, you have to specify a user name, a password, and a computer name for the current installation. Specify a unique computer name because it can be used to access the virtual machine via SSH.

Part 3

To disable (unattended) automatic updates

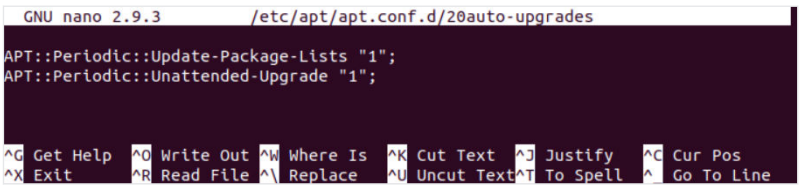
> **Note**
>
> The following method is tested for Ubuntu 18.04.5. For other Linux distributions, different methods might apply.

1 Open a terminal window to work with the command line.

2 Enter **`sudo apt update`** to initially update the package list.

3 Enter **`sudo nano /etc/apt/apt.conf.d/20auto-upgrades`**.

The auto-upgrade list opens:



4 Modify entries or add to the list as follows:

`APT::Periodic::Update-Package-Lists "0";`

`APT::Periodic::Unattended-Upgrade "0";`

5 Press **`Ctrl`** + **`S`** to save the list.

6 Press **`Ctrl`** + **`X`** to exit the editor.

Result

You installed a Linux distribution on a virtual machine running on the SCALEXIO system with Hypervisor Extension.

Next steps

You can now continue with one or more of the following steps:

- Start working with the virtual machine under Linux.
- Add software or packages to the Linux installation. For example:
  - Add an SSH server package so that you can access the virtual machine via an SSH connection. Refer to How to Add an SSH Server to a Linux Distribution on page 39.
  - Add the dSPACE hypervisor tool package so that you can use shared memory. Refer to How to Add Hypervisor Tools to a Linux Distribution on page 40.
  - Add NFS share so that you can use network file sharing. Refer to How to Add NFS share to a Linux Distribution on page 42.
- Back up the virtual machine including its Linux distribution. Refer to Backing Up and Restoring Virtual Machines on page 83.

Related topics

Basics

Installation Workflow.....31

HowTos

How to Add an SSH Server to a Linux Distribution.....39
How to Add Hypervisor Tools to a Linux Distribution.....40
How to Add NFS share to a Linux Distribution.....42
How to Create a Virtual Machine.....36
How to Install the SCALEXIO Hypervisor Extension Boot Medium.....32

How to Add an SSH Server to a Linux Distribution

Objective

To install an SSH server package so that you can access the virtual machine via an SSH connection.

Precondition

A virtual machine with a Linux distribution is installed on the SCALEXIO system. Refer to How to Install a Linux Distribution on a Virtual Machine on page 37.

Method

To install an SSH server package

1. Start the virtual machine.
2. Open a terminal window to perform the installation from the command line.
3. Enter `sudo apt-get install openssh-server`.

4 If required, enter the password.
The SSH server package is installed.

5 Shut down Linux and start the virtual machine again.

Result

You added an SSH server to a Linux distribution.

Related topics

Basics

Installation Workflow.....31

HowTos

How to Add Hypervisor Tools to a Linux Distribution.....40
How to Add NFS share to a Linux Distribution.....42
How to Create a Virtual Machine.....36
How to Install a Linux Distribution on a Virtual Machine.....37
How to Install the SCALEXIO Hypervisor Extension Boot Medium.....32

How to Add Hypervisor Tools to a Linux Distribution

Objective

To install the dSPACE hypervisor tool package so that you can use shared memories and interrupts.

Restriction

The hypervisor tools work only with Ubuntu Linux distributions. If you want to use shared memories and interrupts with a Linux distribution other than Ubuntu, contact dSPACE Support (www.dspace.com/go/supportrequest).

Precondition

A virtual machine with an Ubuntu Linux distribution, such as Ubuntu 18.04.5, is installed on the SCALEXIO system. Refer to How to Install a Linux Distribution on a Virtual Machine on page 37.

Workflow

The hypervisor tools are installed in a two-part process:

1. Specifying a CD-ROM block device from which to install the tools (refer to Part 1 on page 41).
2. Installing the hypervisor tools (refer to Part 2 on page 41).

Part 1

To specify a CD-ROM block device from which to install the hypervisor tools

1 Add the following entry to the block device array in the virtual machine's configuration file:

```
{
  "file" : "%rth-tools%",
  "type" : "cdrom",
  "virtio" : true,
},
```

For information on modifying the configuration, refer to Configuring the Hypervisor and Virtual Machines on page 69.

Part 2

To install the hypervisor tools

1 Start the virtual machine.

2 Open a terminal window to perform the installation from the command line.

3 Enter `cd /media/<username>/dSPACE_RTH_TOOLS`.

4 Enter `sudo apt install ./dsrth-tools_1.0-1.deb`.

5 If required, enter the password.
The hypervisor tools are installed.

6 Shut down Linux and start the virtual machine again.

Result

You added the dSPACE hypervisor tools to a Linux distribution.

Related topics

Basics

Installation Workflow 31

HowTos

How to Add an SSH Server to a Linux Distribution 39
How to Add NFS share to a Linux Distribution 42
How to Create a Virtual Machine 36
How to Install a Linux Distribution on a Virtual Machine 37
How to Install the SCALEXIO Hypervisor Extension Boot Medium 32

How to Add NFS share to a Linux Distribution

Objective

To add NFS share so that you can use network file sharing.

Precondition

A virtual machine with a Linux distribution is installed on the SCALEXIO system. Refer to How to Install a Linux Distribution on a Virtual Machine on page 37.

Workflow

NFS share is installed in a four-part process:

1. Specifying NFS share in the virtual machine's configuration file (refer to Part 1 on page 42).
2. Installing the NFS server (refer to Part 2 on page 42).
3. Specifying the NFS file system (refer to Part 3 on page 42).
4. Configuring a Linux Ethernet connection for NFS share (refer to Part 4 on page 43).

Part 1

To specify NFS share in the configuration file

1 Add the following entry to the virtual machine's configuration file:

```
"network_share" : "192.168.10.2/home/<user name>/share",
"network_share_timeout" : 20,
```

For more information, refer to Configuring the Hypervisor and Virtual Machines on page 69.

Part 2

To install an NFS server

1 Start the virtual machine.

2 Open a terminal window to perform the installation from the command line.

3 Enter `sudo apt install nfs-kernel-server`.

4 If required, enter the password.

- The NFS server package is installed.
- The `vm<virtual machine ID>` folder is created in the `hypervisor` folder of the boot medium.

Part 3

To specify the NFS file system and to modify the access control list

1 Enter `mkdir share` in the command line.

2 Enter `chmod 777 share` .

3 Enter `sudo nano /etc/exports` to edit the access control list for export file systems with the GNU nano editor.

4 Add **/home/<user name>/share *(rw,no_subtree_check)** to the access control list:

```
  GNU nano 2.9.3                    /etc/exports

# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subS
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/home/scalexio/share *(rw,no_subtree_check)

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

5 Press **Ctrl** + **S** to save the list.

6 Press **Ctrl** + **X** to exit the editor.

7 Enter **sudo exportfs -a** in the command line to activate the entry.

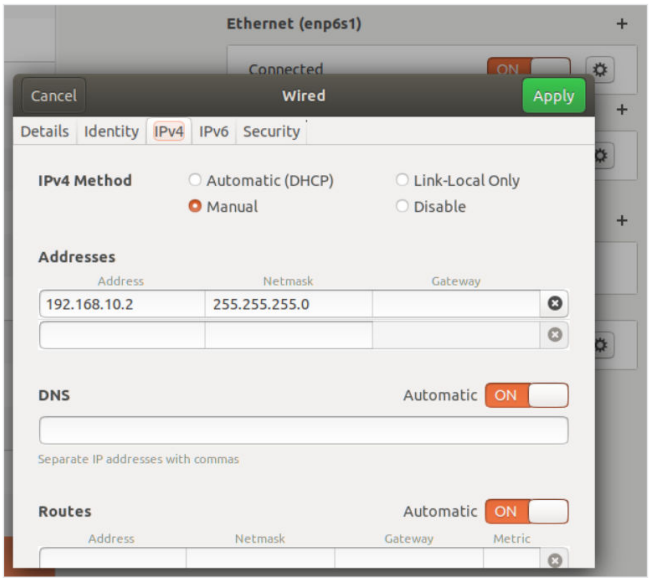
Part 4

To configure a Linux network interface for NFS share

1 In the configuration file, specify the IP from the NFS sharing specification as the IP address of a free Ethernet connection of the Linux distribution.

2 Specify an appropriate network mask.

Refer to the following Ubuntu screenshot as an example:



Result

You added NFS share to a Linux distribution.

Related topics

Basics

Installation Workflow..... 31

HowTos

How to Add an SSH Server to a Linux Distribution..... 39
How to Add Hypervisor Tools to a Linux Distribution..... 40
How to Create a Virtual Machine..... 36
How to Install a Linux Distribution on a Virtual Machine..... 37
How to Install the SCALEXIO Hypervisor Extension Boot Medium..... 32

Starting a SCALEXIO System with Hypervisor Extension

Where to go from here

Information in this section

How to Power Up and Boot a SCALEXIO System with Hypervisor Extension.....45

How to Start a Virtual Machine.....47

Booting a Virtual Machine.....48

How to Power Up and Boot a SCALEXIO System with Hypervisor Extension

Possible methods

Depending on the SCALEXIO system you use, proceed according to one of the following methods:

- For powering up a SCALEXIO rack system, refer to Method 1 on page 45.
- For powering up a stand-alone SCALEXIO Processing Unit, refer to Method 2 on page 46.
- For powering up a DS6001 Processor Board in a SCALEXIO LabBox, refer to Method 3 on page 46.
- For powering up a DS6001 Processor Board in a SCALEXIO AutoBox, refer to Method 4 on page 46.

Method 1

To power up and boot a SCALEXIO rack system

1 - SCALEXIO customized rack system: Use the mains switch of the system and/or proceed according to the power-up instructions of the respective SCALEXIO customized rack system.

- SCALEXIO off-the-shelf rack system (9 U or 12 U): Switch on the battery simulation power supply unit located below the SCALEXIO Processing Unit.

Method 2

To power up and boot a stand alone SCALEXIO Processing Unit

1
- SCALEXIO Processing Unit with E3-1275v6 processor: Set the mains switch at the rear of the Processing Unit to the **On** position. For information on locating the controls, refer to SCALEXIO System Mains Controls on page 92.
- SCALEXIO Processing Unit with E5-2640v3 processor: Check the mains switch at the rear of the Processing Unit. For information on locating the controls, refer to SCALEXIO System Mains Controls on page 92.
  - If the mains switch is in the **Off** position, switch it on. The Processing Unit will power up immediately.
  - If the mains switch is in the **On** position, the Processing Unit was powered down by the On/Off button after the last operation. Press the On/Off button to power it up again.

Method 3

To power up and boot a DS6001 Processor Board in a SCALEXIO LabBox via the mains controls

1 Set the mains switch at the rear of the SCALEXIO LabBox to the **On** position.

2 Press the On/Off button on the front panel to power it up.

For information on locating the controls, refer to SCALEXIO System Mains Controls on page 92.

For more information and for alternative methods to power up a SCALEXIO LabBox, refer to Powering Up and Shutting Down the LabBox (SCALEXIO Hardware Installation and Configuration 🕮).

Method 4

To power up and boot a DS6001 Processor Board in a SCALEXIO AutoBox via the remote pin of the power input connector

1 Connect the SCALEXIO AutoBox to the power supply voltage.

2 Use the output of the ignition/driving switch (KL15), for example, to change the voltage level at the remote pin 4 of the power input connector from below +2 V DC to a voltage level above +6 V DC. This switches the SCALEXIO AutoBox to normal operating mode.

For information on locating the power input connector, refer to SCALEXIO System Mains Controls on page 92.

For more information and for alternative methods to power up a SCALEXIO AutoBox, refer to Powering Up and Shutting Down SCALEXIO AutoBox (SCALEXIO Hardware Installation and Configuration 🕮).

Result

You powered up and booted the SCALEXIO system with Hypervisor Extension. When the Power-On Self-Test is completed, the POST screen on the optional monitor disappears and the boot process starts.

The system is fully available when the following condition is met:

- SCALEXIO Processing Unit: The SYS LED on the front panel is lit green (or orange if the Processing Unit is not connected to the host PC via Ethernet.
- SCALEXIO LabBox or AutoBox: The SYS LED on the front panel is lit green.

Related topics

Basics

Booting a Virtual Machine.....48

HowTos

How to Start a Virtual Machine.....47

How to Start a Virtual Machine

Objective

To start virtual machines on the SCALEXIO system with Hypervisor Extension.

Possible methods

Virtual machines can be started as follows:

- Starting a virtual machine automatically (refer to Method 1 on page 47).
- Starting a virtual machine manually (refer to Method 2 on page 47).

Method 1

To start a virtual machine automatically

1 Locate and modify, or add the autostart entry in the configuration file of the virtual machine as follows:

```
"autostart" : true,
```

For information on modifying the configuration, refer to Configuring the Hypervisor and Virtual Machines on page 69.

The virtual machine automatically starts with the SCALEXIO system. For details on the process, refer to Booting a Virtual Machine on page 48.

Method 2

To start a virtual machine manually

1 Access the SCALEXIO system via an Internet browser. Refer to How to Access the SCALEXIO Web Interface and Virtual Machines by an Internet Browser on page 23.

2 On the **Hypervisor Control** page, start the virtual machine via the related **Start** button. Refer to The Hypervisor Page of the SCALEXIO Web Interface on page 28.

The virtual machine starts. For details on the process, refer to Booting a Virtual Machine on page 48.

Result

You can work with the virtual machine.

Related topics

Basics

Booting a Virtual Machine.....48

HowTos

How to Power Up and Boot a SCALEXIO System with Hypervisor Extension.....45

Booting a Virtual Machine

Purpose

Tracking the boot process of a virtual machine can be helpful both after initial installation and after modifying the system configuration, especially if you encounter problems during the boot process.

Accessing the boot manager

Depending on the configuration of a virtual machine, the boot process starts automatically or is started manually.

If the boot manager is enabled in the configuration file and you are using manual start mode, you can enter the boot manager as follows:

On the Hypervisor Console page, click the screen of the virtual machine and press **`Esc`** (for `bios` boot mode) or **`F12`** (for `efi` boot mode).

You can, for example, select the device from which booting is performed.

Booting steps

After parsing the boot configuration, the virtual machine performs the following steps as specified in the related configuration file:

- Assigning RAM and hardware resources to the virtual machine.
- Creating a hard disk file on the boot medium if the file does not yet exist.
- Creating shared memories.
- Starting Linux if installed and `autostart` is enabled.
- Mounting NFS share.

Error messages

Status and error messages are available on the Messages page of the SCALEXIO processing hardware Web interface.

The following errors during startup abort the boot process:

- Invalid RAM or CPU core assignment.
- Invalid configuration file entries.

Measures

- If the boot process aborts because of invalid resource assignments or configuration file entries, you have to modify the configuration. If you modified the configuration file of the virtual machine, start the virtual machine again. If you modified the configuration file of the hypervisor, you have to restart the entire system. Refer to:
  - Configuring the Hypervisor and Virtual Machines on page 69.
  - Starting a SCALEXIO System with Hypervisor Extension on page 45.
- If a virtual machine does not start at all and a backup is available, try to restore it. Refer to Backing Up and Restoring Virtual Machines on page 83.

  If this fails, you might have to replace the Linux distribution on the related virtual machine. Refer to Procedures for Modifying or Replacing a Linux Distribution on page 85.
- If the problem persists, replace the boot medium. To obtain a replacement boot medium, contact dSPACE Support (www.dspace.com/go/supportrequest).

  When the new boot medium is installed in the SCALEXIO processing hardware, data can be copied from the previous boot medium to the new one under Linux.

Tip

To transfer data between CompactFlash cards or CFast cards, connect an external card reader for the previous card to the SCALEXIO processing hardware so you can access both cards simultaneously.

Related topics

HowTos

How to Power Up and Boot a SCALEXIO System with Hypervisor Extension..... 45
How to Start a Virtual Machine..... 47

Shutting Down a SCALEXIO System with Hypervisor Extension

Avoiding property damage

NOTICE

Incorrectly shutting down a SCALEXIO system with Hypervisor Extension.

Risk of data loss and rendering the files of virtual machines unusable.

- Shut down a SCALEXIO system with Hypervisor Extension strictly according to the required workflow. Refer to Workflow of the System Shutdown Procedure on page 52.
- Back up the user data on the boot medium regularly to minimize the risks of data loss.

Where to go from here

Information in this section

Workflow of the System Shutdown Procedure..... 52

How to Shut Down a Virtual Machine..... 52

How to Switch Off a SCALEXIO System with Hypervisor Extension..... 54

Workflow of the System Shutdown Procedure

Avoiding property damage

NOTICE

Incorrectly shutting down a SCALEXIO system with Hypervisor Extension.

Risk of data loss and rendering the files of virtual machines unusable.

- Shut down a SCALEXIO system with Hypervisor Extension strictly according to the required workflow. Refer to Workflow of the System Shutdown Procedure on page 52.
- Back up the user data on the boot medium regularly to minimize the risks of data loss.

Workflow

1. Unload all running SCALEXIO real-time applications.
2. Close all the Linux applications.
3. Shut down the virtual machines. Refer to How to Shut Down a Virtual Machine on page 52.
4. Switch off the SCALEXIO system. Refer to How to Switch Off a SCALEXIO System with Hypervisor Extension on page 54.

Related topics

HowTos

How to Shut Down a Virtual Machine..... 52
How to Switch Off a SCALEXIO System with Hypervisor Extension..... 54

How to Shut Down a Virtual Machine

Objective

To properly shut down a virtual machine.

Possible methods

You are recommended to shut down a virtual machine by shutting down its Linux operating system. Use one of the following methods:

- Shutting down the Linux operating system from the Linux desktop (refer to Method 1 on page 53).
- Shutting down the Linux operating system from the Linux command line (refer to Method 2 on page 53).

It is also possible, but not recommended, to shut down a virtual machine directly from the SCALEXIO processing hardware Web interface (refer to Method 3 on page 53).

Method 1

To shut down a Linux operating system from the Linux desktop

1 Click the Power button on the Linux desktop.
   - Linux is shut down and the virtual machine is stopped.
   - For access via SSH and/or VNC, the remote connections are closed.
   - For access via the SCALEXIO processing hardware Web interface, the Linux screen is closed and the blank Console page is displayed.
2 Wait until the virtual machine's state display on the Hypervisor page of the Web interface changed to Stopped.

Method 2

To shut down the Linux operating system from the Linux command line

1 Enter `sudo shutdown -h now` in the command line of a Linux terminal window.

   You are prompted for the password.
2 Enter the password to confirm the command.
   - Linux is shut down and the virtual machine is stopped when the shutdown command is executed.
   - For access via SSH and/or VNC, the remote connections are closed.
   - For access via the SCALEXIO processing hardware Web interface, the Linux screen is closed and the blank Console page is displayed.
3 Wait until the virtual machine's state display on the Hypervisor page of the Web interface changed to Stopped.

Method 3

To shut down a virtual machine directly from the SCALEXIO processing hardware Web interface

NOTICE

Shutting down a virtual machine directly from the Web interface

This method might abort the execution of Linux and cause data loss.
- If possible, shut down the virtual machine via Linux. Refer to Method 1 on page 53 or Method 2 on page 53.

1 Go to SCALEXIO Web interface 〉 Hypervisor page 〉 Control page.
2 Click the Stop button of the virtual machine that you want to stop.
   - An ACPI signal is provided to the virtual machine.
   - If the installed Linux operating system supports ACPI signals, it shuts down the virtual machine.
   - After 120 seconds, the shutdown of the virtual machine is enforced.

Result

You shut down Linux and the related virtual machine.

Next steps

You can now continue with one of the following steps:

- Switch off the SCALEXIO processing hardware. Refer to How to Switch Off a SCALEXIO System with Hypervisor Extension on page 54.
- Change the configuration and start the virtual machine again. Refer to
  - Configuring the Hypervisor and Virtual Machines on page 69.
  - How to Start a Virtual Machine on page 47.

Related topics

HowTos

How to Switch Off a SCALEXIO System with Hypervisor Extension.....54
Workflow of the System Shutdown Procedure.....52

How to Switch Off a SCALEXIO System with Hypervisor Extension

Preconditions

NOTICE

Incorrectly shutting down a SCALEXIO system with Hypervisor Extension.

Risk of data loss and rendering the files of virtual machines unusable.

Before you switch off a SCALEXIO system with Hypervisor Extension, obtain the following preconditions:

- All the virtual machines running on the SCALEXIO system are shut down. Refer to How to Shut Down a Virtual Machine on page 52.
- All the SCALEXIO real-time applications are unloaded from the SCALEXIO processing hardware.

Possible methods

Depending on the SCALEXIO system you use, proceed according to one of the following methods:

- For switching off a SCALEXIO rack system, refer to Method 1 on page 55.
- For switching off a stand-alone SCALEXIO Processing Unit, refer to Method 2 on page 55.
- For switching off a DS6001 Processor Board in a SCALEXIO LabBox, refer to Method 3 on page 55.
- For switching off a DS6001 Processor Board in a SCALEXIO AutoBox, refer to Method 4 on page 55.

Method 1

To switch off a SCALEXIO rack system

1
- SCALEXIO customized rack system: Use the mains switch of the system and/or proceed according to the instructions of the respective SCALEXIO customized rack system.
- SCALEXIO off-the-shelf rack system (9 U or 12 U): Switch off the battery simulation power supply unit located below the SCALEXIO Processing Unit.

Method 2

To switch off a stand-alone SCALEXIO Processing Unit

1
- Stand-alone SCALEXIO Processing Unit with E3-1275v6 processor: Set the mains switch at the rear of the Processing Unit to the **Off** position. For information on locating the controls, refer to SCALEXIO System Mains Controls on page 92.
- Stand-alone SCALEXIO Processing Unit with E5-2640v3 processor: Proceed as follows:
  1. Press the On/Off button at the rear of the SCALEXIO Processing Unit and hold it for approx. 5 seconds until the Processing Unit powers down.
  2. Optional: Set the mains switch to the **Off** position.

  For information on locating the controls, refer to SCALEXIO System Mains Controls on page 92.

Method 3

To switch off a DS6001 Processor Board in a SCALEXIO LabBox via the mains controls

1 Press the On/Off button of the SCALEXIO LabBox.

2 Set the mains switch at the rear of the SCALEXIO LabBox to the **Off** position.

For information on locating the controls, refer to SCALEXIO System Mains Controls on page 92.

For more information and for alternative methods to shut down the SCALEXIO LabBox, refer to Powering Up and Shutting Down the LabBox (SCALEXIO Hardware Installation and Configuration 🕮).

Method 4

To switch off a DS6001 Processor Board in a SCALEXIO AutoBox via the remote pin of the power input connector

1 Use the output of the ignition/driving switch (KL15), for example, to change the voltage level at the remote pin 4 of the power input connector from above +6 V DC to a voltage level below +2 V DC. This switches the SCALEXIO AutoBox from normal operating mode to standby mode.

2 Disconnect the SCALEXIO AutoBox from the power supply voltage.

For information on locating the power input connector, refer to SCALEXIO System Mains Controls on page 92.

For more information and for alternative methods to shut down a SCALEXIO AutoBox, refer to Powering Up and Shutting Down SCALEXIO AutoBox (SCALEXIO Hardware Installation and Configuration 🕮).

Result

You switched off the SCALEXIO system with Hypervisor Extension.

Related topics

HowTos

How to Shut Down a Virtual Machine..... 52
Workflow of the System Shutdown Procedure..... 52

Integrating Linux Custom Software Components

Introduction

The SCALEXIO Hypervisor Extension enables you to integrate Linux custom software components into the context of SCALEXIO real-time applications. Communication between Linux applications and SCALEXIO real-time applications is accomplished by integrating hypervisor API functions into the applications during the Linux and ConfigurationDesk build processes.

Where to go from here

Information in this section

Basics on Integrating Linux Custom Software Components.....57

How to Integrate Hypervisor API Functions Into a Custom Linux Application.....58

How to Integrate Hypervisor API Functions Into a SCALEXIO Real-Time Application.....59

Basics on Integrating Linux Custom Software Components

Provided hypervisor API files

To integrate hypervisor API functionality into Linux applications and SCALEXIO real-time applications, hypervisor API files (header and libraries) are required by the Linux compiler and the build process of ConfigurationDesk.

- Hypervisor API files for Linux applications are available after the hypervisor tool package is installed. Refer to How to Add Hypervisor Tools to a Linux Distribution on page 40.

  Additionally, the API files are available in the rth_tools ISO image distributed with the Hypervisor Extension.
- Hypervisor API files for SCALEXIO real-time applications are part of the ConfigurationDesk product set.

Related topics

HowTos

How to Integrate Hypervisor API Functions Into a Custom Linux Application..... 58
How to Integrate Hypervisor API Functions Into a SCALEXIO Real-Time Application..... 59

How to Integrate Hypervisor API Functions Into a Custom Linux Application

Objective

To build a Linux application with integrated hypervisor API functionality, you have to integrate the hypervisor API header and library files.

Precondition

Header files and library files are located on the host PC on which the dSPACE RCP and HIL software package is installed. Refer to Basics on Integrating Linux Custom Software Components on page 57.

Method

To integrate hypervisor API functions into a custom Linux application

1 Include the header and library files as shown in the following code:

```
#include <stdio.h>
#include <dsrth/dsrth.h>
#define PEER1 1
#define SHM_ID 0
int main(int argc, char** argv)
  {
     // Shared Memory is 0 based. Open the first one here
     tShmHandle Handle = dsRthOpen(SHM_ID);
   if (!Handle)
   {
     printf("Error opening Shared Memory with ID %u\n", SHM_ID);
     return 0;
   }
   int* pSharedMem = (int*) dsRthGetShm(Handle);
   printf("Opened VM shared memory of size %x\n", dsRthGetShmSize(Handle));
   printf("Waiting for trigger\n");
   while(1)
   {
      dsRthWaitIrq(Handle, 0);
      // do something with the memory
      printf("Got Value %u\n", *pSharedMem);
      dsRthSendIrq(Handle, PEER1);
   }
   return 0;
  }
```

2 Compile files using the following command:

```
gcc <file name>.c -ldsrth -o <file name>
```

Result

You integrated hypervisor API functions into a custom Linux application by including the hypervisor API files.

Related topics

Basics

Basics on Integrating Linux Custom Software Components.....57

HowTos

How to Integrate Hypervisor API Functions Into a SCALEXIO Real-Time Application.....59

How to Integrate Hypervisor API Functions Into a SCALEXIO Real-Time Application

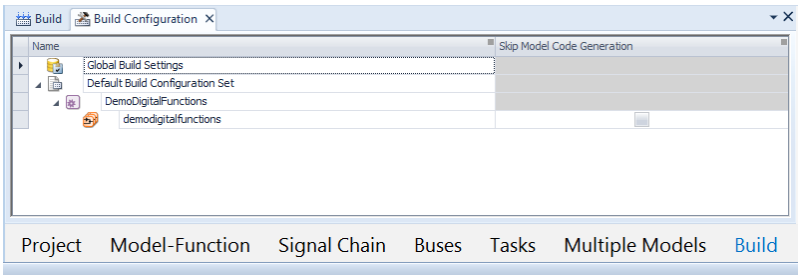
Objective

To build a SCALEXIO real-time application with hypervisor API functionality, you have to integrate the hypervisor API header files and library files into the build process of ConfigurationDesk.

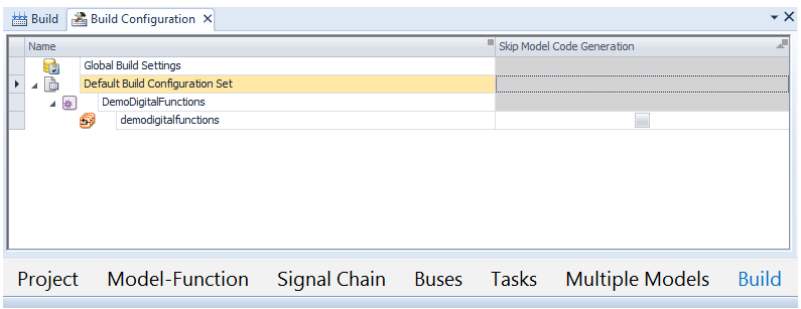
Method

To integrate hypervisor API functions into a SCALEXIO real-time application

1 In the Build view set of ConfigurationDesk, select the Build Configuration tab.

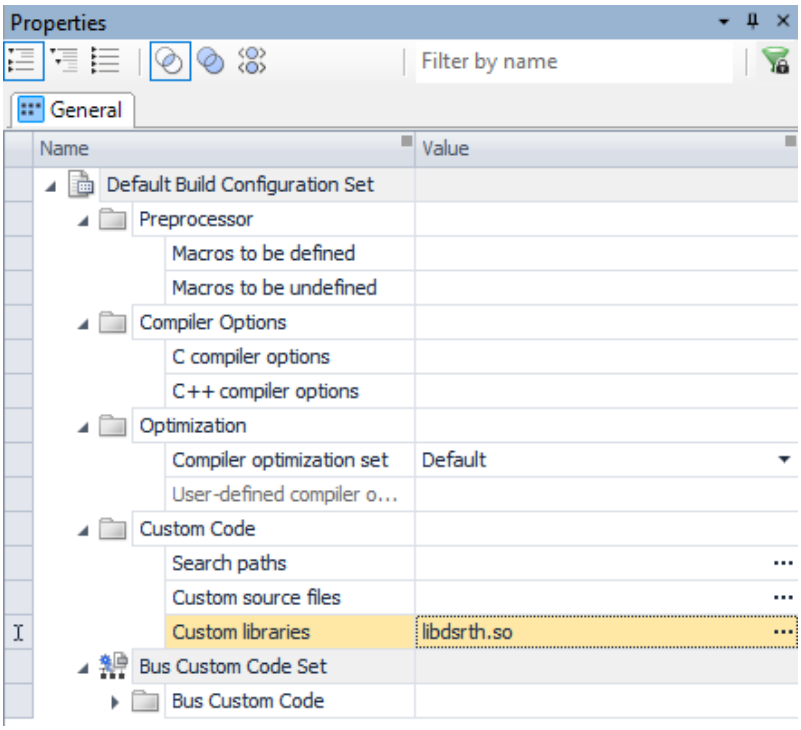


2 Select Default Build Configuration Set.



The properties of the Default Build Configuration Set are displayed in the Properties Browser.

3 In the Properties Browser, enter `libdsrth_host.so` in the Custom libraries edit field.



4 Depending on the location of the hypervisor API files, proceed as follows:
- If the hypervisor API files for SCALEXIO real-time applications are located in `<dSPACE RCP&HIL root directory>\SCALEXIO\include` and `<dSPACE RCP&HIL root directory>\SCALEXIO\lib`, continue with the next step.
- If the hypervisor API files for SCALEXIO real-time applications are located in a directory different from the directories listed above, add the name of the directory with the hypervisor API files to the Search paths edit field of the Properties Browser.

5 Save the ConfigurationDesk application.

Result

You integrated hypervisor API functions into a SCALEXIO real-time application by including the hypervisor API files into a ConfigurationDesk application. Now, the ConfigurationDesk application build process includes the content of the files when you build SCALEXIO real-time applications.

Related topics

Basics

Basics on Integrating Linux Custom Software Components..... 57

HowTos

How to Integrate Hypervisor API Functions Into a Custom Linux Application..... 58

Communication Between Operating Systems

Introduction

Generally, the hypervisor strictly separates operating systems from each other. Nevertheless, communication and data exchange between them are required. The SCALEXIO Hypervisor Extension enables both synchronized and unsynchronized communication between SCALEXIO real-time applications and applications running under Linux. The following methods of communication are provided:

- Virtual Ethernet connections
- Shared memories
- Software interrupts
- Shared file system

Where to go from here

Information in this section

Communication via Virtual Ethernet Connection 63
Communication via Shared Memory 65
Communication via Interrupts 67
Communication via a Shared File System 68

Communication via Virtual Ethernet Connection

Basics on virtual Ethernet connections

The SCALEXIO Hypervisor Extension's virtual Ethernet connection provides the same functionality as standard Ethernet, except for the fact that in the virtual network data is moved through a specific main memory area instead being transferred across a physical network. This is a quick way of moving data between applications running on different operating systems. Using the socket

API, the virtual Ethernet connection handles bidirectional TCP/IP and UDP/IP transmissions.

Hypervisor API files not required If the virtual Ethernet interface is the only applied method of communication between the two operating systems, it is not required to integrate the hypervisor API files into your applications on either operating system. Access to the virtual Ethernet connection is generally provided by the preinstalled socket API.

Synchronized/unsynchronized communication In Linux applications and SCALEXIO real-time applications, use the appropriate socket calls according to the desired communication mode:

Communication Mode	Socket Call Type
Synchronized	Blocking socket calls
Unsynchronized	Non-blocking socket calls

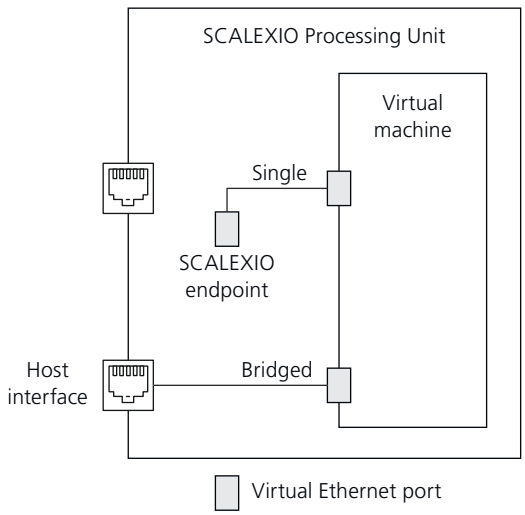
Operation modes of virtual Ethernet devices

Up to ten virtual Ethernet devices can be specified for a SCALEXIO system. They work as single-mode or bridged-mode Ethernet devices.

Single mode In single mode, virtual Ethernet devices can be connected to the SCALEXIO processing hardware by specifying a host_IP address.

Bridged mode In bridged mode, virtual Ethernet devices are routed to the host interface of the SCALEXIO processing hardware and serve as an external Ethernet connection.

Refer to the following illustration.



For configuration details, refer to Configuring the Hypervisor and Virtual Machines on page 69.

Related topics

Basics

Communication via a Shared File System 68
Communication via Interrupts 67
Communication via Shared Memory 65

Communication via Shared Memory

Basics on shared memory

The SCALEXIO Hypervisor Extension offers shared memory to exchange data between applications running on virtual machines and SCALEXIO real-time applications running under the SCALEXIO real-time operating system.

Shared memory size Up to 15 shared memories can be used by a virtual machine. A practical limit is the total amount of physical memory minus the memory that all the operating systems and the hypervisor itself require.

Synchronized/unsynchronized access By default, accessing shared memories is not synchronized between operating systems. To synchronize such operations, use interrupts. Refer to Communication via Interrupts on page 67.

Requirements for using shared memories

The use of shared memories must be prepared as follows:

Specifying shared memories You have to specify shared memories with a unique ID and the size in the configuration files of the virtual machines that use them. Refer to Configuring the Hypervisor and Virtual Machines on page 69.

Adding hypervisor tools You have to add the hypervisor tool package to Linux installations that use shared memories.

Note

The hypervisor tools work only with Ubuntu Linux distributions. If you intend to use shared memory with a Linux distribution other than Ubuntu, contact dSPACE Support (www.dspace.com/go/supportrequest).

For installing the hypervisor tools, refer to How to Add Hypervisor Tools to a Linux Distribution on page 40.

Including hypervisor API files You have to integrate hypervisor API files into SCALEXIO real-time applications and Linux applications. Refer to Integrating Linux Custom Software Components on page 57.

Required start sequence Shared memories are created when the related virtual machine is started.

Note

Start virtual machines first and then start SCALEXIO real-time applications.

Handling shared memory

The following example code shows how to work with shared memory:

```
#include <stdio.h>
#include <dsrth/dsrth.h>
#define PEER1 1
#define SHM_ID 0
int main(int argc, char** argv)
  {
     // Shared memory is zero-based. Open the first one
     tShmHandle Handle = dsRthOpen(SHM_ID);
   if (!Handle)
   {
     printf("Error opening Shared Memory with ID %u\n", SHM_ID);
     return 0;
   }
   int* pSharedMem = (int*) dsRthGetShm(Handle);
   printf("Opened VM shared memory of size %x\n", dsRthGetShmSize(Handle));
   printf("Waiting for trigger\n");
   while(1)
   {
      dsRthWaitIrq(Handle, 0);
      // Do something with the memory
      printf("Got Value %u\n", *pSharedMem);
      dsRthSendIrq(Handle, PEER1);
   }
   return 0;
  }
```

Related topics

Basics

Communication via a Shared File System 68
Communication via Interrupts 67
Communication via Virtual Ethernet Connection 63

Communication via Interrupts

Basics on interrupts

The hypervisor API provided by the SCALEXIO Hypervisor Extension enables you to trigger software interrupts in applications running on the peer operating system.

Interrupts are used for the following purposes:

- To synchronize the execution of applications which implement custom functionality distributed across the two operating systems.
- To notify the peers that new data is available in a shared memory.
- To synchronize shared memory read/write operations.

Note

The following maximum numbers of peers can be used:

- 63 peers for real-time applications per specified shared memory.
- One peer for Linux applications per virtual machine.

Requirements for using interrupts

The us of interrupts must be prepared as follows:

Specifying a shared memory Interrupts are addressed to shared memories. Therefore, you have to specify at least one shared memory.

Adding hypervisor tools You have to add the hypervisor tool package to Linux installations that use interrupts (and therefore at least one shared memory).

Note

The hypervisor tools work only with Ubuntu Linux distributions. If you intend to use interrupts with a Linux distribution other than Ubuntu, contact dSPACE Support (www.dspace.com/go/supportrequest).

For installing the hypervisor tools, refer to How to Add Hypervisor Tools to a Linux Distribution on page 40.

Including hypervisor API files You have to integrate hypervisor API files into SCALEXIO real-time applications and Linux applications. Refer to Integrating Linux Custom Software Components on page 57.

Using interrupts

The following example code shows how to use interrupts:

```
while(1)
{
   dsRthWaitIrq(Handle, 0);
   // Do something with the memory
   printf("Got Value %u\n", *pSharedMem);
   dsRthSendIrq(Handle, PEER1);
}
return 0;
```

Related topics

Basics

Communication via a Shared File System.....68
Communication via Shared Memory.....65
Communication via Virtual Ethernet Connection.....63

Communication via a Shared File System

Overview

A subset of the hypervisor file system on the SCALEXIO Hypervisor Extension boot medium can be shared between SCALEXIO real-time applications and Linux applications.

Linux runs an NFS server, which provides access to the shared folder in the Linux file system. This folder is mounted by the virtual machine during the startup process. Any files stored in the shared folder are accessible to both operating systems, although concurrent modification restrictions might apply when both operating systems attempt to work with the same file at the same time.

For installing NFS share, refer to How to Add NFS share to a Linux Distribution on page 42.

Related topics

Basics

Communication via Interrupts.....67
Communication via Shared Memory.....65
Communication via Virtual Ethernet Connection.....63

Configuring the Hypervisor and Virtual Machines

Introduction

Properties of hypervisor and virtual machines are specified via configuration files. Modifying the configuration might become necessary if system hardware is changed or if applications need new settings, for example, for shared memory or computing power.

Where to go from here

Information in this section

How to Edit a Configuration File 69
Syntax of Configuration Files 70
Configuring a Virtual Machine 72
Configuring the Hypervisor 80

How to Edit a Configuration File

Objective

Modifying the configuration of the hypervisor or of a virtual machine.

Where to find configuration files

Configuration files for the hypervisor and for virtual machines are stored in the `hypervisor` folder of the boot medium:

Configuration File for ...	File Name
Hypervisor	`hypervisor.conf`
Virtual machine	`<file name>.vmconfig`

Refer to How to Access the Hypervisor Extension File System by Windows File Explorer on page 22.

Method

To edit a configuration file

1. Copy the configuration file from the SCALEXIO system to the host PC.
2. Modify the configuration as required with a text editor, such as Notepad. For configuration details, refer to:
   - Configuring a Virtual Machine on page 72
   - Configuring the Hypervisor on page 80
3. Copy the modified configuration file back to the SCALEXIO system.

Next steps

Proceed as follows so that the changes become effective:

- If you modified the configuration file of a virtual machine, start the related virtual machine again.
- If you modified the configuration file of the hypervisor, start the SCALEXIO system again.

Related topics

Basics

Configuring a Virtual Machine....72
Configuring the Hypervisor....80
Syntax of Configuration Files....70

Syntax of Configuration Files

Overview

Configuration files of hypervisor and virtual machines are JSON-based files.

Configuration file structure and entries

The following example shows the structure and entries of configuration files:

```
{
  "name" : "Example_VM_1",
  "id" : 1,
  "autostart" : false,
// Start of the shared memory array
  "shared_memory" :
   [
    {
     "id" : 0,
     "size" : 134217728,
    },
    {
     "id" : 1,
     "size" : 4096,
    }
   ],
```

```
// End of the array
}
```

Refer to the example:

- Configuration files have opening and closing braces (`{}`).
- Entries in the file consist of attribute/value pairs. Attribute and value are separated by a colon (`:`).
- All entries are separated by commas.

Attributes Attributes name the parameter to be specified (name, id, etc.). Attributes are strings. For the properties of strings, refer to the following table.

Values Values specify the parameters and can be strings, numbers, or Boolean values with the following properties:

Data Type	Properties
String	▪ Enclosed in quotation marks. ▪ Only the ASCII characters a…z, A…Z, 0…9, whitespace, hyphen, and underscore are allowed. ▪ Case-sensitive.
Number	Only decimal integers.
Boolean	Valid values: true, false

Arrays Arrays are groups of objects that are specified by more than one parameter, for example, shared memories.

- Arrays are named by strings followed by a colon (`:`), for example `"shared memories" :` .
- Objects are enclosed in braces (`{}`) and contain one or more parameters. Objects are separated by commas.
- All the objects of an array are enclosed in brackets (`[]`).

Comments Comments start at the beginning of a line with a double slash (`//`). Comments that are continued in more than one line require a double slash at the beginning of each line.

Related topics

Basics

Configuring a Virtual Machine....72
Configuring the Hypervisor....80

HowTos

How to Edit a Configuration File....69

Configuring a Virtual Machine

Overview

The configuration of each virtual machine for the SCALEXIO system with Hypervisor Extension is stored in an individual configuration file in the `hypervisor` folder of the boot medium. For accessing and editing refer to How to Edit a Configuration File on page 69.

Virtual machines are configured as follows:

- Specifying name and ID of a virtual machine on page 72
- Assigning resources to a virtual machine on page 72
- Specifying the start behavior of a virtual machine on page 73
- Specifying the virtual graphics adapter for a virtual machine on page 74
- Enabling the VNC server on page 74
- Specifying block devices on page 74
- Specifying shared memories on page 76
- Specifying NFS share on page 77
- Specifying Ethernet devices on page 77
- Passing through devices to a virtual machine on page 78

Specifying name and ID of a virtual machine

You have to specify a unique name and ID for each virtual machine.

"name" The name of a virtual machine is a string with a maximum length of 20 characters. Refer to the following example:

```
"name" : "Example_VM_1",
```

The name must be unique in the SCALEXIO system because it is used as follows:

- For display in the Web interface of the SCALEXIO processing hardware.
- For generating locally administered MAC addresses for Ethernet devices.

"id" The ID of a virtual machine is an integer value [1 … 15]. Refer to the following example:

```
"id" : 1,
```

The ID must be unique on the SCALEXIO system.

Assigning resources to a virtual machine

You have to assign CPU cores and memory to each virtual machine.

"cpus" Assign CPU cores to a virtual machine by an integer number. Refer to the following example:

```
"cpus" : 2,
```

You can assign as many CPU cores to a virtual machine as specified in the hypervisor configuration file (refer to Configuring the Hypervisor on page 80). If you operate more than one virtual machine at a time, the sum of the CPU cores

assigned to all the active virtual machines must not exceed the value specified in the hypervisor configuration file.

"memory" Assign memory to a virtual machine by a string that consists of an integer number followed by *G* or *M* for gigabyte or megabyte. Refer to the following example:

```
"memory" : "2G",
```

You can assign as much memory to a virtual machine as the SCALEXIO processing hardware provides, minus at least two gigabytes, which must be reserved for SCALEXIO real-time applications. If you operate more than one virtual machine at a time, the sum of the memory assigned to all the active virtual machines must not exceed the calculated maximum value.

Specifying the start behavior of a virtual machine

You can specify whether a virtual machine starts automatically with the SCALEXIO system or has to be started by the user. Furthermore, you can specify the boot mode and whether a boot menu can be accessed or not.

"autostart" The value is `true` for autostart or `false` for manual start. Refer to the following example:

```
"autostart" : false,
```

"boot_menu" You can specify whether the boot menu can be accessed during startup. The value is `true` or `false`. Refer to the following example:

```
"boot_menu" : true,
```

"boot_delay" You can specify a time span (in seconds) in which booting is halted to let you enter the boot menu. You can enter the boot menu by pressing Esc in `bios` boot mode or F2 in `efi` boot mode. The value is an integer number. Refer to the following example:

```
"boot_delay" : 2,
```

"boot_mode" You can specify the virtual machine to boot in `efi` mode (recommended) or in `bios` mode. The value is a string. Refer to the following example:

```
"boot_mode" : "efi",
```

The boot mode must not be changed after a Linux distribution is installed on the particular virtual machine.

"realtime" You can determine that CPU cores assigned to the virtual machine run in real-time. Specify `1` to activate and `0` to deactivate real-time operation for the virtual machine's CPU cores. The value is an integer number. Refer to the following example:

```
"realtime" : 1,
```

The real-time mode ensures that CPU cores assigned to the virtual machine have a higher priority than other activities on the CPU of the SCALEXIO processing hardware.

Note

By default, the first CPU core assigned to the virtual machine cannot be run in real-time mode because it is used for serving I/O threads. To move the serving of I/O threads to the CPU service core, refer to Configuring the Hypervisor on page 80.

Specifying the virtual graphics adapter for a virtual machine

"vga" You can specify one of the following graphics adapters to be emulated for the virtual machine:

- `virtio`: For access via a VirtIO interface.
- `vmware`: Emulates an SVGA graphics adapter.
- `cirrus`: Emulates a Cirrus graphics adapter.
- `none`: Recommended for real-time co-simulation.

The VGA mode value is a string. Refer to the following example:

```
"vga" : "virtio",
```

Note

When passing through a real graphics adapter to a virtual machine, you do not have to specify a virtual graphics adapter. Refer to Passing through devices to a virtual machine on page 78.

Enabling the VNC server

You can enable the hypervisor's VNC server for controlling the virtual machine remotely via a tool such as UltraVNC Viewer.

"vnc" You can specify whether the VNC server is enabled. The value is `true` or `false`. Refer to the following example:

```
"vnc" : true,
```

vnc_keymap You can specify either German or English keyboard mapping for the VNC server. The value is a string (`de` or `en-us`). Refer to the following example:

```
"vnc_keymap" : "de",
```

Specifying block devices

You have to specify block devices for a virtual machine.

"block_devices" Declares the array that includes all specified block devices. Block devices have the following attributes:

"file" You have to specify file name and extension for the block device by a string.

"type" You have to specify the media type that is represented by the block device, for example, `hd`, `raw`, or `cdrom`. The value is a string.

> **Note**
>
> Block devices of `raw` and `hd` type behave identical. However, a raw-type file is not interpreted, for example, as a *qcow* image file.

"size" You can specify the size of the file that represents a block device, for example, a hard disk, in megabyte. The value is an integer number. If the size is not specified, the default value of 40,000 MB applies.

"thin_provisioning" You can specify whether the file that represents a component is created with the size as specified at the **size** attribute (thin_provisoning = `false`) or whether the created file is only as large as actually required, expandable up to the value specified at the **size** attribute (thin_provisoning = `true`).

> **Note**
>
> Use `thin_provisioning` only if required because it increases the likelihood of a data loss.

"virtio" Specifies whether the file can be accessed via a para-virtualized interface, which must be supported by the installed Linux distribution. The value can be `true` or `false`.

> **Tip**
>
> If possible, use the para-virtualized interface, which has a much better read/write performance for the device. If `"virtio"` is set to `false`, the device will be emulated as a standard IDE device.

Required block devices You have to specify at least the following block devices:

- A hard disk for the operating system and applications. The file format for a newly created block device is always *qcow2*. Refer to the following example:

```
{
   "file" : "hd1.qcow2",
   "type" : "hd",
   "size" : 10000,
   "thin_provisioning" : true,
   "virtio" : true,
},
```

  This entry lets the system create the file `hd1.qcow2` in the `hypervisor` folder of the boot medium.

  You can let the system create the file on an optionally installed SSD (user storage). In this case, you have to specify the file name as follows:

```
   "file" : "%userstorage<n>%/hd1.qcow2",
```

where *n* is an integer number.

NOTICE

Storing virtual machine data on a FAT32-formatted SSD

If the optional SSD is FAT32-formatted, you might lose stored data if the SCALEXIO system is not properly shut down.

- Shut down the SCALEXIO system properly. Refer to Shutting Down a SCALEXIO System with Hypervisor Extension on page 51.

- For installing a Linux distribution, you have to specify a CD-ROM drive from which the operating system is installed. The CD-ROM drive must reference the ISO image of the operation system that you want to install. The ISO image must be located in the `hypervisor` folder of the boot medium.

  The following example shows the block devices required for installing the Linux distribution Ubuntu 18.04.5:

```
"block_devices" :
 [
   {
      "file" : "hd1.qcow2",
      "type" : "hd",
      "size" : 10000,
      "virtio" : true,
   },
   {
      "file" : "ubuntu-18.04.5-desktop-amd64.iso",
      "type" : "cdrom",
      "virtio" : true,
   },
 ],
```

  The order of the block devices in the array determines the boot order of the virtual machine.

Tip

After the Linux distribution is installed, you can remove the CD-ROM block device from the configuration file.

Specifying shared memories

You can specify shared memories for data exchange between operating systems. All shared memories on a SCALEXIO system must have unique IDs. A virtual machine can use up to 15 shared memories.

For a shared memory that is used by more than one virtual machine, the following applies:

- It must be specified with its unique ID in each configuration file.
- Its size must be specified identically in all configuration files.

"shared_memory" Declares the array that includes all shared memories used by this virtual machine. Each shared memory has the following attributes:

"id" You have to specify the ID of a shared memory by an integer number.

"size" You have to specify the size of a shared memory in bytes by an integer number.

Refer to the following example:

```
"shared_memory" :
 [
   {
      "id" : 0,
      "size" : 134217728,
  },
  {
      "id" : 1,
      "size" : 4096,
   }
 ],
```

Note

To install the hypervisor tools that are required for using shared memories, add the following block device to the `block devices` array:

```
  {
    "file" : "%rth-tools%",
    "type" : "cdrom",
    "virtio" : true,
  },
```

Refer to Specifying block devices on page 74.

Specifying NFS share

You can specify one network file system (NFS) for a virtual machine, which is automatically mounted during the boot process.

"network_share" You have to specify an IP address and the path to the shared files for mounting the NFS. The value is a string.

"network_share_timeout" You have to specify a time span (in seconds), after which the mounting is aborted. The value is an integer number.

Refer to the following example:

```
 "network_share" : "192.168.10.2:/home/<user name>/share",
 "network_share_timeout" : 20,
```

For information on adding the NFS share to the virtual machine, refer to How to Add NFS share to a Linux Distribution on page 42.

Specifying Ethernet devices

You can specify up to ten Ethernet devices per SCALEXIO system.

"ethernet_devices" Declares the array that includes all the specified Ethernet devices for this virtual machine. Each Ethernet device has the following attributes:

"mac" You can specify the MAC address for the Ethernet device by a string. If you do not specify the MAC address, the system generates a *locally administered* MAC address based on the name specified for the virtual machine.

"mode" An Ethernet device can be either routed to the host interface of the SCALEXIO system (`bridged` mode) or to a virtual end point (`single` mode). The value (`bridged` or `single`) is a string.

"bridge_master" Specifies the Ethernet port of the SCALEXIO system to which a bridged Ethernet device is routed. The value is a string and must be `HIF`.

host_ip You have to specify the IP address of the SCALEXIO end point for a single-mode Ethernet connection by a string.

Refer to the following example:

```
"ethernet_devices" :
 [
   {
      "mac" : "00:01:02:03:04:05",
      "mode" : "bridged",
      "bridge_master" : "HIF"
   },
   {
      "mac" : "00:01:02:03:04:06",
      "mode" : "single",
      "host_ip" : "192.168.10.1",
   }
 ],
```

For more information, refer to Communication via Virtual Ethernet Connection on page 63.

Passing through devices to a virtual machine

You can pass through devices so they can be used exclusively by the virtual machine.

"devices" Declares an array that includes all devices to be passed through to the virtual machine. The devices have the following attributes:

"type" The following device types can be passed through:

- `IGPU`: The graphics processing unit of the mainboard of the SCALEXIO Processing Unit or the DS6001 Processor Board.
- `USBn`: USB controllers
- `PCI Device`: Custom PCI devices
- `Ethernet Channel`: Ethernet channels (identified by name)

"name" The device name. For Ethernet channels, the name is specified in ConfigurationDesk.

"addr" The particular device address (only for PCI devices)

Note

For finding out the PCI addresses of the custom PCI devices of your SCALEXIO system, use an Internet browser running on the host PC. Depending on the SCALEXIO processing hardware, enter one of the following addresses in the address edit field of the Internet browser:

- `http://<SCALEXIO Processing Unit IP>/log/lspci`
- `http://<DS6001 Processor Board IP>/log/cn/lspci`

All values in the array are strings. Refer to the following example:

```
"devices" : [
   {
      "type" : "IGPU",
      "name" : "Integrated Graphics Device",
   },
   {
      "type" : "USB1",
      "name" : "USB Host Controller"
   },
   {
      "type" : "USB2",
      "name" : "USB Host Controller #2"
   },
   {
      "type" : "PCI Device",
      "name" : "My Custom PCI Device",
      "addr" : "02:00.0"
   },
   {
      "type" : "Ethernet Channel",
      "name" : "Eth1_1"
   },
   {
      "type" : "Ethernet Channel",
      "name" : "Eth1_2"
   },
 ],
```

Related topics

Basics

Configuring the Hypervisor.....80
Syntax of Configuration Files.....70

HowTos

How to Edit a Configuration File.....69

Configuring the Hypervisor

Overview

Configurable properties of the SCALEXIO hypervisor are stored in the `hypervisor.conf` configuration file (JSON based file) in the `hypervisor` folder of the boot medium.

The hypervisor can be configured as follows:

- Assigning CPU cores to the hypervisor on page 80
- Specifying the CPU core for serving I/O threads on page 81

Assigning CPU cores to the hypervisor

CPU power must be assigned to the hypervisor for the virtual machines running on the SCALEXIO system with Hypervisor Extension.

`"num_cpus"` CPU cores are assigned to the hypervisor by an integer value so they can be used by virtual machines. Refer to the following example:

```
"num_cpus" : 2
```

Note

- At least two CPU cores must be available for SCALEXIO real-time applications.
- If you do not assign a CPU core to the hypervisor (`"num_cpus" : 0`), the hypervisor is deactivated and you can use the SCALEXIO system like a SCALEXIO system without hypervisor extension.

The following table shows how many cores can be assigned to the hypervisor depending on the SCALEXIO processing hardware used:

SCALEXIO Processing Hardware	Number of Assignable CPU Cores
SCALEXIO Processing Unit with E5-2640v3 processor	6
SCALEXIO Processing Unit with E3-1275v6 processor	2

SCALEXIO Processing Hardware	Number of Assignable CPU Cores
SCALEXIO Processing Unit with Intel 6208U processor	14
DS6001 Processor Board	3

Specifying the CPU core for serving I/O threads

For SCALEXIO Processing Units you can specify, whether I/O threads are served by the first CPU core assigned to the virtual machine (default) or by the CPU service core.

"allowIOonServiceCore" Specify `1` for the CPU service core or `0` for the first CPU core assigned to the virtual machine to serve I/O threads. The value is an integer number. Refer to the following example:

```
"allowIOonServiceCore" : 1
```

Note

- For DS6001 Processor Boards, the specification has no effect. I/O threads are always served by the first CPU core assigned to the virtual machine.
- High loads of I/O threads on the CPU service core can compromise real-time services such as DAQ or XCP.
- If the CPU service core is specified to serve I/O threads, the first CPU core of the virtual machine can be specified to run in real-time mode. Refer to "realtime" on page 73.

Related topics

Basics

Configuring a Virtual Machine 72
Syntax of Configuration Files 70

HowTos

How to Edit a Configuration File 69

Backing Up and Restoring Virtual Machines

Procedures for Backing Up and Restoring a Virtual Machine

Overview

Each virtual machine for the SCALEXIO system with Hypervisor Extension consists at least of two files:

- A configuration file: `<file name>.vmconfig` .
- A file for the operating system and data: `<file name>.qcow2` .

  This file is specified in the configuration file as an `hd` block device. If required, refer to Configuring a Virtual Machine on page 72.

Both files are located in the `hypervisor` folder of the SCALEXIO Hypervisor boot medium.

Software tool To back up or restore these files, you have to use an FTP program such as Windows File Explorer. For more information, refer to How to Access the Hypervisor Extension File System by Windows File Explorer on page 22.

Backing up a virtual machine

To back up a virtual machine, you have to copy its configuration file and its hard disk file to a backup location of your choice, for example, to the host PC.

Restoring a virtual machine

To restore a virtual machine, you have to copy its backed up configuration file and hard disk file to the `hypervisor` folder of the SCALEXIO Hypervisor boot medium.

After restoring the files, you can access the SCALEXIO processing hardware Web interface and start the virtual machine so you can work with it. Refer to

- How to Access the SCALEXIO Web Interface and Virtual Machines by an Internet Browser on page 23.
- How to Start a Virtual Machine on page 47.

Modifying or Replacing a Linux Distribution

Procedures for Modifying or Replacing a Linux Distribution

Introduction

You can modify an installed Linux distribution by adding software or packages. If required, you can also replace an entire Linux distribution of a virtual machine.

Modifying a Linux distribution

Adding and installing software or packages to a Linux distribution on the SCALEXIO system with Hypervisor Extension is performed the same way as for Linux distributions on standard PCs. For examples, refer to:

- How to Add Hypervisor Tools to a Linux Distribution on page 40.
- How to Add an SSH Server to a Linux Distribution on page 39.

Replacing a Linux distribution

To replace a Linux distribution of a virtual machine, you are recommended to delete its hard disk file with the operating system and data (<file name>.qcow2) in the `hypervisor` folder of the boot medium.

> **Tip**
>
> This file is specified in the configuration file as an `hd` block device. If required, refer to Configuring a Virtual Machine on page 72.

After deleting the file, you can install a new Linux distribution. For more information on the steps to perform, refer to:

- How to Access the Hypervisor Extension File System by Windows File Explorer on page 22.
- How to Install a Linux Distribution on a Virtual Machine on page 37.

Appendix

Where to go from here

Information in this section

SCALEXIO System Connectors for External Hardware.....87
SCALEXIO System Mains Controls.....92
Virtual Machine Configuration File.....93
Hypervisor Configuration File.....96

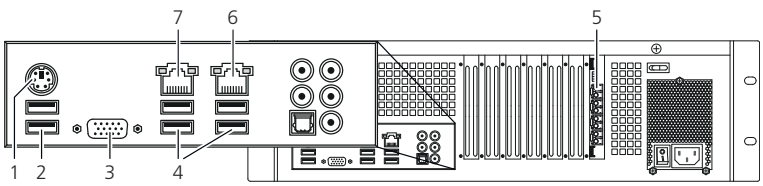
SCALEXIO System Connectors for External Hardware

Introduction

To locate and identify the connectors for external hardware of the SCALEXIO system with Hypervisor Extension, refer to the following illustrations.

SCALEXIO Processing Unit with E5-2640v3 processor

Rear panel connectors

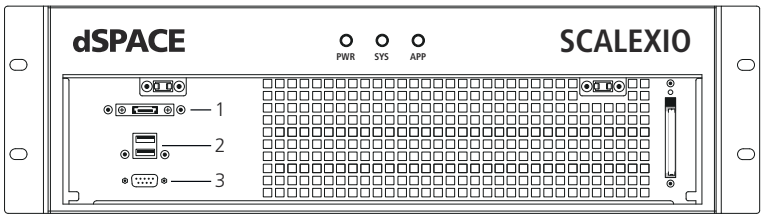


Connector		Default Name in the Configuration File
1	PS/2 keyboard/mouse port	–
2	USB 3.0 port	USB2
3	VGA port for local console output	–
4	USB 3.0 port	USB1

Connector		Default Name in the Configuration File
5	IOCNET (DS2502 IOCNET Link Board)	–
6	Second Ethernet port (Gigabit Ethernet port)	Eth0_1[1)]
7	Host interface (Gigabit Ethernet port)	HIF

1) If you rename an Ethernet port in ConfigurationDesk, you also have to rename it in the configuration file of the virtual machine.

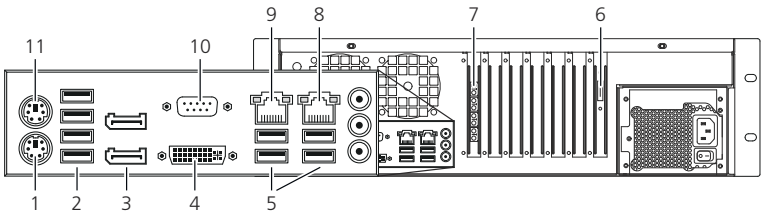
Front panel connectors



Connector		Default Name in the Configuration File
1	Not used	–
2	USB 2.0 port	USB3
3	RS232 port (COM1, 0x3F8, IRQ 4)	–

SCALEXIO Processing Unit with E3-1275v6 processor

Rear panel connectors

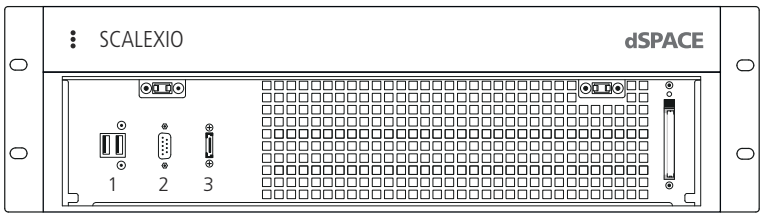


Connector		Default Name in the Configuration File
1	PS/2 keyboard port	–
2	USB 2.0 port	USB1
3	DisplayPort V1.2 for local console output	–
4	DVI-D port	–
5	USB 3.0 port	USB1
6	Not used	–
7	IOCNET (DS2502 IOCNET Link Board)	–
8	Host interface (Gigabit Ethernet port)	HIF
9	Second Ethernet port (Gigabit Ethernet port)	Eth0_1[1)]

Connector		Default Name in the Configuration File
10	RS232 port (COM1, 0x3F8, IRQ 4)	–
11	PS/2 mouse port	–

[1] If you rename an Ethernet port in ConfigurationDesk, you also have to rename it in the configuration file of the virtual machine.

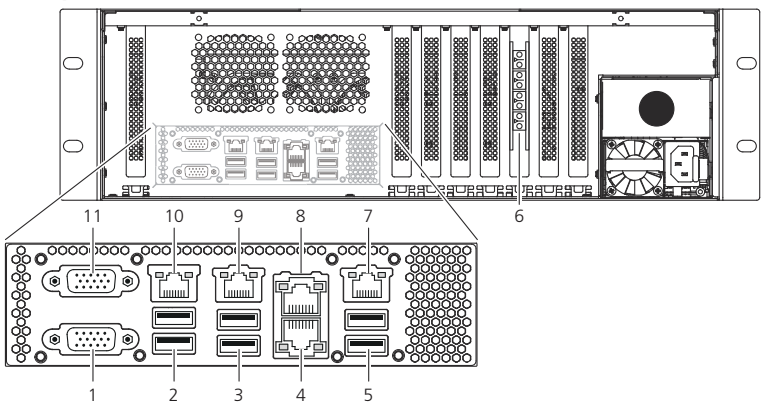
Front panel connectors



Connector		Default Name in the Configuration File
1	USB 3.0 port	USB1
2	RS232 port (COM2, 0x2F8, IRQ 3)	–
3	Not used	–

SCALEXIO Processing Unit with Intel 6208U processor

Rear panel connectors

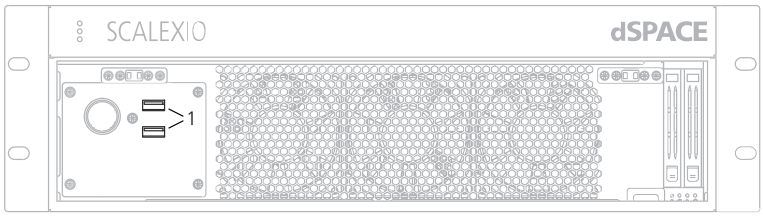


Connector		Default Name in the Configuration File
1	VGA port for local console output	–
2	USB 2.0 port	USB1
3	USB 3.0 port	USB1
4	Eth0_3 (10 Gigabit Ethernet port)	Eth0_3[1]
5	USB 3.0 port	USB1
6	IOCNET (DS2502 IOCNET Link Board)	–

Connector		Default Name in the Configuration File
7	Not used	–
8	Host interface (10 Gigabit Ethernet port)	HIF
9	Eth0_2 (Gigabit Ethernet port)	Eth0_2[1)]
10	Eth0_1 (Gigabit Ethernet port)	Eth0_1[1)]
11	RS232 port (COM1, 0x3F8, IRQ 4)	–

[1)] If you rename an Ethernet port in ConfigurationDesk, you also have to rename it in the configuration file of the virtual machine.

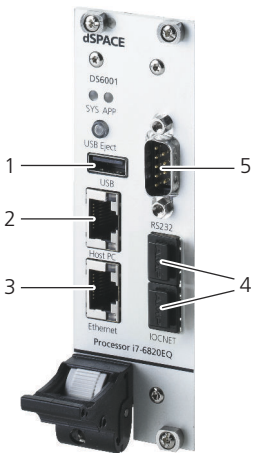
Front panel connectors



Connector		Default Name in the Configuration File
1	USB 3.0 port	USB1

DS6001 Processor Board

Front panel connectors



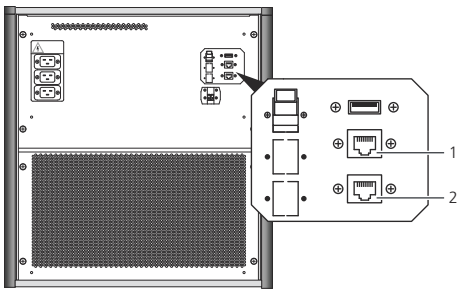
Connector		Default Name in the Configuration File
1	USB 2.0 port	USB1
2	Host interface (Gigabit Ethernet port)	HIF
3	Ethernet (Gigabit Ethernet port)	Eth0_1[1)]
4	IOCNET	–
5	RS232 port (COM1, 0x3F8, IRQ 4)	–

[1)] If you rename an Ethernet port in ConfigurationDesk, you also have to rename it in the configuration file of the virtual machine.

Connectors on SCALEXIO off-the-shelf rack systems

On SCALEXIO off-the-shelf rack systems (9 U or 12 U), the two Ethernet ports of the Processing Unit are extended to a connector panel at the rear of the rack. dSPACE offers custom connector panels on demand to extend more connectors from the mainboard connector panel to the rear of the rack.

The following illustration shows the standard version of the rear connector panel of a SCALEXIO off-the-shelf rack system.



Connector		Default Name in the Configuration File
1	Host interface (Gigabit Ethernet port)	HIF
2	Second Ethernet port (Gigabit Ethernet port)	Eth0_1[1)]

[1)] If you rename an Ethernet port in ConfigurationDesk, you also have to rename it in the configuration file of the virtual machine.

Connectors on SCALEXIO custom rack systems

Refer to the documentation of the respective SCALEXIO custom rack system.

Related topics

Basics

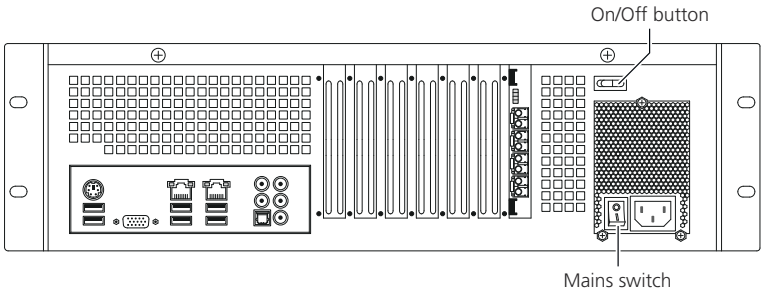
SCALEXIO System Mains Controls.....92

SCALEXIO System Mains Controls

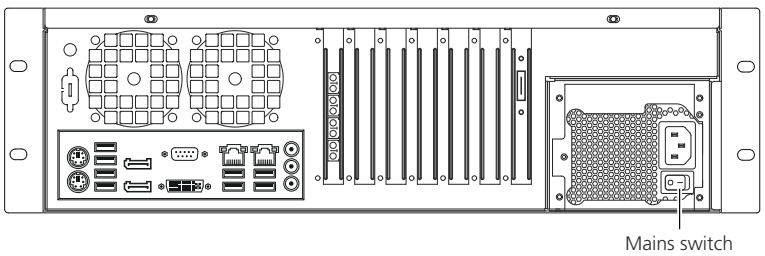
Introduction

To locate and identify the mains controls of the SCALEXIO system with Hypervisor Extension, refer to the following illustrations.

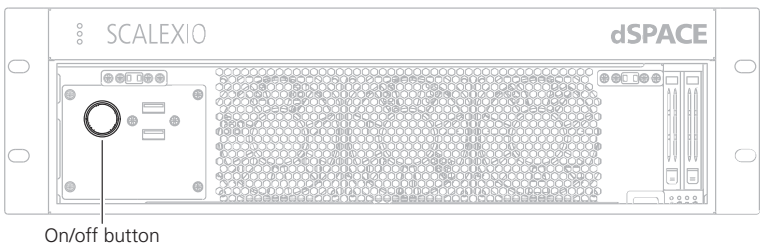
SCALEXIO Processing Unit with E5-2640v3 processor



SCALEXIO Processing Unit with E3-1275v6 processor



SCALEXIO Processing Unit with Intel 6208U processor

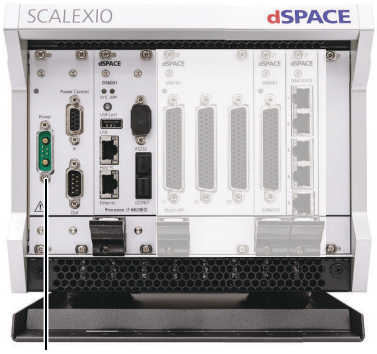


SCALEXIO LabBox with DS6001 Processor Board

On/off button with status LED

Mains switch

SCALEXIO AutoBox with DS6001 Processor Board



Power input connector with remote pin

Related topics

Basics

SCALEXIO System Connectors for External Hardware.....87

HowTos

How to Power Up and Boot a SCALEXIO System with Hypervisor Extension.....45
How to Switch Off a SCALEXIO System with Hypervisor Extension.....54

Virtual Machine Configuration File

Example configuration

The `template.vmconf` configuration file is located in the `hypervisor` folder on the boot medium of the SCALEXIO Hypervisor Extension. It can be used as a template to create and configure your own configuration files. Refer to How to Create a Virtual Machine on page 36 and Configuring the Hypervisor and Virtual Machines on page 69.

```
{
// Name of the VM. Max 20 chars
```

```
"name" : "Example_VM_1",
// Unique ID of the VM on this platform [0..15]
"id" : 1,
// Start VM on boot?
"autostart" : false,
// Number of CPUs assigned to this VM
"cpus" : 2,
// Memory assigned to this VM
"memory" : "2G",
// Block devices to provide to VM
"block_devices" : [ {
"file" : "hd1.qcow2",
"type" : "hd",
// Size of the hard disk in MB. If the file does not exist,
// it will be created
"Size" : "40000",
"virtio" : true,
},
{
"file" : "ubuntu-18.04.5-desktop-amd64.iso",
"type" : "cdrom",
"virtio" : true,
},
// {
// "file" : "extra.img",
// "type" : "raw",
// "virtio" : false,
// }
],
// Show boot menu?
"boot_menu" : true,
// Boot menu delay
"boot_delay" : 2,
// Boot mode: either "efi" or "bios"
"Boot_mode" : "efi",
// VGA mode to use: virtio, vmware, cirrus, none
// Use none for realtime co-simulation
"vga" : "virtio",
// Enable VNC server?
"vnc" : true,
// Keymap to use for VNC server , "de" or "en-us"
"vnc_keymap" : "de",
// Shared memory configuration
// Up to 15 shared memories may be used by a VM
"shared_memory" : [
{
// Unique ID of this shared memory on this platform
"id" : 0,
// Size in bytes of the shared memory region
"size" : 134217728,
},
{
"id" : 1,
"size" : 4096,
}
],
// Specify NFS share to be auto-mounted for this VM
// "network_share" : "192.168.10.2:/home/<user name>/share",
// "network_share_timeout" : 20,
// Ethernet device description
// Maximum of 10 Ethernet devices supported
```

```
"ethernet_devices" : [
{
// MAC to be assigned to this Ethernet device
// If none is specified, a unique MAC address will be generated
// for this VM based on the name of the VM
//"mac" : "00:01:02:03:04:05",
"mode" : "bridged",
"bridge_master" : "HIF"
},
{ "mac" : "00:01:02:03:04:06",
"mode" : "single",
// For single-mode connections SCALEXIO <-> VM, we can assign
// an IP address for the SCALEXIO endpoint here
"host_ip" : "192.168.10.1",
}
],
// Devices to pass exclusively to the VM
"devices" : [
//
{
// "type" : "IGPU",
// "name" : "Integrated Graphics Device",
// },
// {
// "type" : "USB1",
// "name" : "USB Host Controller"
// },
// {
// "type" : "USB2",
// "name" : "USB Host Controller #2"
// },
// {
// "type" : "PCI Device",
// "name" : "My Custom PCI Device",
// "addr" : "02:00.0"
// },
// {
// // Pass through Ethernet channels by name
// "type" : "Ethernet Channel",
// "name" : "Eth1_1"
// },
],
}
```

Related topics

Basics

Configuring the Hypervisor and Virtual Machines 69
Hypervisor Configuration File 96

Hypervisor Configuration File

Hypervisor configuration file

The `hypervisor.conf` configuration file is located in the `hypervisor` folder on the boot medium of the SCALEXIO Hypervisor Extension. It can be modified if required. Refer to Configuring the Hypervisor and Virtual Machines on page 69.

```
{
"num_cpus": 2
}
```

Related topics

Basics

Configuring the Hypervisor and Virtual Machines..... 69
Virtual Machine Configuration File..... 93

A

a virtual machine on the SCALEXIO system
  accessing via virtual network control 26
accessing a SCALEXIO system with Hypervisor Extension 21
accessing a virtual machine on the SCALEXIO system via an SSH connection 24
accessing a virtual machine on the SCALEXIO system via virtual network control 26
accessing a virtual machine on the SCALEXIO system via VNC 26
accessing the Hypervisor Extension file system by Windows File Explorer 22
accessing the SCALEXIO processing hardware Web interface by Internet browser 23

B

backing up the virtual machine 83
basics
  interrupts 67
  shared memory 65
boot medium
  installing 32
booting a virtual machine 48
booting the SCALEXIO system with Hypervisor Extension 45

C

Common Program Data folder 8
communication
  via interrupts 67
  via shared file system 68
  via shared memory 65
  via virtual Ethernet connection 63
connectors for external hardware 87
creating a virtual machine 36
custom Linux application
  integrating hypervisor API functions 58

D

Documents folder 8

F

features
  SCALEXIO Hypervisor Extension 16

H

hardware
  required for the SCALEXIO Hypervisor Extension 17
hypervisor
  configuring 69
Hypervisor Extension file system
  accessing by Windows File Explorer 22
hypervisor tools
  adding to the Linux distribution 40

I

installation workflow
  SCALEXIO system with Hypervisor Extension 31
installing a Linux distribution on a virtual machine 37
installing the SCALEXIO Hypervisor Extension 31
installing the SCALEXIO Hypervisor Extension boot medium 32
integrating hypervisor API functions
  into a custom Linux application 58
  into a SCALEXIO real-time application 59
integrating Linux custom software components 57
interrupts
  basics 67
  communication via 67
  triggering 67
  using 67
  waiting for 67

L

Linux custom software components
  integrating 57
Linux distribution
  adding an SSH server 39
  adding hypervisor tools 40
  adding NFS share 42
  installing on a virtual machine 37
  modifying 85
  replacing 85
Linux operating system
  shutting down 52
Local Program Data folder 8

M

mains controls 92
modifying the Linux distribution 85

N

NFS share
  adding to the Linux distribution 42

O

overview
  SCALEXIO Hypervisor Extension 13

P

powering up the SCALEXIO system with Hypervisor Extension 45

R

real-time Linux
  introduction 19
replacing the Linux distribution 85
requirements
  SCALEXIO Hypervisor Extension 17
restoring a virtual machine 83

S

safety precautions
  for working with the SCALEXIO Hypervisor Extension 12
SCALEXIO Hypervisor Extension
  features 16
  installing 31
  optional software 18
  overview 13
  required hardware 17
  required software 18
  requirements 17
  safety precautions for working with 12
SCALEXIO Hypervisor Extension boot medium
  installing 32
SCALEXIO processing hardware Web interface
  accessing by Internet browser 23
SCALEXIO real-time application
  integrating hypervisor API functions 59
SCALEXIO system with Hypervisor Extension
  accessing 21
  booting 45
  installation workflow 31
  powering up 45
  shutting down 51
  starting 45
  switching off 54
shared file system
  communication via 68
shared memory
  basics 65
  communication via 65
  size 65
shutting down a SCALEXIO system with Hypervisor Extension 51
shutting down a virtual machine 52
shutting down the Linux operating system 52
size
  shared memory 65
software
  required for the SCALEXIO Hypervisor Extension 18
SSH server
  adding to the Linux distribution 39
starting
  virtual machine 47
starting a SCALEXIO system with Hypervisor Extension 45
switching off a SCALEXIO system with Hypervisor Extension 54
system shutdown procedure
  workflow 52

U

using interrupts 67

V

virtual Ethernet connection
  communication via 63
virtual machine
  backing up 83
  booting 48
  creating 36
  installing a Linux distribution 37
  restoring 83
  shutting down 52
  starting 47
Virtual machine on the SCALEXIO system
  accessing via an SSH connection 24
virtual machines
  configuring 69

W

workflow
  system shutdown procedure 52