

acontis technologies GmbH

SOFTWARE

Hypervisor-Quickstart-Guide

acontis Real-time Hypervisor Quickstart

Version 8.x

Edition: November 21, 2023

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

© Copyright acontis technologies GmbH

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



Table of Contents

| 1 | Getting Started - step-by-step tutorial | | | | | | | | |
|----|--|---|--|--|--|--|--|--|--|
| | 1.1 | Prerequisites | | | | | | | |
| | 1.2 | Update EtherCAT package 5 | | | | | | | |
| 2 | Inst | allation and basic configuration 6 | | | | | | | |
| | 2.1 | USB stick boot media creation | | | | | | | |
| | 2.2 | Installation | | | | | | | |
| | 2.3 | Remote Access to the Hypervisor using Remote Desktop Connection 8 | | | | | | | |
| | 2.4 | Basic system configuration | | | | | | | |
| | 2.5 | Validate the configuration | | | | | | | |
| 3 | Ethe | ernet Device Assignment 17 | | | | | | | |
| | 3.1 | Device Identification | | | | | | | |
| | 3.2 | Device Assignment | | | | | | | |
| | 3.3 | RTOS Assignment | | | | | | | |
| | 3.4 | Device Verification | | | | | | | |
| | 3.5 | Adjust the configuration | | | | | | | |
| | 3.6 | Reboot Hypervisor Host | | | | | | | |
| 4 | Run | ning the acontis EtherCAT Master Demo (RT-Linux) 21 | | | | | | | |
| | 4.1 | Adjust the configuration | | | | | | | |
| | 4.2 | Step 1: Scan the EtherCAT network | | | | | | | |
| | 4.3 | Step 2: Set the EtherCAT network operational 21 | | | | | | | |
| Th | This Ouickstart Guide shows how to install the acontis Hypervisor and how to run an EtherCAT demo. | | | | | | | | |

This Quickstart Guide shows how to install the acontis Hypervisor and how to run an EtherCAT demo application.



1 Getting Started - step-by-step tutorial

This manual will guide you through the following steps:

- Hypervisor Hardware/Software setup
- Assigning an Intel PRO/1000 network adapter for real-time use (connected with EtherCAT slaves).
- [OPTIONAL]: Scan and configure the EtherCAT network.
- [OPTIONAL]: Set an EtherCAT network into **OPERATIONAL** mode.

Hint: sudo is similar to Windows run as administrator and is valid for a single command only.

Hint: sudo -s switches the user to root (administrator), all following commands then will be executed with elevated rights.

1.1 Prerequisites

- PC/IPC with at least 1 Ethernet adapter.
 - CPU: at least 2 cores. This guide assumes a 4 cores CPU is used.
 - SSD/HDD: at least 10GB. Additional 50GB in case a Windows guest shall be used.
 - One Ethernet adapter to be used for TCP/IP (e.g. to remotely connect to the Hypervisor). This adapter has to be connected with your company network.
 - One additional Intel Gigabit Ethernet adapter (like I210), as some of the shipped demo scripts are preconfigured for this type of network adapter.
- Several EtherCAT slaves connected with the <code>Intel Gigabit</code> network adapter

Caution: You must disable secure boot in the BIOS, otherwise the RTOSVisor cannot be installed or will not work correctly!

Caution: In this tutorial the Hypervisor will be **installed** onto the hard disk and **all** data will be **overwritten**. Assure required data **is saved** before installation!

- Recommended BIOS settings (for real-time operation)
 - USB Legacy mode has to be *disabled*
 - Hyper-Threading shall be disabled
 - Power saving settings have to be *disabled* (C-States, Speedstep, ...)



1.2 Update EtherCAT package

Caution: This section is only needed if an EtherCAT network will be used!

The hypervisor package contains **yet** an EtherCAT master package. In **rare** cases an *updated* EtherCAT master package is received along *this* hypervisor package.

The following steps are needed in such a case:

- Real-time Linux:
 - 32bit: Copy from EC-Master-V3.*.*.+EC-WinRT-Linux-x86_32Bit-SDK_Protected.tar\Bin\Linux\x86* to /hv/guests/etc/rt-linux/ files/ecat/
 - 64bit: Copy from EC-Master-V3.*.*.+EC-WinRT-Linux-x86_64Bit-SDK_Protected.tar\Bin\Linux\x64* to /hv/guests/etc/rt-linux/ files/ecat/
- On Time RTOS-32:

Copy from EC-Master-V3.*.*.*RTOS-32-x86_32Bit-SDK_Protected.zip\ Bin\RTOS-32\x86* to /hv/guests/etc/rtos-32/rtfiles/ecmasterdemo/



2 Installation and basic configuration

The Hypervisor has to be installed onto an empty installation media. It is directly booted from the BIOS. Side-by-side installation with an existing Windows or Linux is not recommended, but possible. If an existing Windows or Linux system shall be kept, assure a free partition for booting the Hypervisor is available. The minimum required size for the installation media is 10 GByte.

2.1 USB stick boot media creation

You may use the free Rufus tool to create the boot media.



| 🖋 Rufus 3.14.1788 (Portable) | - 🗆 X | | | | | | |
|--|--------------------------|--|--|--|--|--|--|
| Drive Properties | | | | | | | |
| Drive Properties | | | | | | | |
| Device | | | | | | | |
| KIV (F:) [10 GB] | ~ | | | | | | |
| Boot selection | | | | | | | |
| acontisHypervisor.iso | ✓ SELECT ▼ | | | | | | |
| Persistent partition size | 0 (b) | | | | | | |
| | 0 (No persistence) | | | | | | |
| Partition scheme | Target system | | | | | | |
| MBR | BIOS or UEFI V | | | | | | |
| Show advanced drive properties | | | | | | | |
| Format Options | | | | | | | |
| Volume label | | | | | | | |
| RTV | | | | | | | |
| File system | Cluster size | | | | | | |
| FAT32 (Default) \checkmark | 8192 bytes (Default) 🗸 🗸 | | | | | | |
| Hide advanced format options | | | | | | | |
| V Quick format | | | | | | | |
| Create extended label and icon files | | | | | | | |
| Check device for bad blocks | 1 pass v | | | | | | |
| Chatas | | | | | | | |
| status — | | | | | | | |
| READY | | | | | | | |
| | | | | | | | |
| S) (i) ≵ III | START CLOSE | | | | | | |
| | | | | | | | |
| Using image: acontisHypervisor.iso | | | | | | | |

Fig. 2.1: Settings to create the acontis Hypervisor BOOT stick.

As the ISO is a hybrid ISO, Rufus will ask for the mode. Use the default ISO Image mode.



| ISOHybrid image detected | | | | | | | |
|--------------------------|--|--|--|--|--|--|--|
| ? | The image you have selected is an 'ISOHybrid' image. This means it can be written eith in ISO Image (file copy) mode or DD Image (disk image) mode. Rufus recommends using ISO Image mode, so that you always have full access to the drive after writing it. However, if you encounter issues during boot, you can try writing this image again in I Image mode. | | | | | | |
| | Please select the mode that you want to use to write this image: | | | | | | |
| | Write in ISO Image mode (Recommended) | | | | | | |
| | ○ Write in DD Image mode | | | | | | |
| | OK Cancel | | | | | | |

Fig. 2.2: Image mode selection.

2.2 Installation

Currently, a light-weight Ubuntu derivate (Xubuntu) is used as the Hypervisor service operating system.

- Select Install acontis RTOSVisor to install the Hypervisor, other options are not supported.
- Follow the installation instructions.
- Do not keep any existing Windows or Linux installation, let the installer erase previously installed OSes.
- A user account **must** be created while installing the Hypervisor.

Note: A password MUST be defined!

- After the installation has finished, you have to reboot the system.
- When the BIOS startup finished, a boot menu is shown and the default entry Ubuntu Configure RTOSVisor is selected, keep this and do not select any other entry.

2.3 Remote Access to the Hypervisor using Remote Desktop Connection

When working with Windows for software development, it is recommended to remotely connect to the Hypervisor using a Remote Desktop Connection (using the RDP protocol). On Xubuntu, the xrdp application will provide such access. When using Remote Desktop you may simply copy paste commands from this manual into the Hypervisor shell.

First you need to determine the IP address of the Hypervisor. Open a shell (right click on desktop and select '*Open Terminal here*' or press CRTL + ALT + T)

\$ ifconfig



| 🖉 rte@rte-Systemboard-basic-SYBbasic: ~ | _ | | × |
|--|------------------|--------------------|--------|
| <pre>rte@rte-Systemboard-basic-SYBbasic:~\$ rte@rte-Systemboard-basic-SYBbasic:~\$ ifconfig enpls0: flags=4163<up.broadcast,running,multicast> mtu 1500 inet 172.17.10.68 netmask 255.255.0.0 broadcast 172.17.255.255 inet6 fe80::382d:2745:2c84:cda prefixlen 64 scopeid 0x20<link/> inet6 2a02:590:801:2c00:cl35:44c3:715d:ef88 prefixlen 64 scopeid inet6 2a02:590:801:2c00:fab1:f0e8:c47a:9878 prefixlen 64 scopeid ether 00:60:c8:06:02:9a txqueuelen 1000 (Ethernet) RX packets 68666 bytes 4379604 (4.3 MB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 283275 bytes 399647392 (399.6 MB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 device memory 0x91600000-9161ffff</up.broadcast,running,multicast></pre> | i 0x0< i 0x0< | global: global: | ~ |
| <pre>enp2s0: flags=4163<up,broadcast,running,multicast> mtu 1500 inet6 fe80::ble6:557b:4f49:3f51 prefixlen 64 scopeid 0x20<link/> ether 00:60:c8:06:02:9b txqueuelen 1000 (Ethernet) RX packets 0 bytes 0 (0.0 B) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 186 bytes 32316 (32.3 KB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 device memory 0x91500000-9151ffff</up,broadcast,running,multicast></pre> | | | |
| <pre>enp3s0: flags=4163<up,broadcast,running,multicast> mtu 1500 inet6 fe80::b65:66ba:8elf:85d8 prefixlen 64 scopeid 0x20<link/> ether 00:60:c8:06:02:9c txqueuelen 1000 (Ethernet) RX packets 0 bytes 0 (0.0 B) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 189 bytes 33342 (33.3 KB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 device memory 0x91400000-9141ffff</up,broadcast,running,multicast></pre> | | | |
| <pre>lo: flags=73<up,loopback,running> mtu 65536 inet 127.0.0.1 netmask 255.0.0.0 inet6 ::1 prefixlen 128 scopeid 0x10<host> loop txqueuelen 1000 (Local Loopback) RX packets 658 bytes 52015 (52.0 KB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 658 bytes 52015 (52.0 KB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0</host></up,loopback,running></pre> | | | |
| rte@rte-Systemboard-basic-SYBbasic:~\$ | | | \sim |

Now its possible to access the Hypervisor system through Windows Remote Desktop using the previously determined IP address.

Hint: Use the user account, which was created when installing the Hypervisor.



| nervice Desktop Connection 55 to 2010 | n — | | × |
|---------------------------------------|-----------------|----------|---|
| Login to TP101 | | | ^ |
| xrdf con | Just mecting | | |
| Session Xorg | | - | |
| username | | | |
| password | | | |
| | | | |
| ОК | Canc | el | |
| < | | | > |

Fig. 2.3: xrdp remote login screen.

Caution: To log-in remote into the hypervisor through xrdp, **no** other user must be logged-in into the hypervisor. You can change this behavior by modifying the xrdp startup script /etc/xrdp/ startwm.sh. Add the following 3 lines before the test -x command:

```
if test -r /etc/profile; then
                . /etc/profile
fi
unset DBUS_SESSION_BUS_ADDRESS
unset XDG_RUNTIME_DIR
. $HOME/.profile
```

test -x /etc/X11/Xsession && exec /etc/X11/Xsession

Alternatively you may kill the current session at the Desktop, run the who -u command to determine the process id:



The line with the (:0) is the session running at the physical display. You can kill this session: \$ sudo kill -9 1182 If this does not work, you may kill all sessions of the currently logged in user via the killall command. For example, to kill all sessions of the user hvuser, run the following command:

\$ sudo killall hvuser

2.4 Basic system configuration

2.4.1 Power Management

To assure deterministic real-time behavior it is necessary to disable all power saving settings.

Several settings in the PC BIOS will have to be adjusted. You may take a look at the acontis website for example BIOS settings or pre-validated hardware BIOS settings

You need also to assure the Display Power Management is turned off in the Hypervisor Host.

First, open the Power Manager:



| ¥ | |
|---|---|
| <u> </u> | |
| Desktop Display File Manager Settings Keyboard Mouse and Touchpad Notifications Onboard Settings Panel Power Manager PulseAudio Volume Control Removable Drives and Media Session and Startup Settings Editor Software & Updates Synaptic Package Manager Window Manager | Favorites Recently Used All Applications Settings Settings Accessories Internet Multimedia System |
| A hvuser | ••••••••••••••••••••••••••••••••••••••• |
| | |

Then assure the Display Power Manager is turned off (pull all sliders to the left before):



| • | Power Manager | | | | | | | |
|-------------------------|--------------------------|---------|--|--|--|--|--|--|
| General System | Display Devices | | | | | | | |
| Display power managem | Display power management | | | | | | | |
| Blank after | O | | | | | | | |
| Put to sleep after | ONever | | | | | | | |
| Switch off after | Never | | | | | | | |
| Brightness reduction | herei | | | | | | | |
| On inactivity reduce to | 0 | | | | | | | |
| Reduce after | 0 | | | | | | | |
| | Never | | | | | | | |
| 🕐 Help | | × Close | | | | | | |

2.4.2 hypervisor initialization

Open a shell (right click on desktop and select 'Open Terminal here' or press CRTL + ALT + T)

Change to the bin folder:

\$ cd /hv/bin

In the next step, the memory configuration for the real-time guest OS is set (this setting is **rtos-dependent!**). Use the inithv script:

sudo ./inithv.sh [-baseaddr rtos_baseaddr] rtos_memsize shm_memsize

The optional -baseaddr parameter of the inithv.sh script will set the physical base address where the RTOS shall be located. This parameter is required for RTOS guests which are not relocatable (e.g. VxWorks or RTOS-32). The next parameter defines the amount of memory to be assigned to the real-time guest OS, the last parameter will define the shared memory pool size. All units are in MB.

- Real-time Linux guest: sudo ./inithv.sh 256 16
- On Time RTOS-32 guest: sudo ./inithv.sh -baseaddr 64 64 16
- VxWorks guest: sudo ./inithv.sh -baseaddr 64 64 16

The first parameter of the inithv.sh script will set the amount of memory to be assigned to the real-time guest OS, the second parameters will define the shared memory pool size in MB.



Caution: If you want to install a Windows or Ubuntu guest, the shared memory pool size must be at least 16 MByte and a power of 2 (16, 32, 64, 128, etc.)!

Caution:

If you get the message WARNING: CPU frequency not stable... when running the inithv.sh script, you may not have properly disabled power settings in the BIOS. Please follow the respective instructions above (Prerequisites).

After! rebooting the Hypervisor, resetting the configuration using the hv_resethv command may fix this issue (hv_resethv implicitly re-runs inithv).

Caution:

If you want to run the example guests **after** you had configured guests via the System Manager tool, you must reset the hypervisor configuration.

In that case, all existing example guests will be preserved in /hv/guests/examples.bak and the original example guests copied at /hv/guests/examples.

Run the following steps to do so:

\$ hv_resethv

\$ sudo reboot

Hint:

If you want to change the memory configuration at a later time, please run the hv_adjmemconf command.

The following example shows how to change the configuration to use 768 MByte RAM for a relocatable real-time guest and 32 MByte RAM for the shared memory pool.

\$ hv_adjmemconf 768 32

The following example shows how to change the configuration to use 32 MByte RAM located at a base address of 64 MByte for a non-relocatable real-time guest and 8 MByte RAM for the shared memory pool.

\$ hv_adjmemconf -baseaddr 64 32 8

Now, to get the changes effective, a reboot is required:

\$ sudo reboot



2.5 Validate the configuration

The last step now is to validate if the system is correctly configured. For that purpose, a real-time demo application running in the example real-time guest will be executed. The command to start the guest will automatically start a Debug Console connection to show the output of the guest OS. The Debug Console connection is a low level connection to the guest, similar to a serial line connection on physical guests.

• Real-time Linux:

When starting a Real-time Linux guest, you will have to log in first and start the demo manually.

```
$ cd /hv/guests/examples/rt-linux
$ hv_guest_start -view
```

Log in into Real-Time Linux and run the demo:

```
$ vmf64 login: root
$ password: root
$ RealtimeDemo
```

• On Time RTOS-32:

Caution: The On Time RTOS-32 example guest image is located at a fixed address of 64 MByte. Please assure, you have used the parameter -baseaddr 64 when calling inithv. sh, hv_adjmemconf or hv_resethv.

On RTOS-32 the demo is started automatically and the output shown in the console windows. Adjust the guest configuration setting to prepare starting the *RTOS-32Demo*:

```
$ cd /hv/guests/examples/rtos-32
$ gedit usr_guest_config.sh
```

and add the following line:

export osImage=\$HV_ROOT/guests/examples/rtos-32/Loader.bin

Adjust the link to the demo application:

```
$ cd /hv/guests/examples/rtos-32
```

```
$ rm rtos32app.dlm
```

```
$ ln -s /hv/guests/examples/rtos-32/files/RTOS-32Demo.dlm rtos32app.dlm
```

Run the demo:

```
$ cd /hv/guests/examples/rtos-32
$ hv_guest_start -view
```

Terminate the Debug Console connection to the real-time guest and stop showing the respective output messages: CTRL + C

Adjust the link to the RealtimeDemo application:

```
$ cd /hv/guests/examples/rtos-32
$ rm rtos32app.dlm
$ ln -s /hv/guests/examples/rtos-32/files/RealtimeDemo.dlm
$ rtos32app.dlm
```

Run the demo:



\$ cd /hv/guests/examples/rtos-32
\$ hv_guest_start -view

• VxWorks:

```
$ cd /hv/guests/examples/vxworks
$ hv_guest_start -view
```

Run the demo:

\$ demoStart

To terminate the Debug Console connection to the real-time guest and stop showing the respective output messages: CTRL + C

As an alternative to the Debug Console on RT-Linux, you may also use a ssh connection:

\$ ssh root@192.168.157.2

The connection can be terminated by entering the exit command:

\$ exit

Finally, stop the Real-time guest OS:

• Real-time Linux:

\$ cd /hv/guests/examples/rt-linux

- \$ hv_guest_stop
- On Time RTOS-32:

\$ cd /hv/guests/examples/rtos-32
\$ hv_guest_stop

- VxWorks:
 - \$ cd /hv/guests/examples/vxworks
 - \$ hv_guest_stop

To restart a running guest, you can use the hv_guest_restart command, for example:

```
$ cd /hv/guests/examples/rt-linux
$ hv_guest_restart
```



3 Ethernet Device Assignment

In this quickstart guide an Ethernet device will be assigned in **polling** mode to the RTOS.

Hint: For more specific options like *interrupt mode* please check the chapter **RTOS Devices** (**Partitioning**) of the Hypervisor Manual.

3.1 Device Identification

In a first step, it is required to determine the Ethernet device that shall be used by the Real-time guest. There are several ways how to detect the desired adapter.

3.1.1 Identify by hardware information

An easy way to identify an adapter is its hardware information:

```
$ lshw -class network
```

returns

```
*-network:1
   description: Ethernet interface
   product: 82545EM Gigabit Ethernet Controller (Copper)
   vendor: Intel Corporation
   physical id: 6
   bus info: pci@0000:02:06.0
   logical name: enp2s0
   version: 01
   serial: 00:0c:29:94:bb:c3
   size: 1Gbit/s
   capacity: 1Gbit/s
   width: 64 bits
   clock: 66MHz
   capabilities: bus_master cap_list rom ethernet physical logical tp_
→10bt 10bt-fd 100bt 100bt-fd 1000bt-fd autonegotiation
   configuration: autonegotiation=on broadcast=yes driver=e1000_
→driverversion=5.15.0-88-acontis duplex=full ip=172.17.10.26 latency=0_
→link=yes mingnt=255 multicast=yes port=twisted pair speed=1Gbit/s
   resources: irq:16 memory:fd580000-fd59ffff memory:fdfe0000-fdfeffff_
→ioport:2080(size=64) memory:fd520000-fd52ffff
```

We can see many information helping on identification: The network adapter enp2s0 is an Intel type 82545EM with MAC-ID 00:0c:29:94:bb:c3 and current link state link=yes.



3.2 Device Assignment

Run the following command to assign the above device.

```
$ sudo hv_addeth enp2s0 rtos_eth1 1
```

3.3 RTOS Assignment

To assign a specific device to Real-time guests in general, the partitioning script /hv/config/ usr_hvpart.sh must be adjusted.

The device assignment scripts <RTOS device name>.sh usually shall be executed automatically on system startup. To accomplish this, you need to add the respective <RTOS device name>.sh calls into the file /hv/config/usr_hvpart.sh. In our tutorial we use rtos_eth1 as the unique <RTOS device name>, as mentioned earlier.

```
$ gedit /hv/config/usr_hvpart.sh
```

The usr_hvpart.sh file should contain at least the following string after editing: source \$HV_CONFIG/rtos_eth1.sh \$cmd

The example below shows how the device with the *unique* name rtos_eth1 is assigned.

```
#!/bin/bash
cmd="add"
[ $1 == "delete" ] && cmd="delete"
# unbind devices (assign to RTOS)
source $HV_CONFIG/rtos_eth1.sh $cmd
```

Please run the hv_hvpart command with the parameter add or reboot the system to make the change effective.

\$ hv_hvpart add

You may use the delete parameter to assign all RTOS devices back to the Hypervisor Host.

\$ hv_hvpart delete

Hint: The <code>\$HV_BIN/hvpart.sh</code> script will be automatically started via the *systemd* service controlled via /etc/systemd/system/hv_part.service. This script will call the usr_hvpart. sh script which includes user specific partitioning commands. This service can be *enabled* or *disabled* as shown below (by default, it is enabled)

```
$ sudo systemctl enable /hv/services/hv_part.service
$ sudo systemctl disable hv_part
```



3.4 Device Verification

You can verify if the assignment is active by checking if the Hypervisor Host's *original* driver is **not** used in conjunction with the devices assigned to the Real-time guest.

\$ lspci -k

The output will look similar like the following excerpt:

| | : | | : | : | : | : | : | : | : | : | _ |
|--|---------------|---------------------|-------------|--------------------|------------------------|-----------------------|------------|-----------|------------|---|----------|
| \hookrightarrow | | : | | : | | | | | | | |
| | : | | : | : | : | : | : | : | : | : | — |
| \hookrightarrow | | • | | : | | | | | | | |
| 01:00.0 Ethernet controller: Intel Corporation I210 Gigabit Network. →Connection (rev 03) | | | | | | | | | | | |
| | | Subsyst Kernel | tem dr: | : Intel iver in | Corpora use: ig | ation I210 gb | Gigabit N | etwork Cc | onnection | | |
| | | Kernel | moo | dules: | igb | | | | | | |
| 02:0 →Co |)0.0 onneo | Etherne ction (: | et (rev | control 03) | ller: Int | cel Corpor | ation I210 | Gigabit | Backplane | | |
| | | Subsyst Kernel | tem dr: | : Intel iver in | L Corpora n use: po | ation I210 ci-stub | Gigabit B | ackplane | Connection | | |
| | | Kernel | mo | dules: | igb | | | | | | |
| | : | | : | : | : | : | : | : | : | : | — |
| \hookrightarrow | | : | | : | | | | | | | |
| | : | | : | : | : | : | : | : | : | : | — |
| \hookrightarrow | | : | | : | | | | | | | |

In the above example, the instance 01:00.0 is used by Ubuntu (driver: *igb*, *e1000e* etc.) and the instance 02:00.0 is assigned to a Real-time guest (driver: *pci-stub*).

3.5 Adjust the configuration

After creating the device configuration file <RTOS device name>.config, it needs to be included into the guest configuration file to become effective for the respective guest. In this tutorial we use rtos_eth1 as the unique <RTOS device name>.

- RTOS-32 guest:
 - \$ gedit /hv/guests/examples/rtos-32/usr.config
- Real-time Linux guest:
 - \$ gedit /hv/guests/examples/rt-linux/usr.config

Add or uncomment #include "/hv/config/rtos_eth1.config".

The following example shows the 'modified' usr.config file:

```
RtosConfig
;------; acontis technologies GmbH
;
; Guest user configuration
;-----
```

(continues on next page)

(continued from previous page)

```
#include "/hv/config/rtos_eth1.config"
;-----; End of file
```

;-----

3.6 Reboot Hypervisor Host

\$ sudo reboot now



4 Running the acontis EtherCAT Master Demo (RT-Linux)

In this section we will get the EtherCAT master demo application running. It is assumed that an Ethernet adapter has been assigned to the RTOS.

4.1 Adjust the configuration

You may also have to adjust the parameters for the master stack demo on behalf.

Caution: This should **only be done** if the supplied default values for this quickstart guide **arn't** sufficient!

\$ gedit /hv/guests/etc/rt-linux/files/ecat/noeni.sh

respectively

\$ gedit /hv/guests/etc/rt-linux/files/ecat/real.sh

4.2 Step 1: Scan the EtherCAT network

```
$ cd /hv/guests/examples/rt-linux
$ hv_guest_start -view
```

Start the EC-Master (*without* an ENI):

```
$ vmf64 login: root
$ password: root
$ cd /mnt/rtfiles/etc/rt-linux/files/ecat
$ ./noeni.sh
```

Finally, terminate the console connection to the real-time guest and stop the Real-time guest OS:

CTRL + C \$ hv_guest_stop

4.3 Step 2: Set the EtherCAT network operational

In this section we will run the EtherCAT network into **OPERATIONAL** mode. You need to be familiar with how to create an ENI file for EtherCAT networks.

In a first step, please generate a network configuration file (ENI) for the connected network.

Copy your ENI file to /hv/guests/etc/rt-linux/files/ecat/real.xml.

Start the EC-Master (*with* an ENI):



```
$ vmf64 login: root
$ password: root
$ cd /mnt/rtfiles/etc/rt-linux/files/ecat
$ ./real.sh
```