acontis technologies GmbH

SOFTWARE

Hypervisor-Quickstart-Guide

acontis Real-time Hypervisor Quickstart

Version 9.x

Edition: July 29, 2025

Table of Contents

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 `Intel Gigabit` 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

To install RTOSVisor, you will need to create a bootable USB stick using the file:*RTOSVisor.iso* file. Use a tool such as Rufus (for Windows) or the dd command (for Linux/macOS) to write the file:*RTOSVisor.iso* file to the USB stick. Once the USB stick is successfully created, it can be used to boot and install RTOSVisor on the target system.

In this guide, we will use Rufus to create the bootable media.

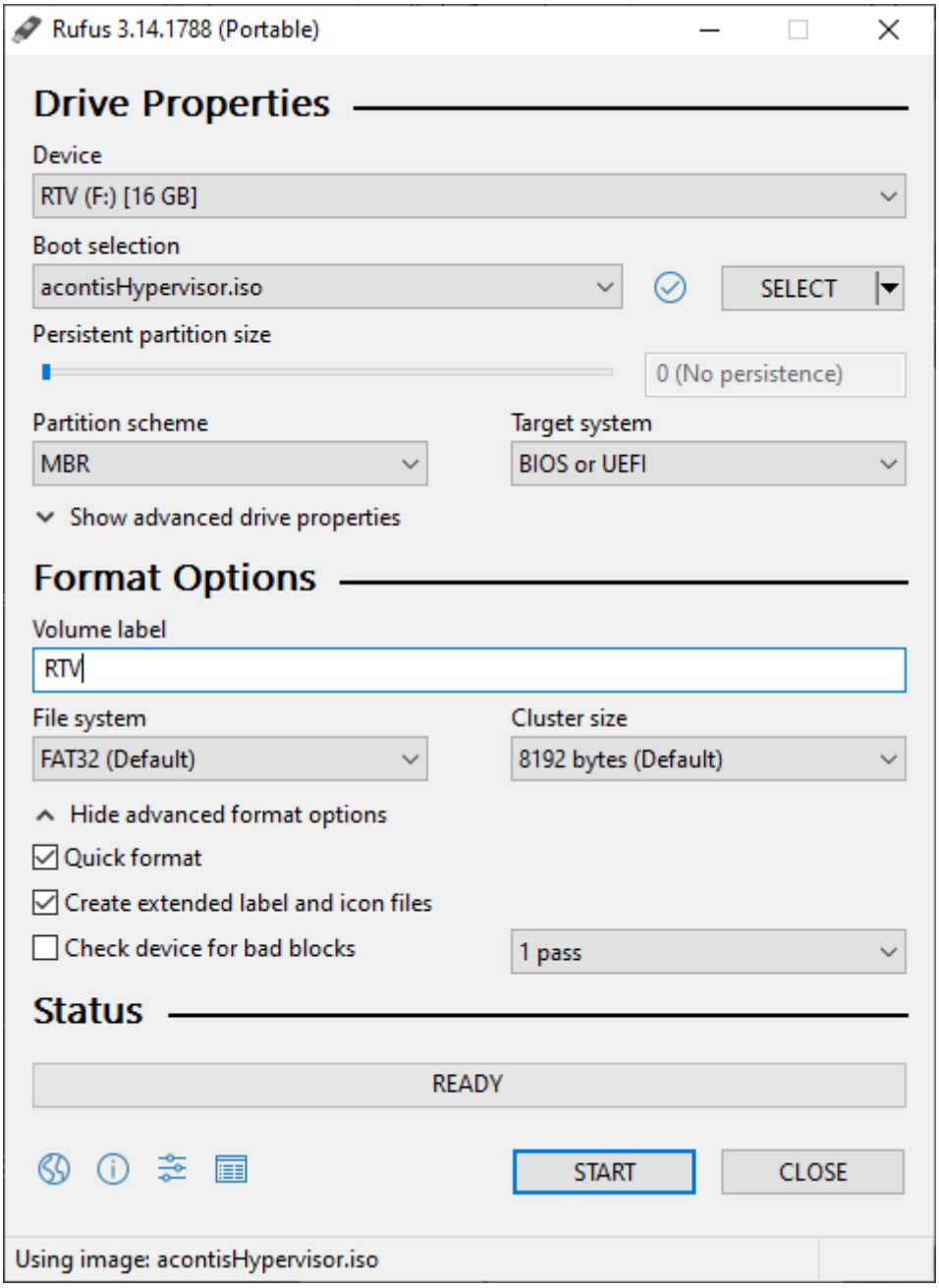


Fig. 2.1: Settings to create the acontis Hypervisor BOOT stick.

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

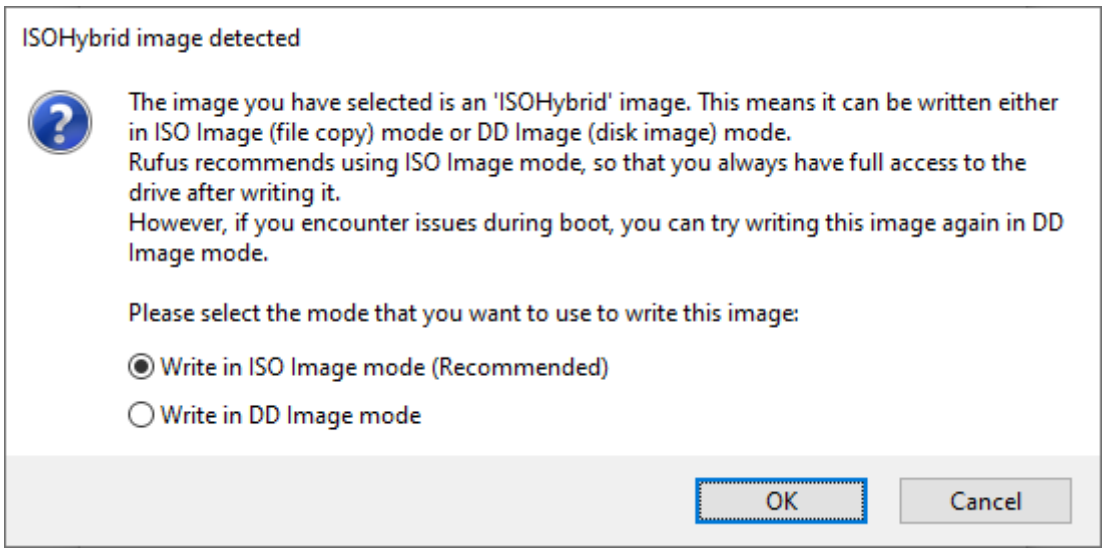


Fig. 2.2: Image mode selection.

2.2 Installation

Boot the target PC from the USB stick you just created.

A stripped down and light-weight Debian Linux 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. You may also wish to reset the BIOS settings so that USB is no longer the primary boot device.
- When the BIOS startup finished, a boot menu is shown and the default entry `Configure RTOSVisor` is selected, keep this and do **not** select any other entry.

2.3 Remote Desktop Access to the Hypervisor

When working with Windows for software development, you may want to connect remotely to the Hypervisor Host using a Remote Desktop Connection (using the RDP protocol). The `xrdp` application of the Hypervisor Host will provide such access. When using Remote Desktop you may simply copy paste commands from this manual into the Hypervisor shell. If you want to work directly on the Hypervisor Host, you can skip this chapter.

First, you need to connect your Hypervisor Host to your company network and then determine the IP address. Open a shell (right click on Hypervisor Host desktop and select *'Open Terminal here'* **or** press `CRTL` + `ALT` + `T`)

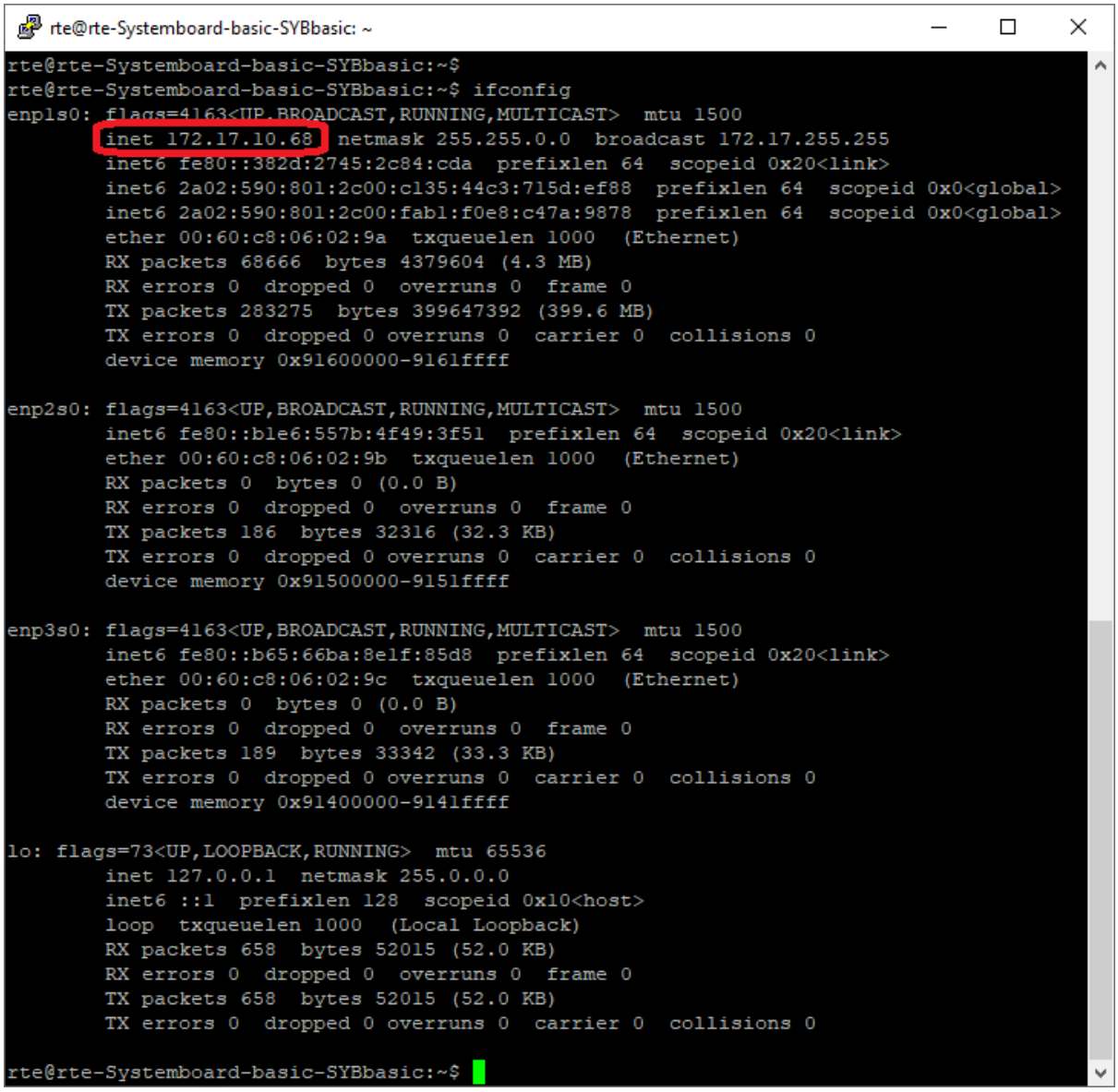
Caution: An **active** RDP session **MUST be** left by a explicit *log-out*. Closing the RDP window **without explicit** log-out leaves the current session **active** and this **prevents** a later *local* log-in into the system! In such case, you need to terminate the existing session. First, the process ID of the session needs to be determined, then the process be killed (replace <username> by your username, in the below example, the process id is 2637):

```
$ pgrep -a -u <username> Xorg
$ 2637 /usr/lib/xorg/Xorg :11 -auth .Xauthority -config xrdp/xorg.conf
↪  -noreset -nolisten tcp -logfile .xorgxrdp.%s.log
$ sudo kill 2637
```

To automatically end RDP sessions after a few seconds of inactivity or disconnection, you need to configure the xrdp-sesman service. This is handled by the `/etc/xrdp/sesman.ini` file. You may adjust the below two parameters to force killing disconnected sessions after 2 seconds and then restart the service:

```
$ sudo mousepad /etc/xrdp/sesman.ini
$
$ :   :   :   :
$ [Sessions]
$ KillDisconnected=true
$ DisconnectedTimeLimit=2
$ :   :   :   :
$
$ sudo systemctl restart xrdp
```

```
$ ifconfig
```



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.

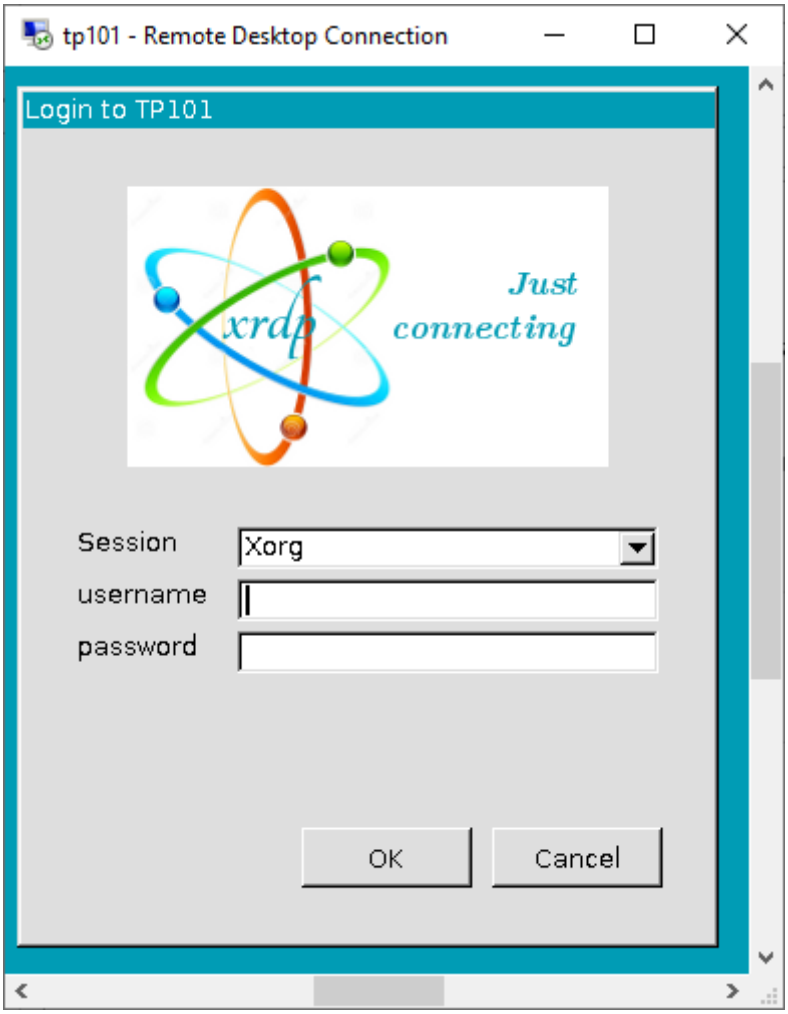


Fig. 2.3: xrdp remote login screen.

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 killing the current session at the Desktop, run the `who -u` command to determine the process id:

```
$ who -u
hvuser   tty7         2023-01-29 18:26 00:08         1182 (:0)
hvuser   pts/2        2023-01-29 18:31   .           2410 (192.168.178.
↪21)
$
```

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

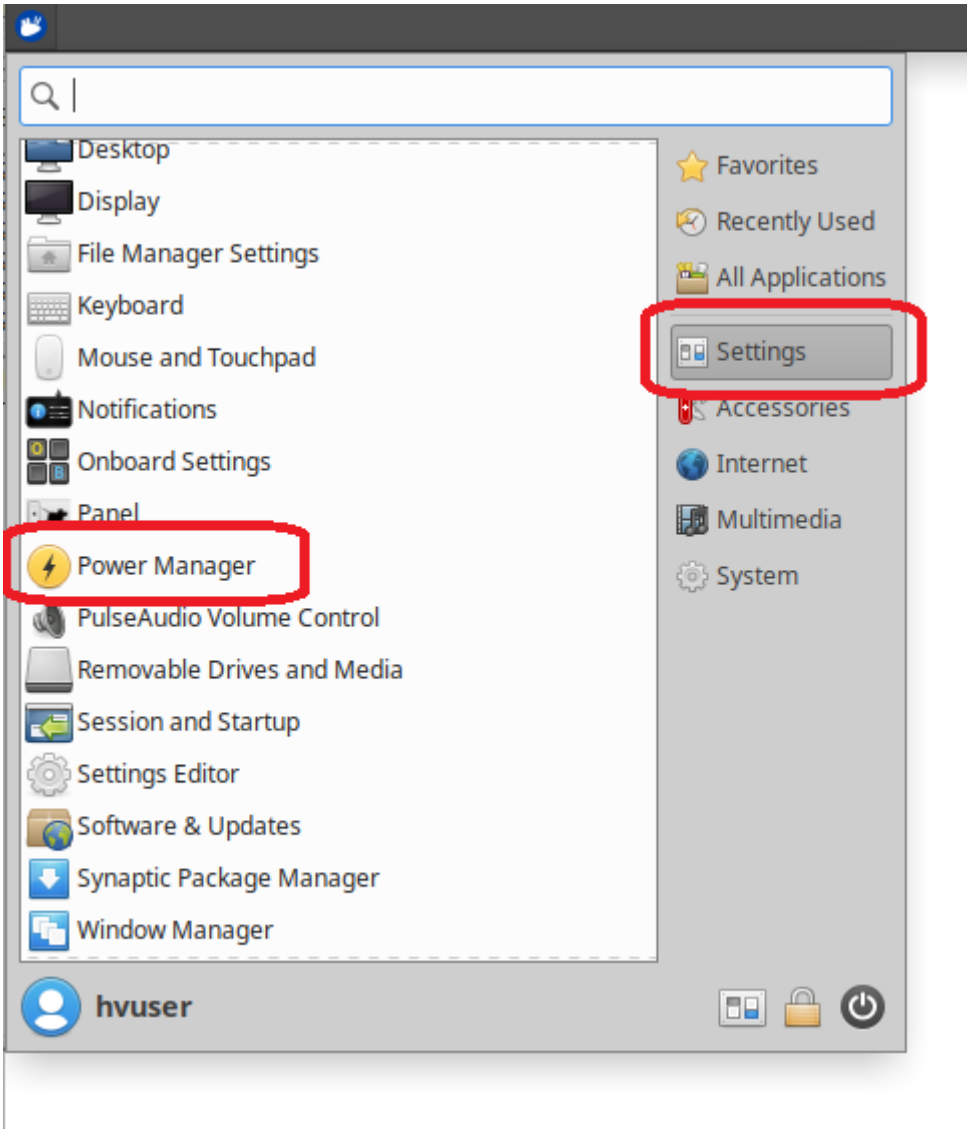
2.4.1 Power Management

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

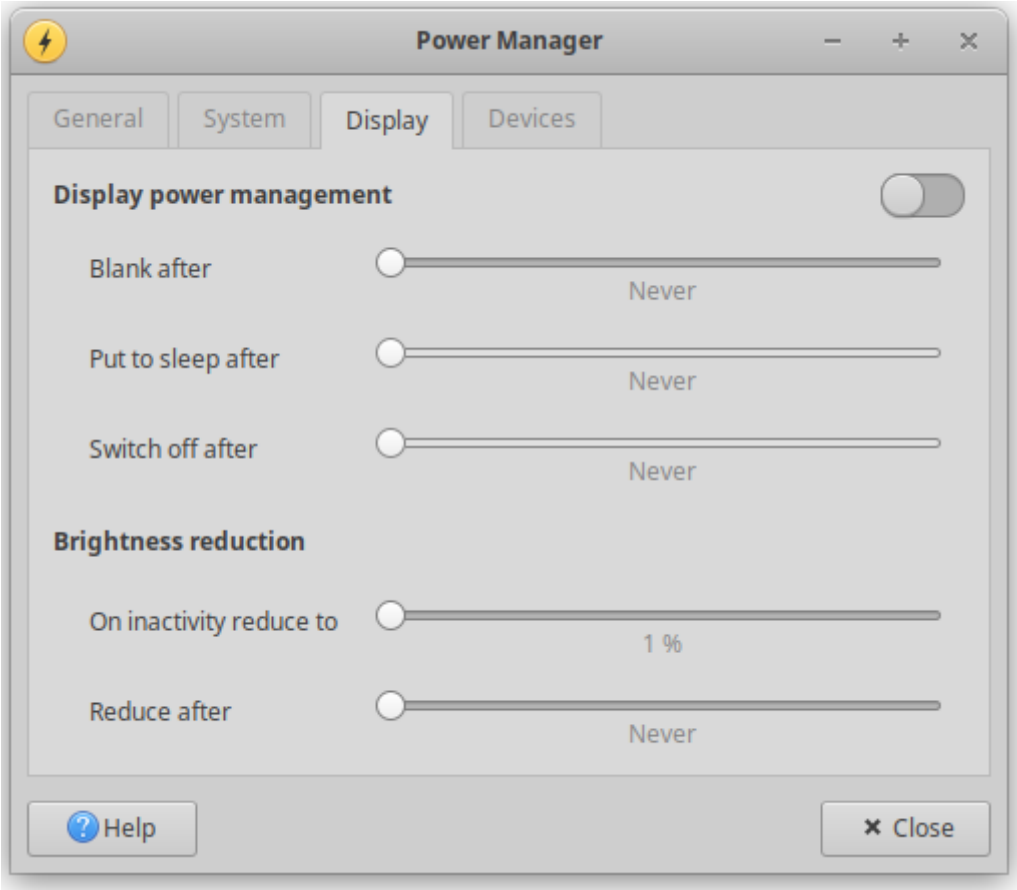
Several settings in the PC BIOS will have to be adjusted. You may take a look at the acontis website for example BIOS settings or pre-validated hardware BIOS settings

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

First, open the Power Manager:



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


Power Manager
General
System
Display
Devices
Display power management
Blank after
Never
Put to sleep after
Never
Switch off after
Never
Brightness reduction
On inactivity reduce to
1 %
Reduce after
Never
Help
Close

3 Validate Example Guests

The final step is to verify the correct configuration of the system. To achieve this, a demo application will be run on various sample real-time guests.

> **Caution:** When the example guests are initialized, the corresponding *pre-configured* System Manager projects will be loaded. This process will **delete** any *existing* guests, subject to your confirmation. You can preserve your current configuration by using the System Manager to save it **before** initializing an example. Ensure you also save any manually added content in the `/hv/guests/guestxxxx` folders, as these will be **deleted** as well.

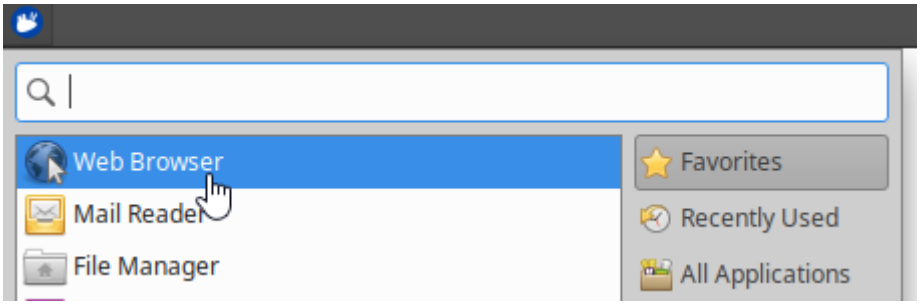
3.1 Preparation

The System Manager is a *web-based* graphical interface designed for managing RtosVisor projects.

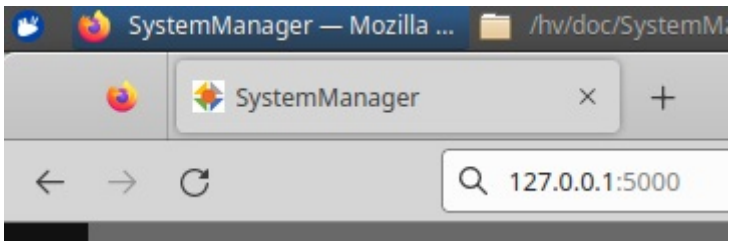
Hint: Detailed documentation on the System Manager is available in the System Manager Manual.

3.1.1 Browser Start

The Firefox browser can be launched directly on the Hypervisor Host. Click the top left icon and select `Web Browser`.

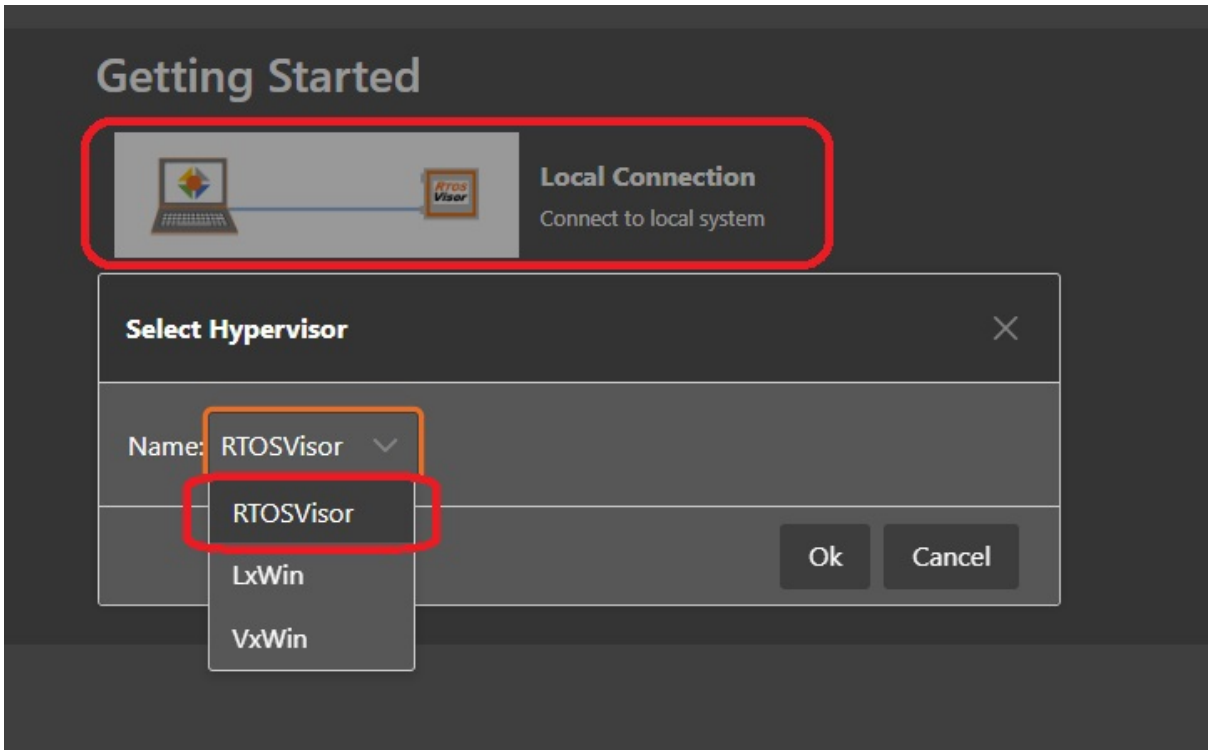


Type in the *localhost* IP address `127.0.0.1` and connect to port `5000`.
The following URL can be used: `http://127.0.0.1:5000`

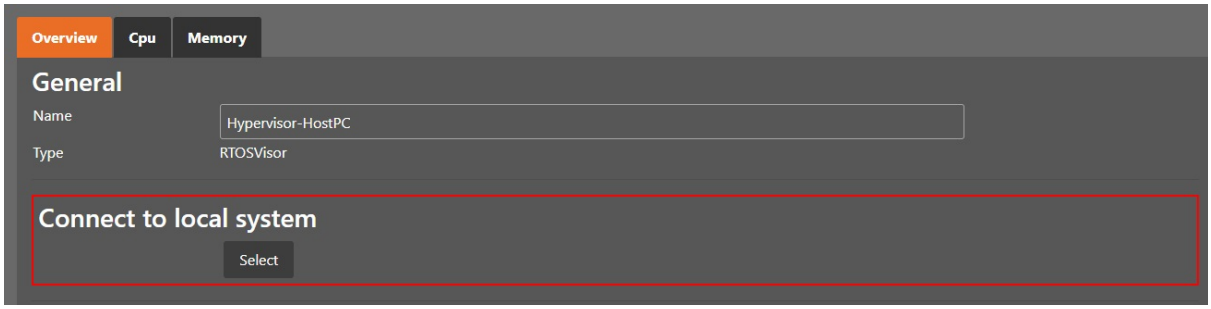


3.1.2 Local Connection

To connect the browser with the Hypervisor backend, please select `Local Connection` and the respective hypervisor type (`RTOSVisor`).



After acknowledging `Connect to local system`, the Hypervisor Host connection dialog will be shown. You should provide an appropriate name for this Hypervisor Host and press the `Select` button.



3.1.3 Initial Synchronization

When you have started the System Manager for the first time, you need to run an initial synchronization step.

Click the synchronization button.

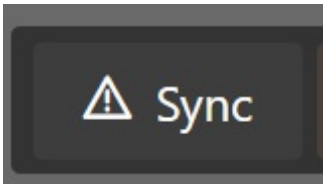
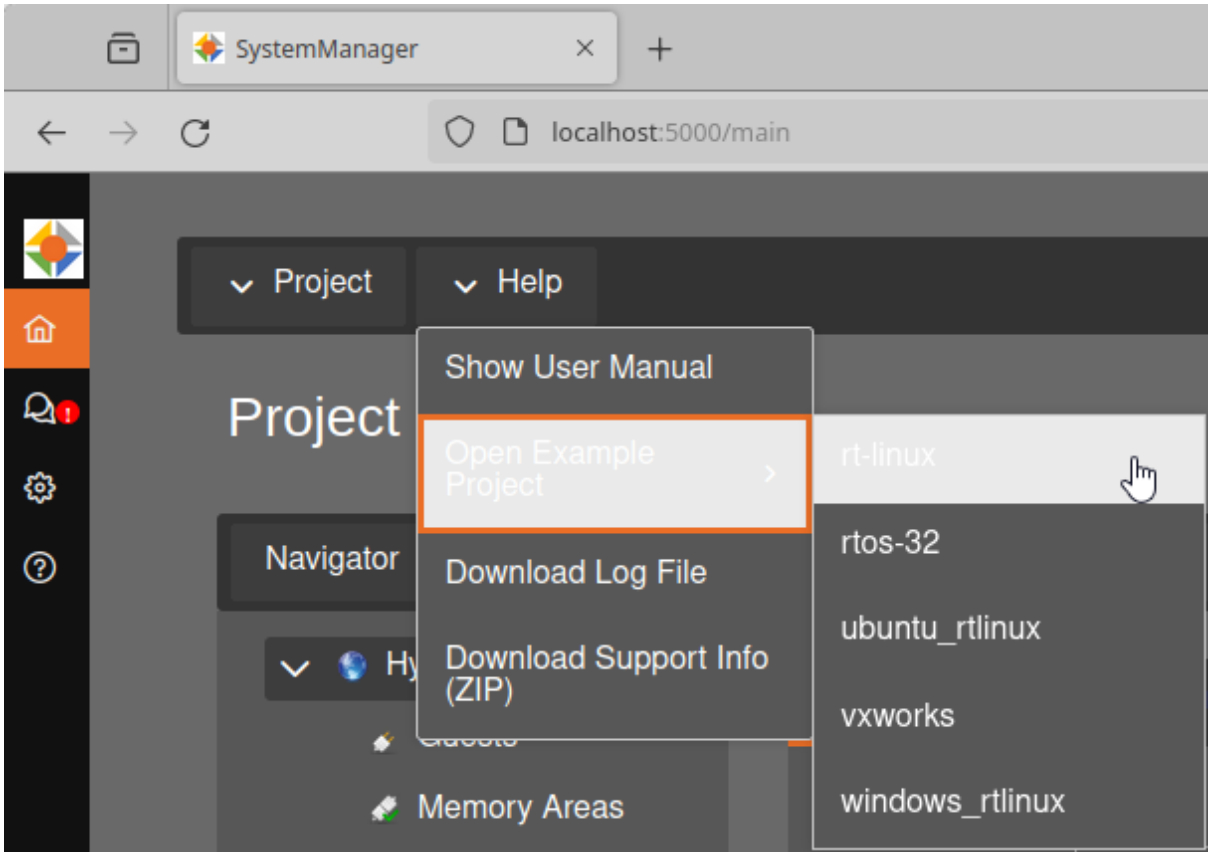


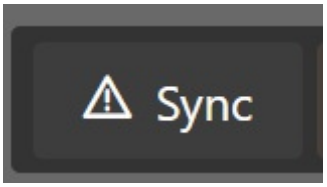
Fig. 3.1: A confirmation popup will then appear, where you need to press the `Apply` button.

3.2 RT-Linux Guest Example

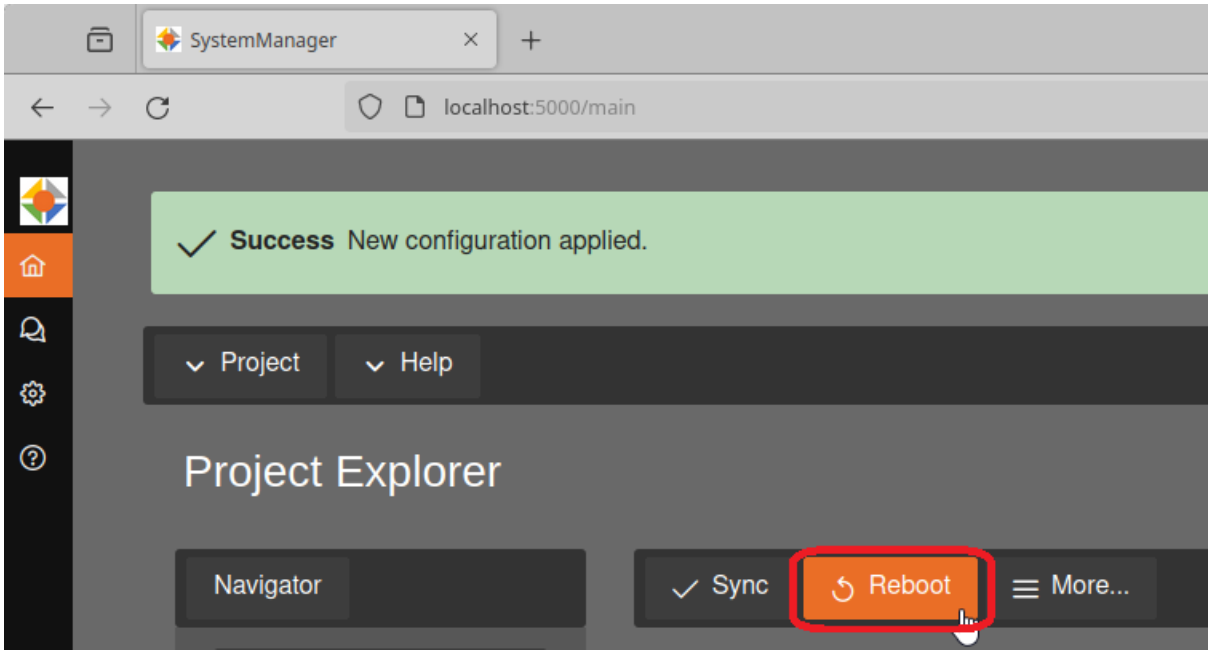
To open the Real-time Linux example with the System Manager, open the 'Help' menu, select 'Open Example Project' and select `rt-linux`.



After Loading or changing a project you have to do the synchronization step again and click the `Apply` in the confirmation popup. This action writes the configuration to the Hypervisor Host.



Once synchronization is successful, the Hypervisor Host has to be rebooted.



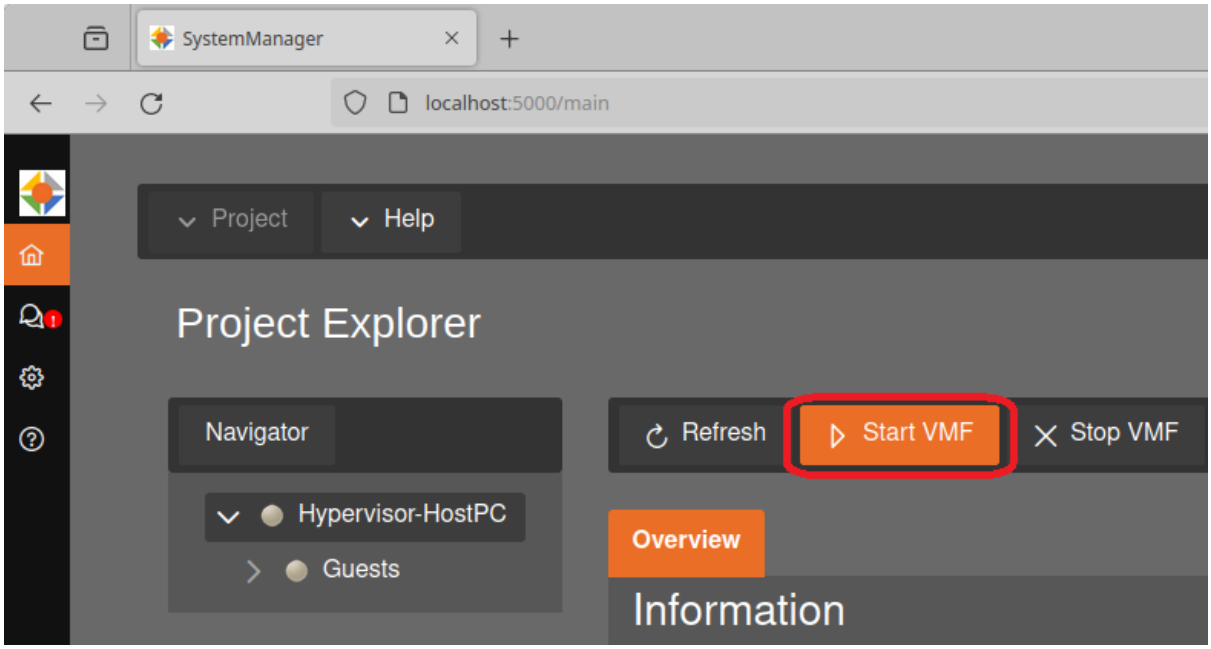
After rebooting the system, repeat the steps from the beginning.

- Start the browser
- Connect to `127.0.0.1:5000`
- Select `Local Connection`, then select `RTOSVisor` in the popup window
- Click the `Select`-button in the section `Connect to local system`

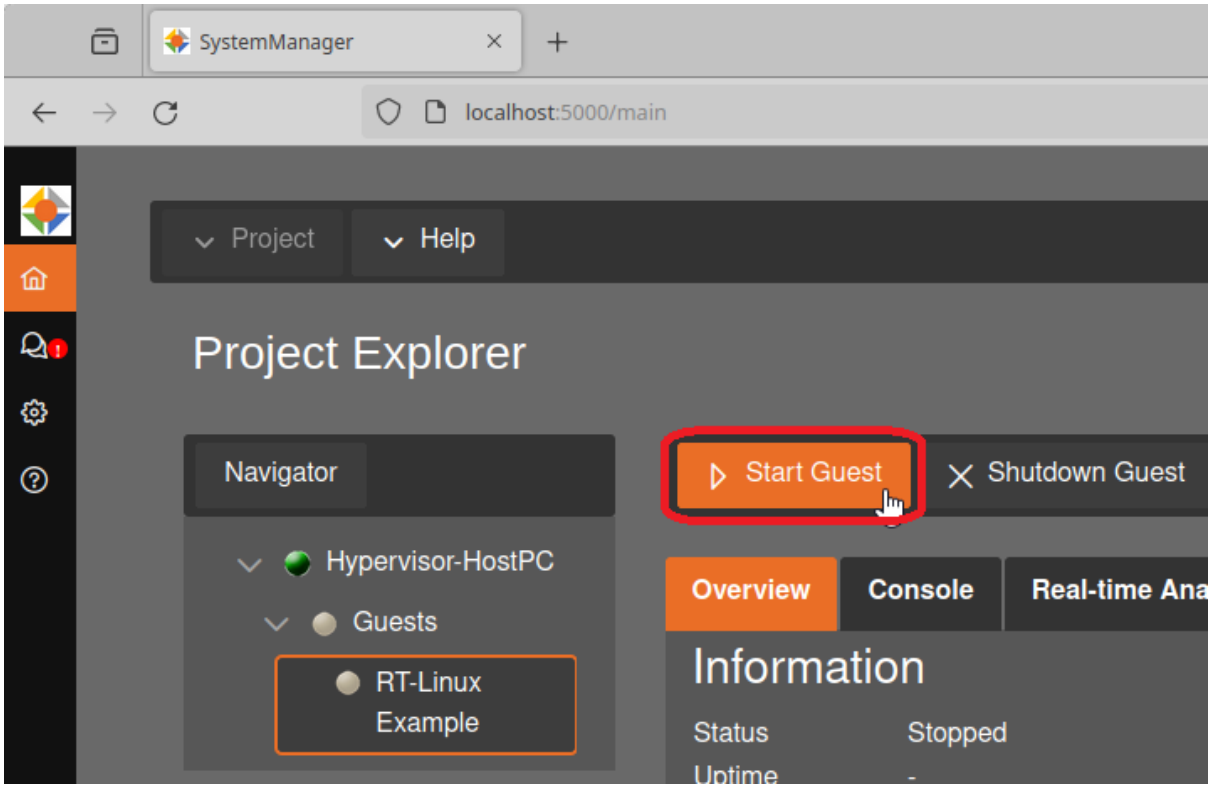
With this the configuration is loaded and the real-time Linux guest can be launched.

Change to `Run` mode by pressing the button `Run` in the top right corner:

Start the *Virtual Machine Framework (VMF)*. Starting the VMF will load the configuration for **all** guests. Real-time guests can **only** be launched when the VMF is running.

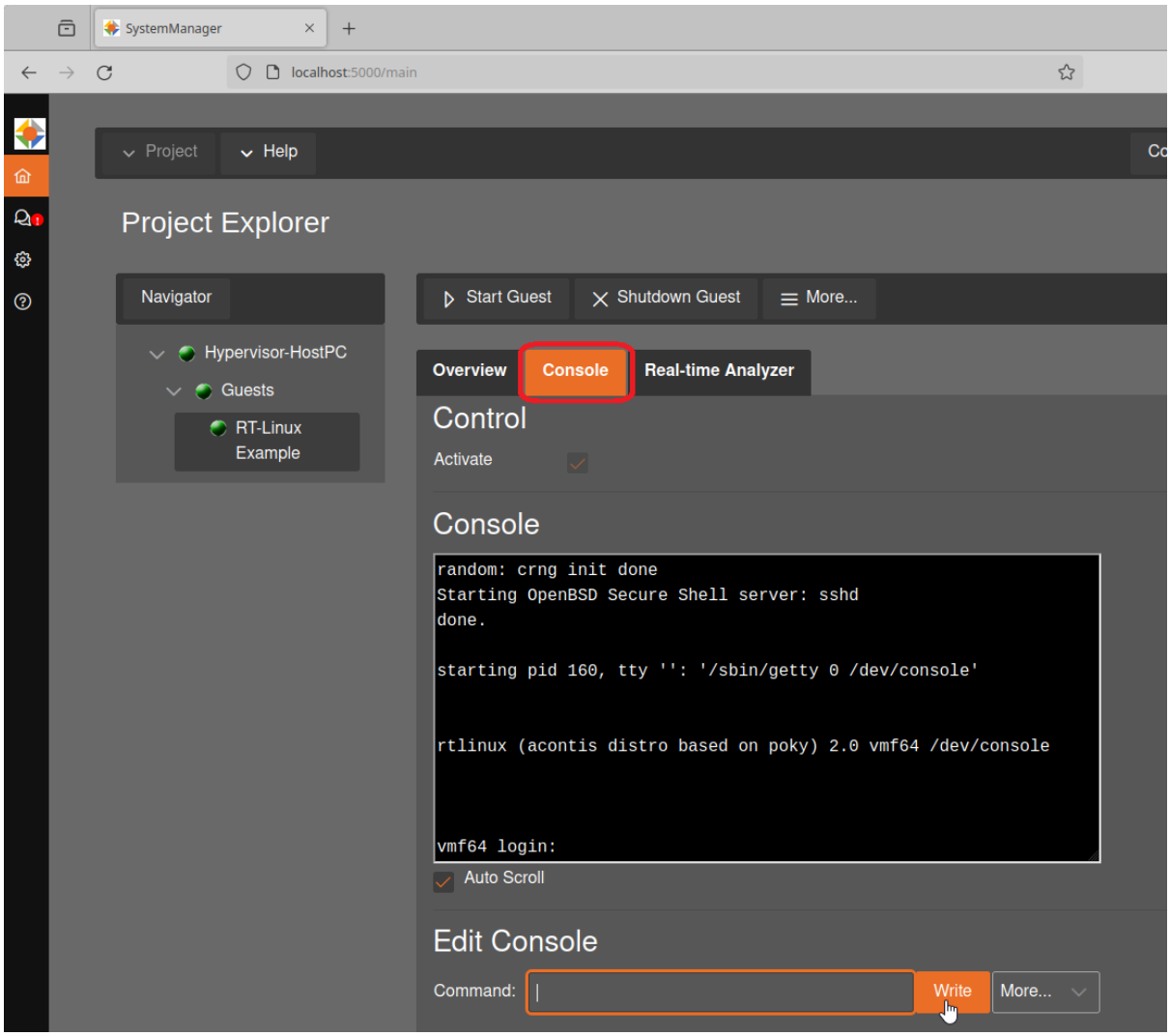


Select the `RT-Linux Example` guest in the `Navigator` on the left side.
Click the `Start Guest` button.

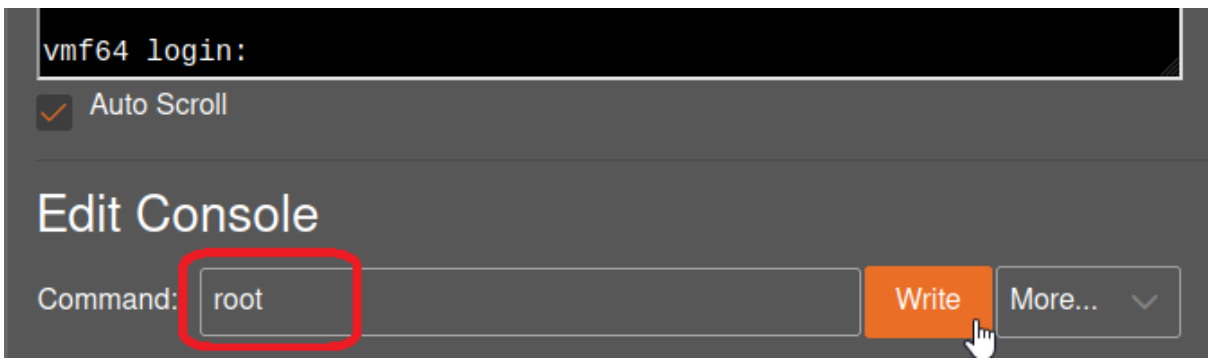


Click the `Console` button to access the console view, displaying the output from the real-time Linux guest.
Commands can be entered in the `Command` field. Begin by logging in with the username `root` and the

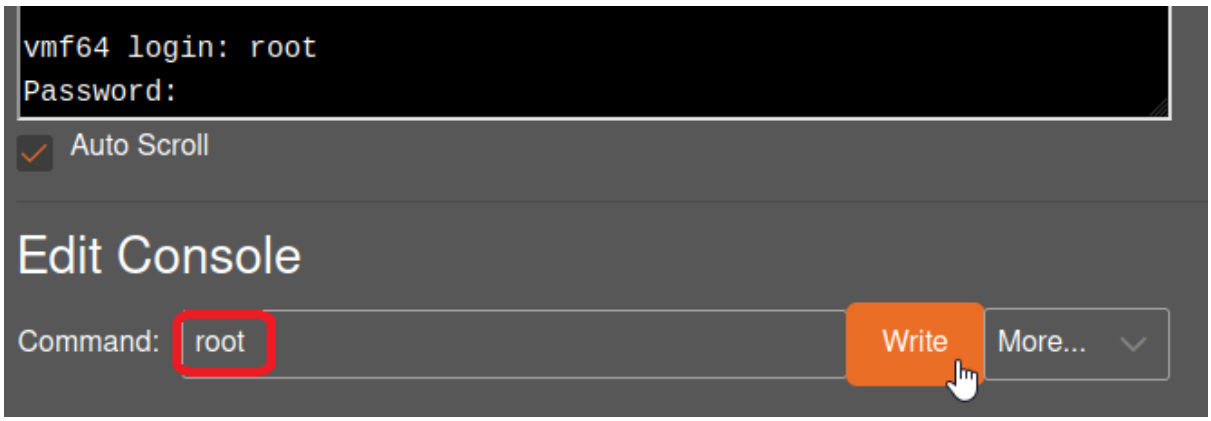
password `root`. To launch the RealtimeDemo application, enter the `RealtimeDemo` command. This is detailed in the following steps:



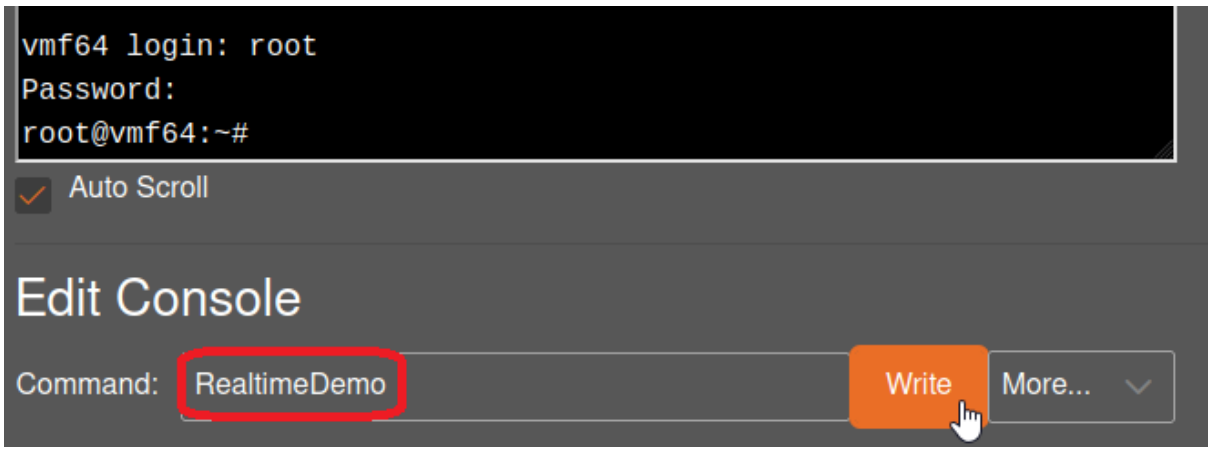
Enter `root` for vmf64 login.



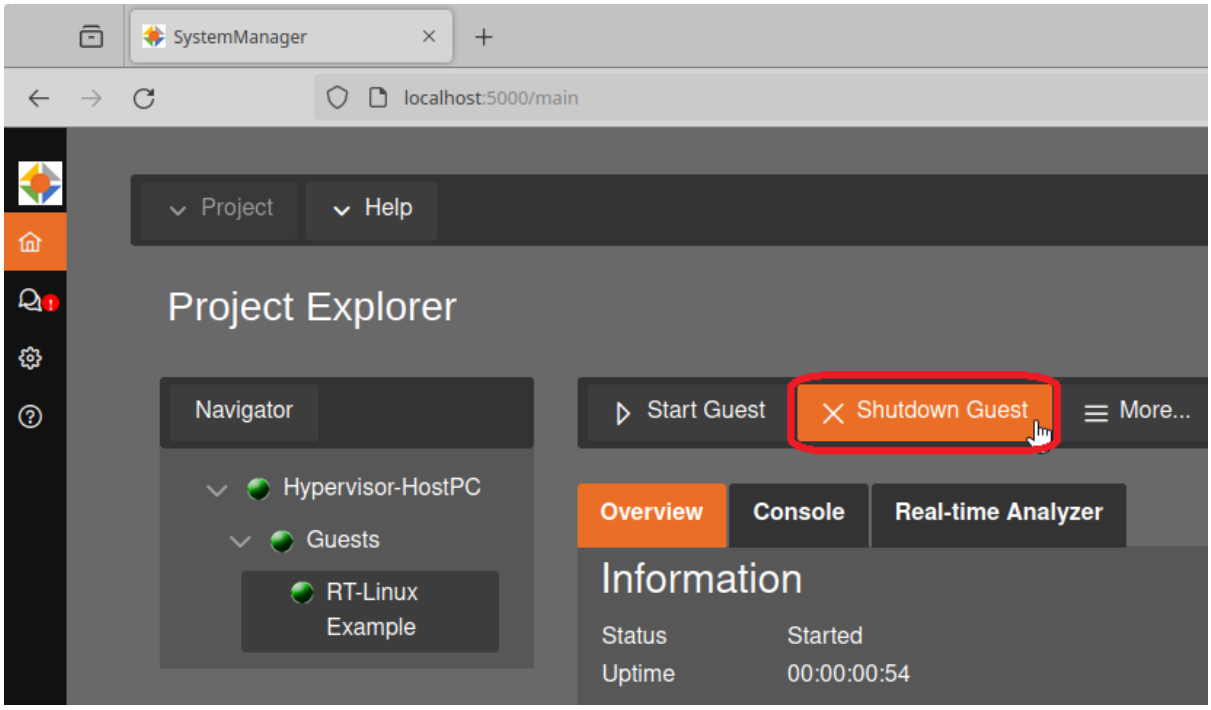
Enter `root` for the password.



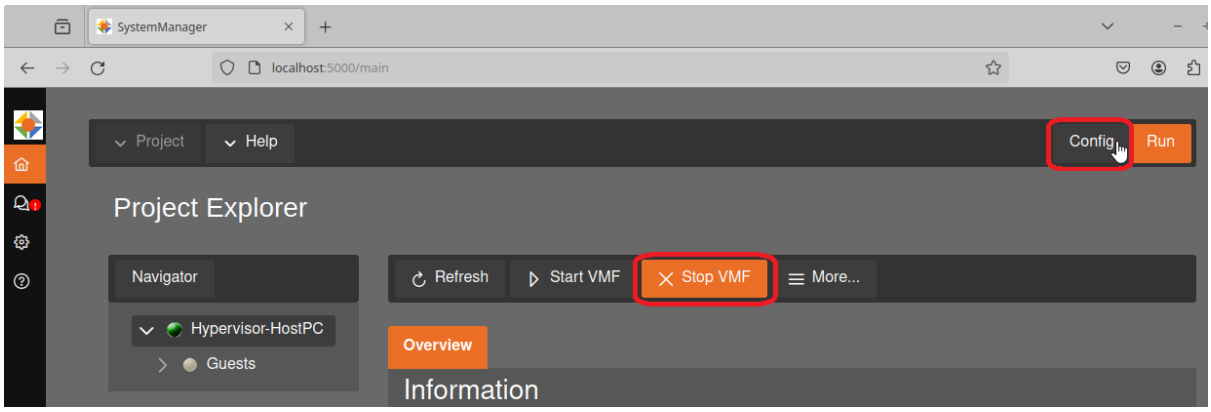
Enter `RealtimeDemo` to start the real-time demo application.



To finish the `RealtimeDemo` application, you can stop the guest by clicking the `Shutdown Guest` button.

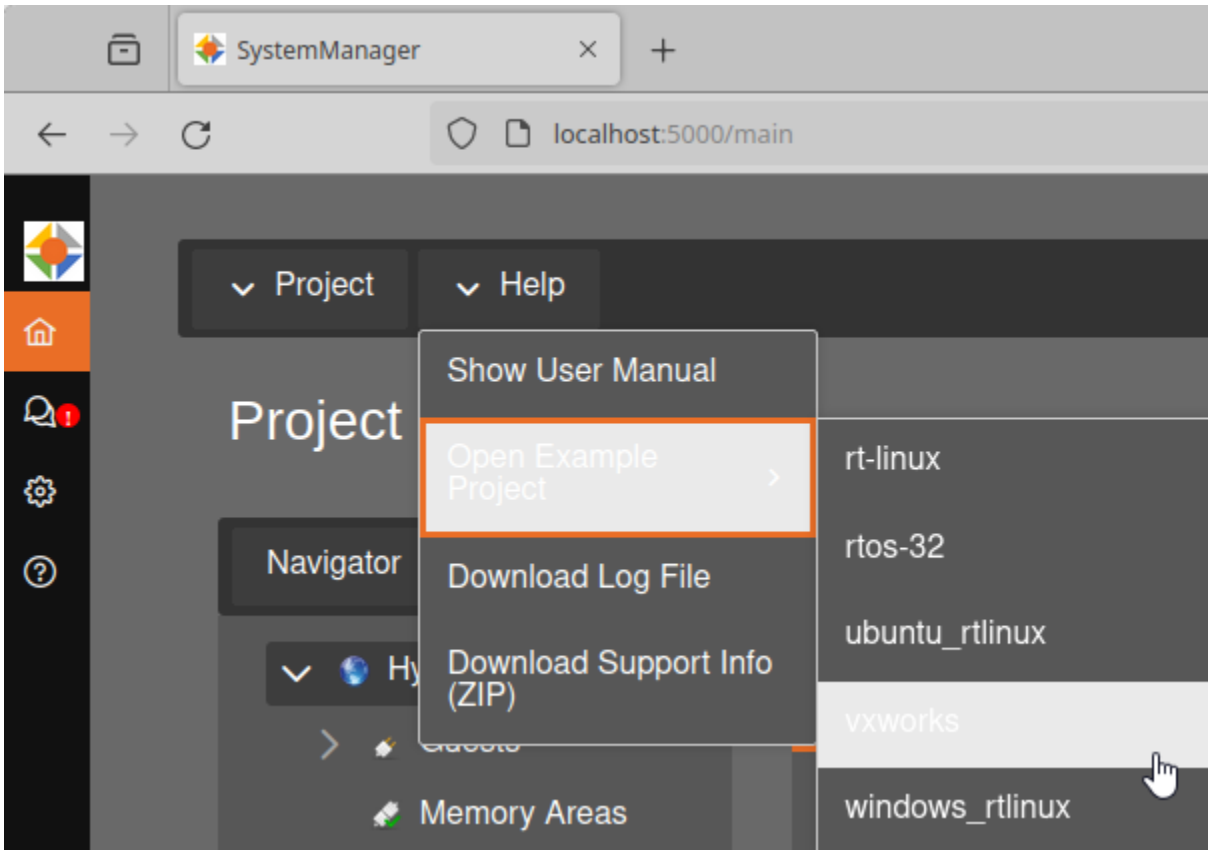


Once the guest is stopped, select the Hypervisor-HostPC on the `Navigator` tab, stop the VMF and switch back to `Config` mode by clicking the button in the top right corner.

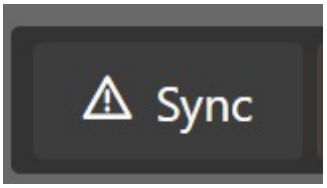


3.3 VxWorks Guest Example

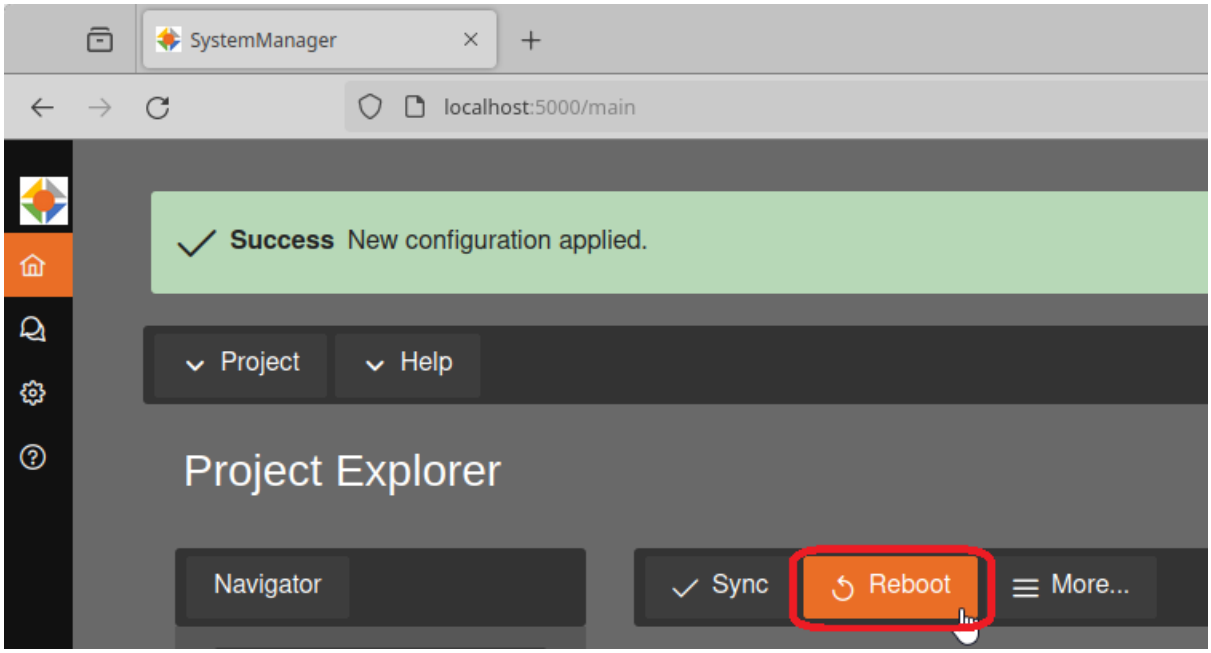
To open the VxWorks example, open the 'Help' menu, select 'Open Example Project' and select 'vxworks'.



After loading or changing a project you have to press `Sync again and press `Apply` in the confirmation popup. This action writes the configuration to the Hypervisor Host.



Once synchronization is successful, the Hypervisor Host has to be rebooted.



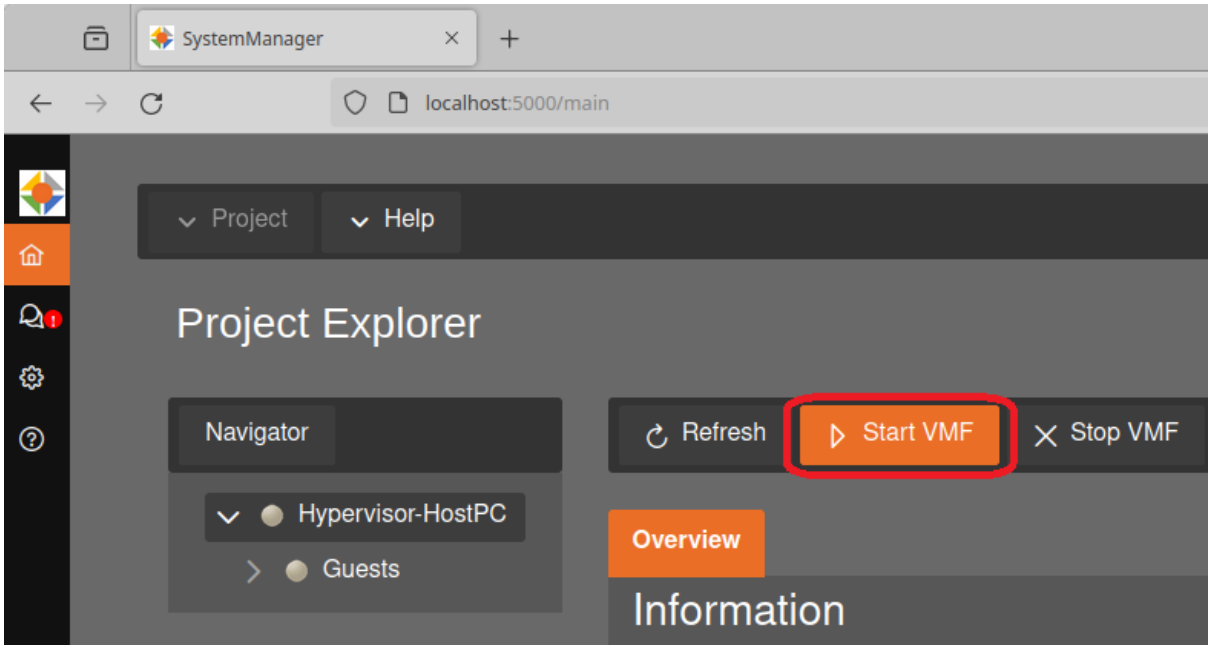
After rebooting the system, repeat the steps from the beginning.

- Start the browser
- Connect to `127.0.0.1:5000`
- Select `Local Connection` and select `RTOSVisor` as Hypervisor
- Click on the `Select` button in the `Connect to local system` section

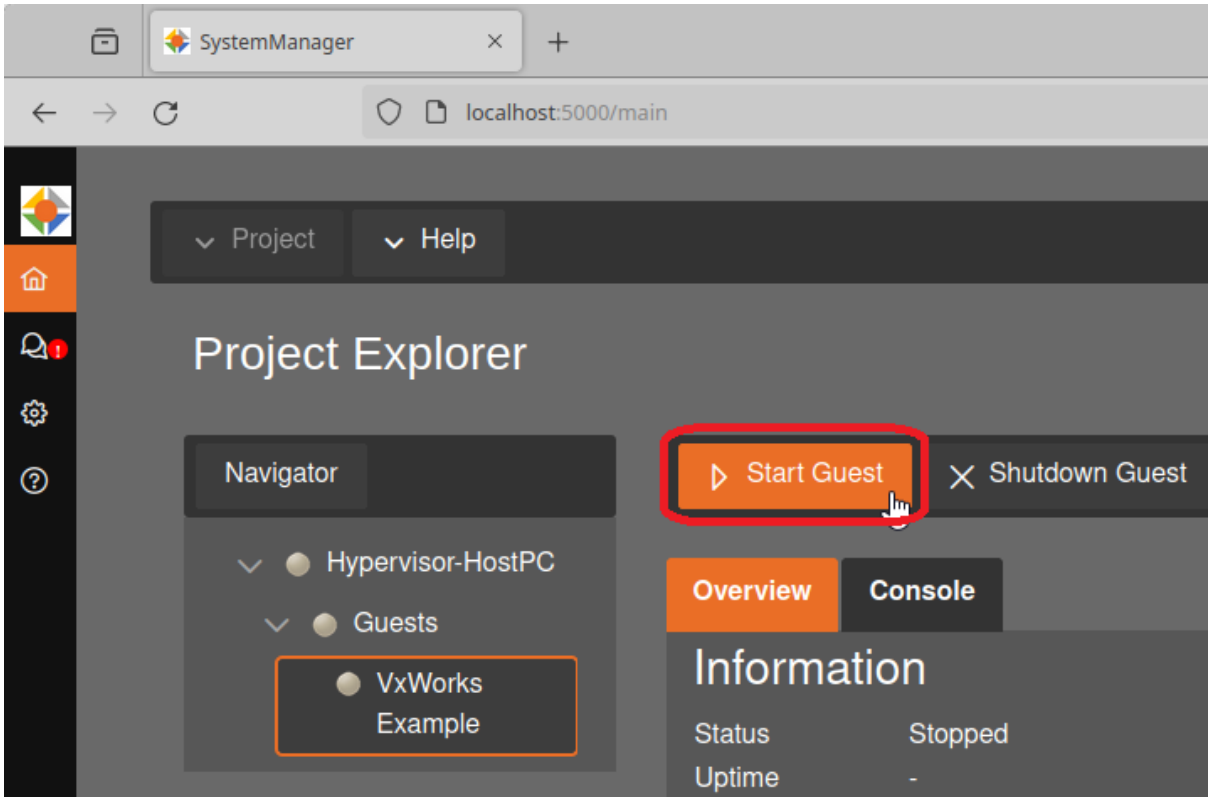
With this the configuration is loaded and the VxWorks guest can be launched.

Change to `Run` mode.

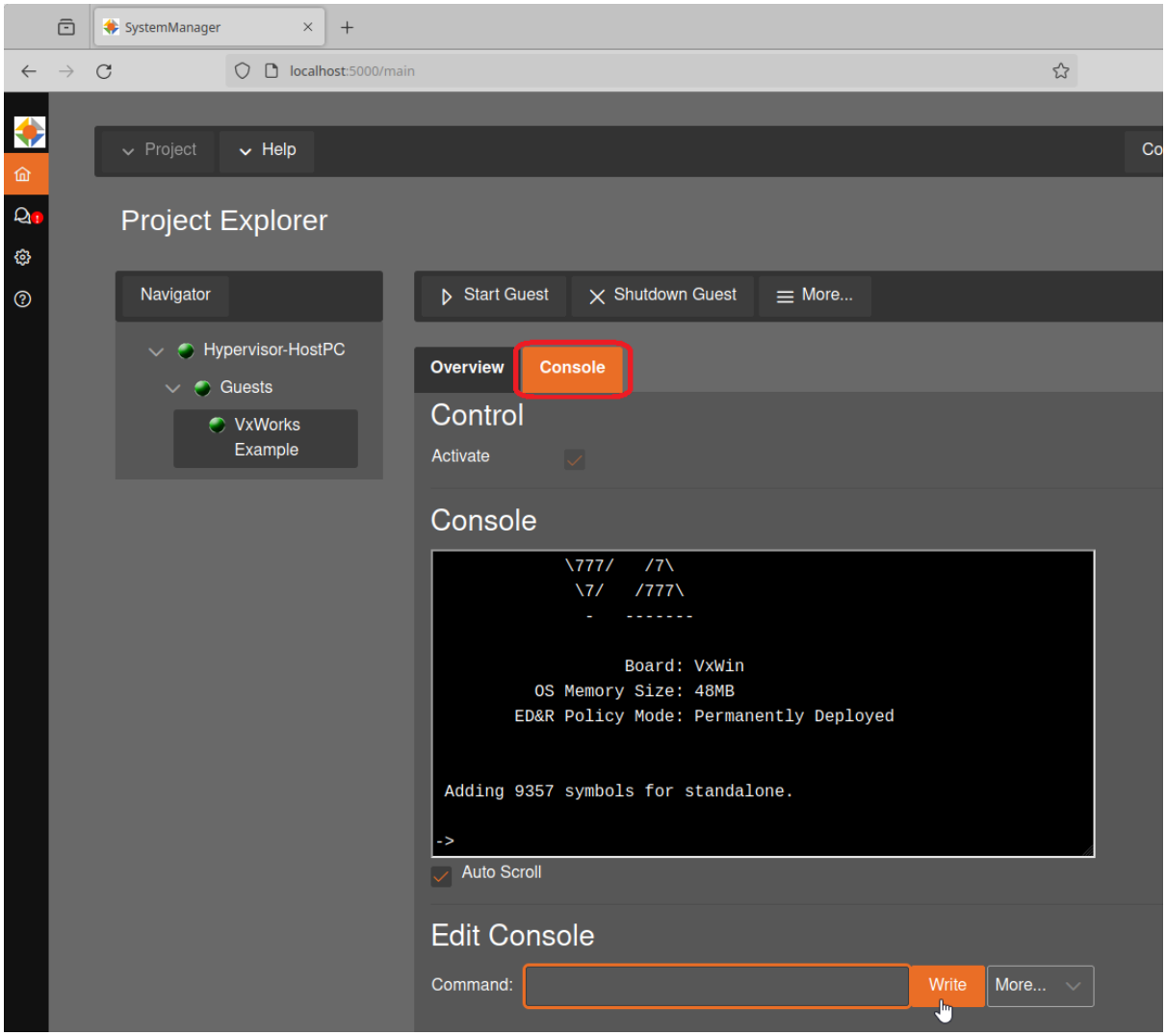
Start the *Virtual Machine Framework (VMF)*. Starting the VMF will load the configuration for **all** guests. Real-time guests can **only** be launched when the VMF is running.



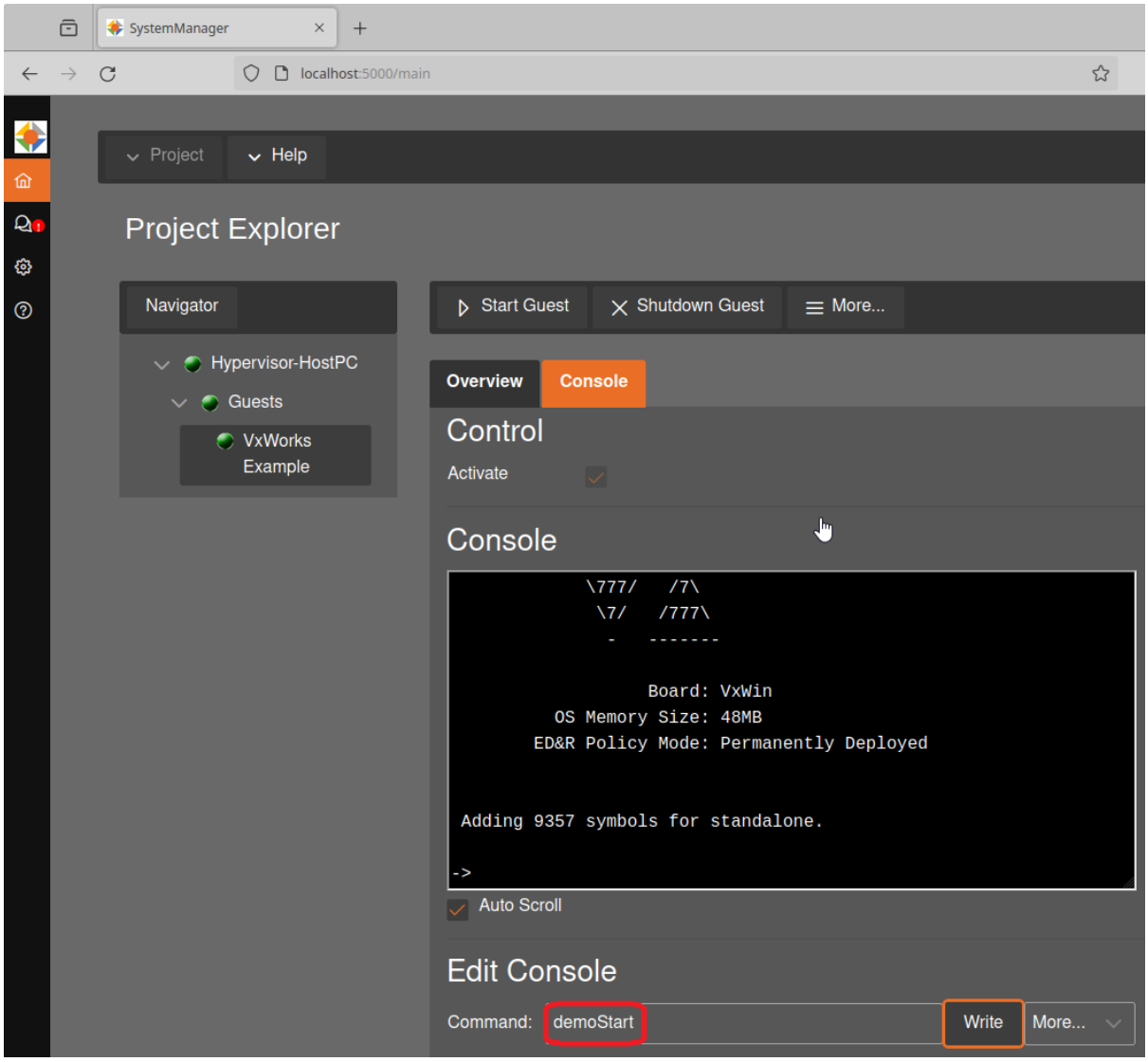
Select the `VxWorks Example` guest in the `Navigator` on the left side.
Click the `Start Guest` button.



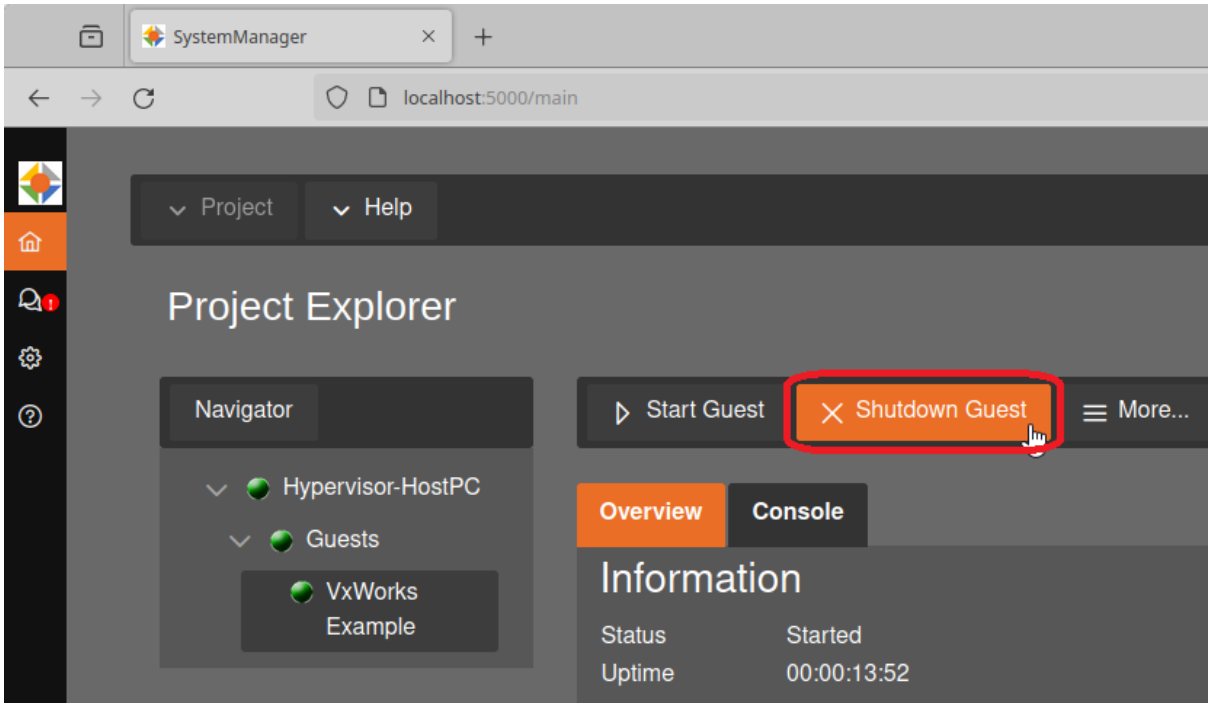
Click the `Console` button to access the console view, displaying the output from the VxWorks guest. Commands can be entered in the `Command` field. The `RealtimeDemo` can be started with the `demoStart` command.



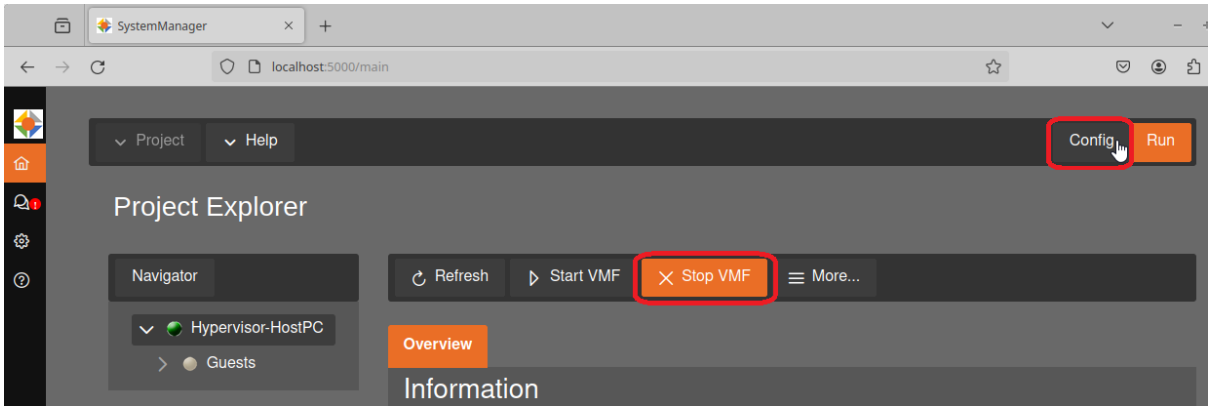
Enter `demoStart` to start the real-time demo application.



To finish the demo application, click on `Shutdown Guest`.



Once the guest is stopped, select the Hypervisor-HostPC on the `Navigator` tab, stop the VMF and switch back to `Config` mode.



3.4 Command Line operation

This chapter demonstrates how to use the example projects using the command line instead of the System Manager.

- **Real-time Linux**:

  When starting a Real-time Linux guest, you will have to log in first, open the example project and synchronize it.

  Open a terminal and enter the following:

  ```
  $ hv_open_example  rt-linux
  $ hv_sync_example  rt-linux
  ```

  Reboot if the console messages ask you to.

```
$ cd /hv/guests/guestrtlinux
$ hv_guest_start -view
```

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

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

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
```

- **On Time RTOS-32**:

  In RTOS-32 the demo is started automatically and the output is displayed in the console window. Adjust the guest configuration setting to prepare starting the *RTOS-32Demo*.

  Initialize the RTOS-32 example

  ```
  $ hv_open_example rtos-32
  $ hv_sync_example rtos-32
  ```

  Reboot if the console output asks you to.

  ```
  $ cd /hv/guests/guestrtos32
  $ mousepad usr.config
  ```

  Add the following lines to modify the fileserver path directing to the guest directory.

  ```
  [Host\FileServer]
      "HomeDir"="/hv/guests/guestrtos32"
  ```

  Save your changes and switch back to the console. Update the link to the demo application:

  ```
  $ rm -f rtos32app.dlm
  $ ln -s /hv/templates/example_guests/rtos-32/files files
  $ ln -s /hv/guests/guestrtos32/files/RTOS-32Demo.dlm rtos32app.dlm
  ```

  Run the demo:

  ```
  $ cd /hv/guests/guestrtos32
  $ hv_guest_start -view
  ```

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

  To run a different example the link to rtos32app.dlm must be changed. To run the real-time demo, adjust the link as follows:

  ```
  $ cd /hv/guests/guestrtos32
  $ hv_guest_stop
  $ rm rtos32app.dlm
  $ ln -s /hv/guests/guestrtos32/files/RealtimeDemo.dlm rtos32app.dlm
  ```

  Run the demo:

```
$ cd /hv/guests/guestrtos32
$ hv_guest_start -view
```

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

- **VxWorks**:

  Initialize the VxWorks example.

  ```
  $ hv_open_example vxworks
  $ hv_sync_example vxworks
  ```

  Reboot if the console output asks you to.

  ```
  $ cd /hv/guests/guestvxworks
  $ hv_guest_start -view
  ```

  Run the demo:

  ```
  $ demoStart
  ```

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

Finally, stop the Real-time guest OS:

- **Real-time Linux**:

  ```
  $ cd /hv/guests/guestrtlinux
  $ hv_guest_stop
  ```

- **On Time RTOS-32**:

  ```
  $ cd /hv/guests/guestrtos32
  $ hv_guest_stop
  ```

- **VxWorks**:

  ```
  $ cd /hv/guests/guestvxworks
  $ hv_guest_stop
  ```

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

```
$ cd /hv/guests/guestrtlinux
$ hv_guest_restart
```

4 Ethernet Device Assignment

In this quickstart guide an Ethernet device will be assigned to the RTOS.

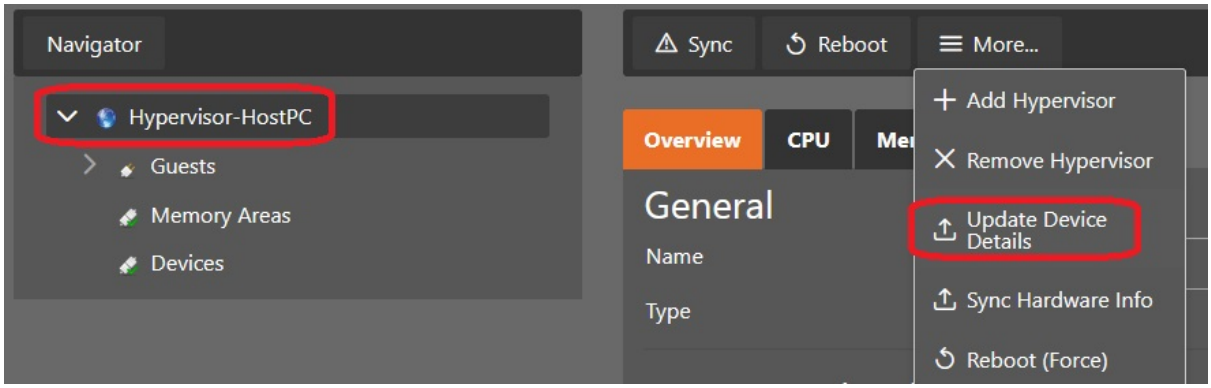
Hint: For more specific options like *polling mode* or *interrupt mode* please check the chapter **RTOS Devices (Partitioning)** of the Hypervisor Manual.

4.1 Determine the desired adapter

If your computer has multiple Ethernet adapters, you will have to determine the right one.

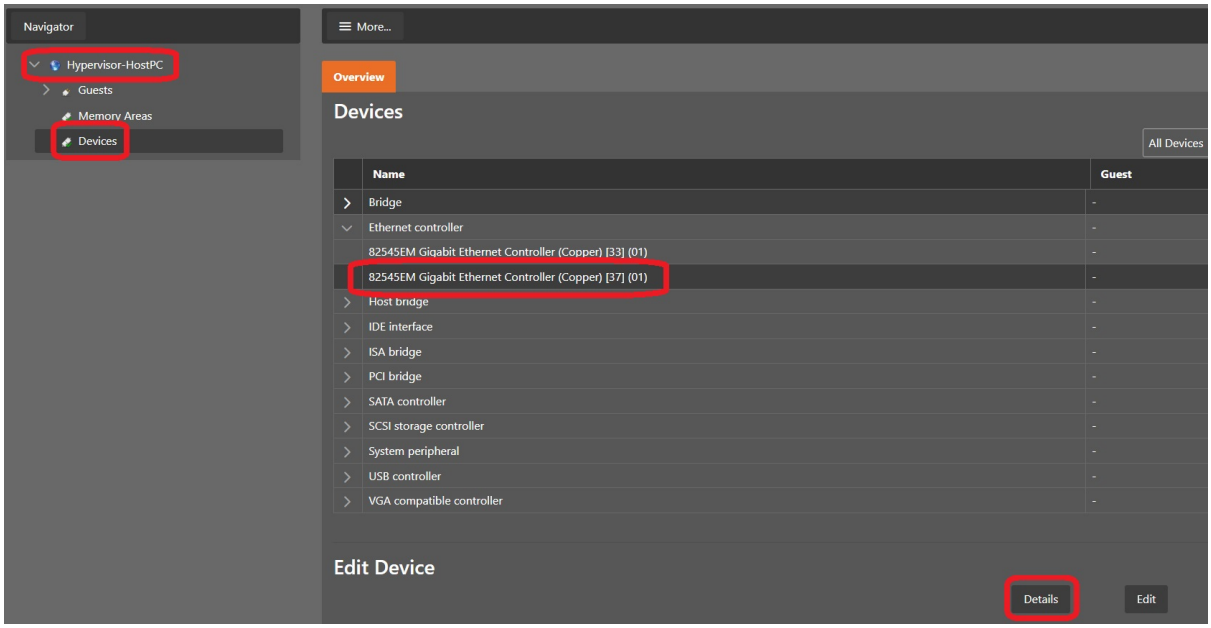
In a first step, please physically disconnect the Ethernet cable from the Ethernet device which shall be assigned to the RT-Linux guest.

Now we need to update the device information (in case the cable had been connected before). Select the Hypervisor Host (e.g. *Hypervisor-HostPC*) and press the *Update Device Details* entry below the *More...* combo-box.

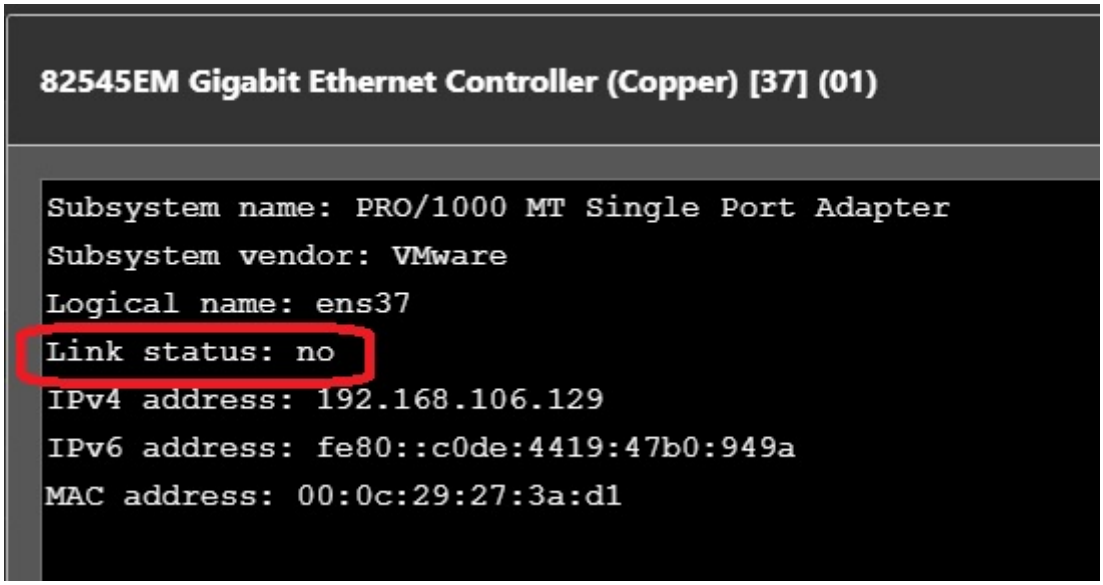


After updating the device information, select the *Devices* tab in the tree view on the left side (do **not** select the *Devices* tab in the rtlinux guest!).

Then unfold the *Ethernet controller* entries, select one of the Ethernet controllers and press the *Details* button.



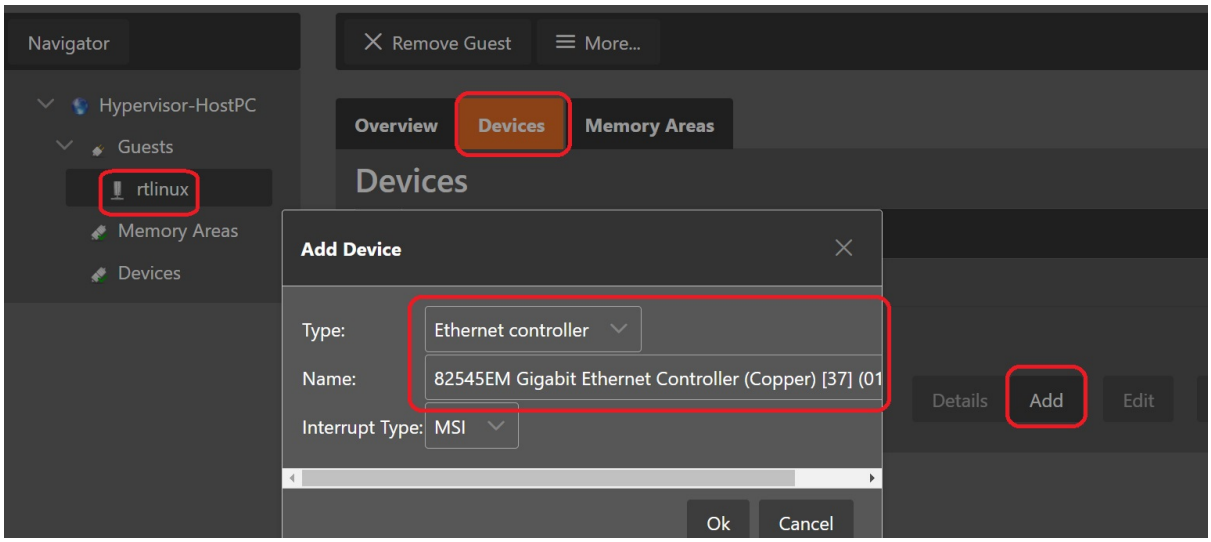
You will see various information about this Ethernet controller, one of these informations is the **Link status**. If it is set to **no**, no cable is connected.



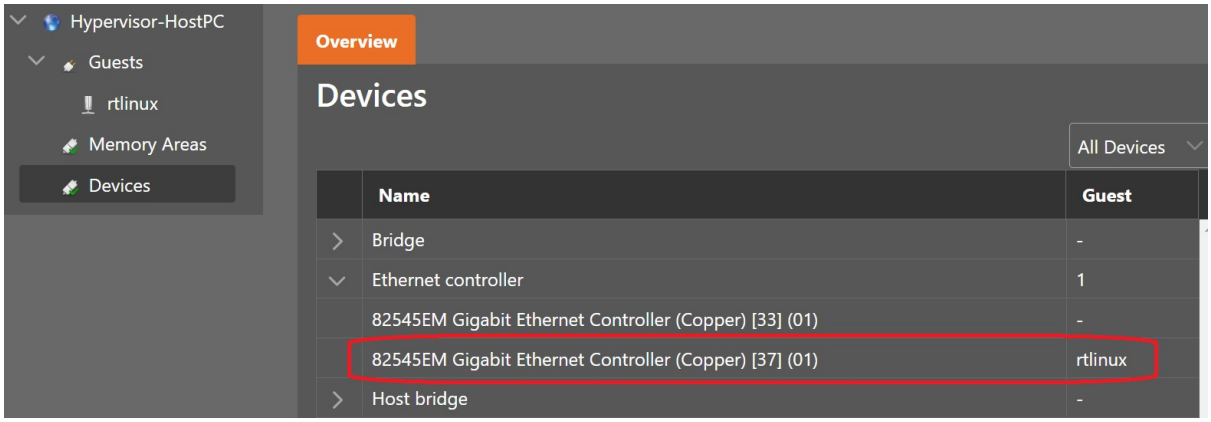
Then you should physically connect the Ethernet cable to the adapter and assure it is connected to a powered on switch. Update the device details again and check if the **Link status** now has changed to **yes**. If this is the case, you have correctly determined the desired adapter, remember its name for later.

4.2 Assign the desired adapter to the guest

Now you need to select the rtlinux guest again, switch to the *Devices* tab in the guest view, press the *Add* button and select the Ethernet adapter you want to assign to Real-time Linux.



Verify the result by selecting selecting the Hypervisor Host (e.g. *Hypervisor-HostPC*) and switch to the *Devices* tab.



In a final step, you must again synchronize your changes, by pressing the *Sync* button in the Hypervisor Host section.

4.3 Command Line assignment

This chapter demonstrates how to assign the Ethernet controller using the command line instead of the System Manager.

4.3.1 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.

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`.

4.3.2 Device Assignment

Run the following command to assign the above device. The first parameter `enp2s0` is the name of the device how it is found in the Hypervisor host. The second parameter `rtos_eth1` has to be a unique RTOS guest related device name. The third parameter `1` is a unique number (to be increased sequentially) for each assigned device, starting with `1`.

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

4.3.3 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.

```
$ mousepad /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 `$HV_BIN/hvpart.sh` 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
```

4.3.4 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:

```
    :          :          :          :          :          :          :          :
↪      :          :
    :          :          :          :          :          :          :          :
↪      :          :
01:00.0 Ethernet controller: Intel Corporation I210 Gigabit Network
↪Connection (rev 03)
```

(continues on next page)

(continued from previous page)

```
        Subsystem: Intel Corporation I210 Gigabit Network Connection
        Kernel driver in use: igb
        Kernel modules: igb
02:00.0 Ethernet controller: Intel Corporation I210 Gigabit Backplane␣
↪Connection (rev 03)
        Subsystem: Intel Corporation I210 Gigabit Backplane Connection
        Kernel driver in use: pci-stub
        Kernel modules: igb
    :          :        :          :        :          :        :          :  ␣
↪      :          :
    :          :        :          :        :          :        :          :  ␣
↪      :          :
```

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*).

4.3.5 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>`.

Hint: Initially, the Hypervisor Host does not provide any example guest folders. To switch to an example guest, you must execute the corresponding initialization. For instructions on how to initialize the examples, refer to the chapter **RTOS Guests** in the Hypervisor Manual.

- **RTOS-32 guest**:

  ```
  $ mousepad /hv/guests/guestrtos32/usr.config
  ```

- **Real-time Linux guest**:

  ```
  $ mousepad /hv/guests/guestrtlinux/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
;-------------------------------------------------------------------------

#include "/hv/config/rtos_eth1.config"

;-------------------------------------------------------------------------
; End of file
;-------------------------------------------------------------------------
```

4.3.6 Reboot Hypervisor Host

```
$ sudo reboot now
```

5 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.

5.1 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 **aren't** sufficent!

```
$ mousepad /hv/guests/etc/rt-linux/files/ecat/noeni.sh
```

respectively

```
$ mousepad /hv/guests/etc/rt-linux/files/ecat/real.sh
```

5.2 Step 1: Scan the EtherCAT network

Start the RT-Linux example guest and open the console view as described in Chapter *RT-Linux Guest Example*.
Or start the guest from the command line.

```
$ cd /hv/guests/guestrtlinux
$ hv_guest_start -view
```

Login to the RT-Linux guest and 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
```

5.3 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
```

5.4 Running EcMasterDemo on the Host

The EcMasterDemo could be started on the host, because host kernel is built with real-time capabilities.

At first you should unbind network adapter from the host OS. Example:

```
$ echo 0000:00:19.0 > /sys/bus/pci/drivers/e1000e/unbind
```

For more details about using of unbind command and determining adapter instance, please refer to “EC-Master Class B” documentation.

Load `atemsys` kernel module:

```
$ sudo insmod $HV_BIN/atemsys.ko
```

Run EcMasterDemo from `/hv/guests/etc/rt-linux/files/ecat/` directory:

```
$ cd /hv/guests/etc/rt-linux/files/ecat/
$ sudo LD_LIBRARY_PATH=$HV_BIN:$LD_LIBRARY_PATH ./noeni.sh
```