DRIVING EMBEDDED EXCELLENCE

ETAS RTPC V6.5.2

User Guide

www.etas.com

ETAS RTPC V6.5.2 - User Guide R01 EN - 03.2022

Contents

1 About this Document

This document is intended, on the one hand, for all users who install, maintain and uninstall ETAS RTPC on the target PC, and, on the other, for system administrators who make ETAS RTPC available on a file server for installation via the network. It provides important information on the delivery scope, hardware and software requirements for single station and network installation as well as on how to prepare the installation.

This document consists of the following chapters:

- “About this Document” on page 5

  This chapter
- “Introduction” on page 8

  This chapter contains information on the system concept, the target group and the data protection declaration.
- “Installation” on page 11

  This chapter contains information on system requirements and on how to install ETAS RTPC including details of the environment configuration.
- “Operation” on page 25

  This chapter describes the operating modes of ETAS RTPC, as well as the creation and migration of projects for ETAS RTPC.
- “Configuration” on page 49

  This chapter describes the uses and configuration of ETAS RTPC with the web interface.
- “Troubleshooting” on page 64

  This chapter presents the description of some typical problems with possible solutions.
- “Glossary” on page 67

1.1 Classification of Safety Messages

The safety messages used here warn of dangers that can lead to personal injury or damage to property:

DANGER

indicates a hazardous situation with a high risk of death or serious injury if not avoided.

WARNING

indicates a hazardous situation of medium risk, which could result in death or serious injury if not avoided.

CAUTION

indicates a hazardous situation of low risk, which may result in minor or moderate injury if not avoided.

NOTICE

indicates a situation, which may result in damage to property if not avoided.

1.2 Presentation of Instructions

The target to be achieved is defined in the heading. The necessary steps for his are in a step-by-step guide:

Target definition

1. Step 1
2. Step 2
3. Step 3

> Result

1.3 Typographical Conventions

Software

`OCI_CANTxMessage msg0 =`	Code snippets are presented in the Courier font. Meaning and usage of each command are ex-plained by means of comments. The comments are enclosed by the usual syntax for comments.
Choose **File** → **Open**.	Menu commands are shown in boldface.
Click **OK**.	Buttons are shown in boldface.
Press <ENTER>.	Keyboard commands are shown in angled brackets in small caps.
The "Open File" dialog box is displayed.	Names of program windows, dialog boxes, fields, etc. are shown in quotation marks.
Select the file `setup.exe`.	Text in drop-down lists on the screen, pro-gram code, as well as path- and file names are shown in the Courier font.
A *distribution* is always a one-dimensional table of sample points.	General emphasis and new terms are set in italics.

1.4 Presentation of Supporting Information

NOTE

Contains additional supporting information.

2 Introduction

ETAS RTPC transforms your standard PC into a high-performance simulation target. As ETAS RTPC consistently uses standard commercial components, you benefit from the continuous development of PC technology with regard to the available processing power and communication bandwidth. ETAS RTPC uses all processor cores of state-of-the-art Intel® multi-core processors.

2.1 System Concept

2.1.1 "LABCAR Test Systems" Use Case

Once the ETAS RTPC software has been installed, your standard PC is transferred into a real-time PC for LABCAR test systems.

Fig. 2-1 shows the hardware setup of a ETAS RTPC system. The real-time PC is connected to the user PC via Ethernet and can be selected as a simulation target in the LABCAR software.

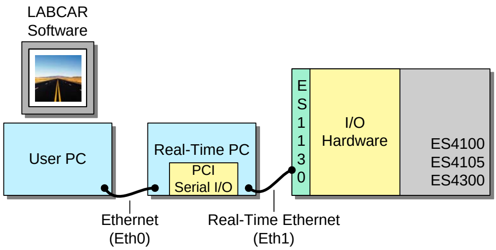


Fig. 2-1 RTPC System

The real-time PC is controlled entirely from the user PC via an intuitive Web interface. Keyboard and monitor are not needed (apart from during installation).

The I/O hardware communicates with the real-time PC via Ethernet – the I/O data gets to its target deterministically and securely via a highly optimized point-to-point connection.

A PC with ETAS RTPC is also used as a simulation PC in the open and scalable ETAS HiL system PT-LABCAR which is designed for closed-loop tests of ECUs in the drive train area.

2.1.2 "EVE - Validation of AUTOSAR Software" Use Case

ISOLAR-EVE allows to validate AUTOSAR compliant software on different platforms. ETAS RTPC is one of those platforms.

ETAS RTPC allows to create and operate a VECU (virtual electronic control unit) with access to external hardware resources like CAN, DIO, ADC, PWM.

ISOLAR-EVE runs on a user PC and operates the real-time PC mainly via automation interfaces in order to build the executable and to control the execution.

General configuration like licensing and monitoring may be done via ETAS RTPC's intuitive Web interface.

2.2 Target Group

This document addresses qualified personnel working in the fields of automobile control unit development and calibration. Specialized knowledge in the areas of measurement and control unit technology is required.

Users should be familiar with the Microsoft Windows XP, Windows Vista, or Windows 10 operating system. Furthermore, the users should be familiar with the Windows file storage system, especially the connections between files and directories.

The users have to know how to use the basic functions of the Windows File Manager and Program Manager or the Windows Explorer, respectively, and they should be familiar with the "drag-and-drop" functionality.

ISOLAR-EVE users who are not yet familiar with automotive software descriptions based on the AUTOSAR framework are advised to get a suitable training upfront.

Knowledge about the general features and operation of Eclipse based software products will be beneficial but not mandatory.

2.3 Data Protection Declaration

2.3.1 Data Processing of ETAS RTPC

Please note that personal or person-related data or data categories are processed when using ETAS RTPC.

NOTE

The purchaser of ETAS RTPC is responsible for the legal conformity of processing the data in accordance with Article 4 No. 7 of the General Data Protection Regulation (GDPR).

As the manufacturer, ETAS GmbH is not liable for any mishandling of this data.

2.3.2 Data and Data Categories

When using the ETAS License Manager in combination with user-based licenses, particularly the following personal or person-related data or data categories are recorded for the purposes of license management:

```
Communication data: IP address
User data: User ID
```

2.3.3 Technical and Organizational Measures

ETAS RTPC does not encrypt the collected personal or person-related data.

Please ensure the data security of the recorded data by suitable technical or organizational measures of your IT system, e.g. by classical anti-theft and access protection on the hardware.

3 Installation

This chapter includes information on the following topics:

3.1 Delivery Scope

The delivery scope of ETAS RTPC V6.5.2 includes

- ETAS RTPC Software DVD
- This User Guide
- Possibly a PC from ETAS

3.2 System Requirements

You need the following to install and run the ETAS RTPC software:

- ETAS RTPC Software CD
- ETAS RTPC software license
- Suitable PC hardware (see "“Hardware Requirements” on page 12")
- Ethernet crossover cable (2x RJ45 socket) for the connection between the real-time PC and user PC

3.2.1 Hardware Requirements

ETAS RTPC is compatible with most standard PCs. To run the software, the following requirements must be fulfilled:

Processor	Intel® Xeon® family Intel® Core™ i family Intel® Core™ 2 family (Duo and Quad)
Ethernet adapter for the connection to the user PC	All Ethernet adapters supported by Linux kernel 3.18
Ethernet adapter for the connection to the ES1130	All Ethernet adapters supported by Linux kernel 3.18 For example - Intel® PRO/100 card S, revision 12 (0C) or later - Intel® PRO/1000 family - Intel® PRO/1000 PT Quad Port Server Adapter - Intel® Gigabit CT Desktop Adapter
RAM	Min. 4 GB, 8 GB and more recommended
Hard disk	IDE or SATA, min. 250 GB
Graphic	VGA-compatible
Monitor	Is only required for basic installation
Keyboard	PS/2 or USB, is only required for basic installation
CD-ROM drive	IDE, SATA or USB, is only required for basic installation (bootable)
Operating system	Not necessary

NOTE

During ETAS RTPC operation, success or error messages are emitted in the form of sound. If your PC does not have an output device for acoustic signals, you can access information on the current status via the web interface (see section “The Web Interface” on page 51).

For more precise details on the compatibility of PCs made by different manufacturers, refer to the compatibility list (http://192.168.40.14/documentation/).

3.2.2 Hardware Support

ETAS RTPC supports a maximum number of hardware connections related to the ES5300 series.

NOTE

Before you design or configure a system with the specified maximum number of hardware connections, get in touch with your ETAS contact person. The ETAS contact person supports you to select the individual hardware.

The following table contains the maximum number of the hardware connections:

Hardware	Maximum Number
CAN Boards	32
Network Ports	32
FlexRay Boards	8
RS232 Ports/RS485 Ports	16

If you extend the RS232/RS485 port number to 16, a change of the port order is possible. If you use these ports, verify the hardware configuration and change it accordingly.

3.2.3 Special BIOS Settings

3.2.3.1 Boot Settings

To ensure the installation from CD-ROM works, the system's BIOS has to be configured so that the PC boots from the CD-ROM ("primary boot device = CD-ROM"). If you want to use the PC later without a keyboard, you also have to specify this setting ("no halt on keyboard"). The names of the settings in brackets may vary. For more details, refer to your PC's User Guide.

NOTE

For details of the ETAS PC hardware BIOS settings, refer to http://192.168.40.14/documentation/.

3.2.3.2 Hyper-Threading

The usage of hyper-threading has an impact on the real-time performance of the used real-time PC as two hyper-threaded cores share some CPU resources. For example, the FPU and the ALU are typically shared among two hyper-threaded cores.

This will affect the execution of the code running on the hyper-threaded cores, because one hyper-threaded core may have to wait for the other hyper-threaded core to release a shared CPU resource.

If you have issues with real-time violations, it might be helpful to deactivate hyper-threading in the BIOS settings of the used real-time PC. Especially if you are running simulation models with very fast cycle times (faster than 100 µs), it is recommended to disable hyper-threading completely as the additional time for waiting for CPU resources may break the cycle times.

However, there are some situations where hyper-threading can be used without getting real-time violations. In these cases, you may benefit from the additional CPU power.

Typical situations for this are setups that are using simulation models with moderate cycle times (slower than 500 µs) or also for real-time PCs that are used as build machines only. In these situations, the higher overall computation power can be used beneficially.

3.3 Installation

The installation of ETAS RTPC is divided into four parts which are each described in the sections below:

- “Installing ETAS RTPC on the Real-Time PC” on page 14
- “Installing ETAS RTPC in a Virtual Machine” on page 16
- “Hardware Integration” on page 16
- “System Configuration” on page 17
- “Licensing” on page 18
- “Resetting the Boot Setting” on page 19
- “Cloning Installed Versions” on page 19
- “RTIO Configuration of the ES1130” on page 21
- “Clock Synchronization” on page 21

NOTE

ETAS also offers ETAS RTPC as a preinstalled version on PC hardware which is recommended and tested by ETAS. In this case, please start by reading the section “Hardware Integration” on page 16.

3.3.1 Installing ETAS RTPC on the Real-Time PC

If ETAS RTPC has already been installed by ETAS, please continue with the next section “Hardware Integration” on page 16.

To install ETAS RTPC on the real-time PC

1. Connect the monitor and keyboard to the PC hardware.
2. Switch on the PC and the monitor.
3. Insert the ETAS RTPC Software DVD into the real-time PC’s DVD drive.
4. In the start screen select the desired option (default: `ETAS RTPC Vx.y.z – Standard Install`).

   NOTE

   With this option the file system "ext4" is installed – if you have to be compatible with ETAS RTPC versions before V5.1.0., then select the "Install for parallel use with RTPC versions before V5.1.0" option.

5. Wait until the ETAS RTPC installation dialog box is displayed.
6. Select the option `[Install RTPC software]` using the cursor keys.
7. Press <ENTER>.

   NOTE

   All data on the hard disk is deleted by the ETAS RTPC installation!

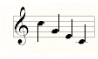
8. Confirm the warning by selecting the option `[Yes, continue with installation]`.
9. Wait until the ETAS RTPC installation has been completed (takes about 5 minutes).
10. Remove the ETAS RTPC Software DVD.
11. Select the option [`Reboot`].

    The PC reboots.

12. Wait until the PC plays an ascending sequence of notes.

    ETAS RTPC has been installed successfully.

13. Shut down the PC by quickly pressing the on/off switch.



    After a few seconds, the PC plays a descending sequence of notes and shuts down.

    This indicates that ETAS RTPC has been shut down correctly and can be disconnected from the mains.

    For more details on these acoustic signals, refer to the chapter "Operation" on page 25.

NOTE

ETAS RTPC uses a robust file system, but hardware and software can still be damaged when powered off without shutting down or during a sudden voltage drop. ETAS thus recommends shutting down ETAS RTPC by briefly pressing the on/off switch to avoid these problems.

14. Remove monitor and keyboard (see "Special BIOS Settings" on page 13).

To test the reachability of ETAS RTPC

1. Toggle to the host.
2. Select **Run** from the Start menu.
3. Enter "cmd".

   The command prompt opens.
4. Enter the following:

   `ping 192.168.40.14`

```
U:\>ping 192.168.40.14

Pinging 192.168.40.14 with 32 bytes of data:

Reply from 192.168.40.14: bytes=32 time=3ms TTL=128
Reply from 192.168.40.14: bytes=32 time<10ms TTL=12
Reply from 192.168.40.14: bytes=32 time<10ms TTL=12
Reply from 192.168.40.14: bytes=32 time<10ms TTL=12

Ping statistics for 192.168.40.14:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  3ms, Average =  0ms
```

5. Alternatively, you can open the web interface (`http://192.168.40.14`) in your browser.

3.3.2 Installation on EFI Hardware

ETAS RTPC V6.5.2, V 6.5.1, V6.5.0, V6.4.1 and V6.4.0 support EFI boot. When installed on an EFI hardware (e.g. ES5300.2) it is not recommended to install versions older than ETAS RTPC V6.4.0 in parallel. The installation from inside ETAS RTPC V6.4.0 and higher will work but the older versions are not aware of the EFI boot.

NOTICE

Installing another older version of ETAS RTPC from a version prior to ETAS RTPC V6.4.0 will damage the RTPC.

3.3.3 Installing ETAS RTPC in a Virtual Machine

The ETAS RTPC software can be installed in a virtual machine. Use the installation image of the delivered DVD for this. The virtual machine must be configured according to the system requirements of the ETAS RTPC (see “System Requirements” on page 11"). For the configuration and installation of the virtual machine, please refer to the instructions of the corresponding tool.

To operate the ETAS RTPC in a virtual machine, ETAS recommends the use of a "host-only" interface as network interface eth0. The steps for the configuration of such a network interface can be found in the documentation of the corresponding tool.

3.3.4 Hardware Integration

This section describes the integration of the real-time PC into a Hardware-in-the-Loop test system (HiL). Other configurations such as Model-in-the-Loop or HiL system with measuring and calibration access are described in the section “Applications” on page 49.

ETAS RTPC uses the identifier Eth0 (Host) for the onboard Ethernet interface – the Ethernet board has the identifier Eth1 (HW).

To connect the user PC and the ES1130

1. Connect the real-time PC to the experimental target interface of the user PC using the onboard Ethernet interface (Eth0) with a crossover cable.

> **NOTE**
>
> Crossover cables have crossover connections and are often indicated on one or both ends with yellow or orange connectors.

2. Connect the real-time PC to the ES1130 Simulation Controller Board using the Ethernet interface (Eth1). The ETAS CBE100.1 cable (for ES1130.2 and ES1130.3) or K107 (for ES1130.1) can be used for this connection.

3.3.5 System Configuration

ETAS RTPC is easy to configure from the user PC via a Web browser (e.g. Internet Explorer or Mozilla Firefox) – the configuration dialog can be accessed via the IP address 192.168.40.14.

Carry out the steps described below to configure ETAS RTPC for Hardware-in-the-Loop tests (HiL). For more details on configuration modes, refer to the section “Applications” on page 49.

To open the configuration dialog

1. Launch the real-time PC by pressing the on/off switch on the front.
2. Switch on the chassis with the I/O hardware.

3. Wait a few seconds until you hear the ascending sequence of notes.
4. Launch a Web browser on the user PC.
5. Invoke the URL http://192.168.40.14.

> **NOTE**
>
> You can also assign an alias to this IP address, which is easier to remember – for more details, refer to the section “Access to the Web Interface” on page 51.

The ETAS RTPC configuration dialog opens.

The simulation controller has to be stopped to configure the ETAS RTPC Software (see “"Simulation Controller" Mode” on page 25).

> **NOTE**
>
> As long as the simulation controller is running, the *Main Page* link is inactive!

To stop the simulation controller

1. Click the **Stop Simulation Controller** button.

2. Click the **Main Page** link.

To run the configuration

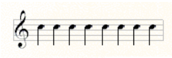
1. Click the **Configure ETAS RTPC** link.
2. Click the **Autoconfigure** button.

   The autoconfiguration is run and the result is set in the "RTP-C_USAGE_ETH1" box.

To restart the simulation controller

1. Click the **Main Page** link.
2. Click the **Simulation Controller** link.
3. Click the **Start Simulation Controller** button.

   When you hear the ascending sequence of notes, the simulation controller is active again.



NOTE

If you hear the sequence of error notes instead, the simulation controller could not be started. For more details, refer to the chapter “Troubleshooting” on page 64.

The configuration for HiL mode has now been carried out successfully. For more details on other configuration modes refer to the section “Troubleshooting” on page 64.

3.3.6 Licensing

To operate ETAS RTPC V6.5.2, you require a license which can be obtained in the form of a license file from the ETAS License Portal. For more details, please refer to the manuals "LABCAR-OPERATOR V5.4.13 - Getting Started" or "ETAS ISOLAR-EVE Vx.y[1] - Installation Guide" or "COSYM V2.5.0 - Installation Guide".

NOTE

The license for ETAS RTPC is not installed using the ETAS License Manager – as described above – but in the web interface (see “License Management” on page 62 and “The Web Interface” on page 51).

License Versions

- RTPC_EVE

  To execute ISOLAR-EVE projects
- LCS_RTPC_RT

  To execute LABCAR-OPERATOR projects in real time
- COSYM_RT

  To execute the build process of a COSYM HiL system and to view the real-time execution in the ETAS Experiment Environment.

1. The corresponding version of ISOLAR-EVE had not yet been determined at the time of the ETAS RTPC V6.5.2release.

- LCS_RTPC_OLC

  When running LABCAR-OPERATOR projects and COSYM projects, the minimum period duration is 200 ms.

NOTE

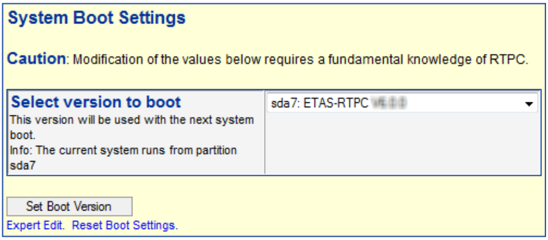
No license is required for the Build process of a LABCAR-OPERATOR project!

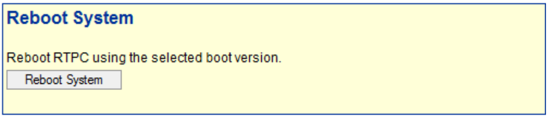
3.3.7 Resetting the Boot Setting

If there are problems when changing the boot settings (e.g. after repartitioning), these settings can be reset.

To reset the boot settings

1. Select **Main Page >> Configuration**.
2. Go to the "System Boot Settings" section.
3. Select a version
4. Click **Set Boot Version**.





5. Reboot the system with **Reboot System**.

3.3.8 Cloning Installed Versions

You can clone versions of ETAS RTPC that have already been installed by generating an installable file (*.tgz) and downloading this file to your operating PC.

To clone a version

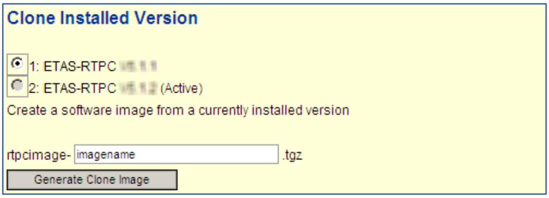
1. Select **Main Page >> Installation / Update >> Clone Installed Version**.
2. Select one of the installed versions.

NOTE

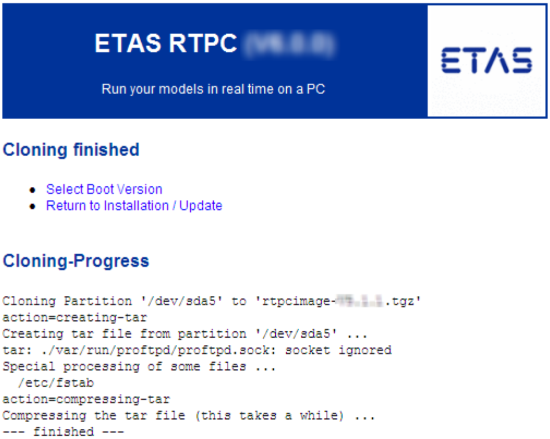
The currently active version cannot be cloned!

3. Enter a name for the clone to be created.

4. Click **Generate Clone Image**.



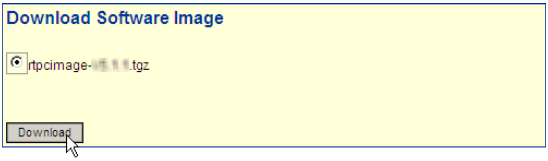
The procedure is started and the progress displayed in the web interface.



Once the procedure has been completed, you can download the clone to use on your operating PC.

<u>To download a clone</u>

1. Click the **Return to Installation / Update** link.
2. Go to "Download Software Image".
3. Select the version just created.
4. Click **Download**.



A download dialog box opens where you can save this file.

5. If you want to delete this file from your real-time PC, select the clone under "Delete Installed Software Images".

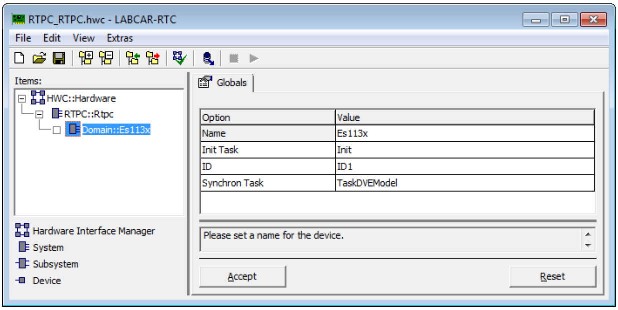
6. Click **Delete**.

> The file is deleted from your real-time PC.

3.3.9 RTIO Configuration of the ES1130

In the case of an ES1130 which is logically connected after a real-time PC, communication between ETAS RTPC and the ES1130 takes place synchronously to a selectable task. This selection is made in LABCAR-RTC (RTIO) in the "Globals" tab of the ES1130 ("Synchron Task").

The communication between ETAS RTPC and ES1130 is optimized to this task. The best results are obtained when the simulation model and the periodic hardware accesses run in exactly one task ("TaskDVEModel").



3.3.10 Configuration of an NTP Server

To use an NTP[1] server the environment variable `RTPC_NTP_SERVER_ADDR` has to be set to the address of the NTP server which is accessible from the RTPC (in most cases the host pc). Please use the "expert edit" of the RTPC web interface to set this environment variable.

3.3.11 Clock Synchronization

To synchronize the time on the user PC and the real-time PC, you can activate the NTP server integrated in Windows.

Activation takes place by making entries in the Windows registry – you will find the file for setting the relevant keys via the web interface at

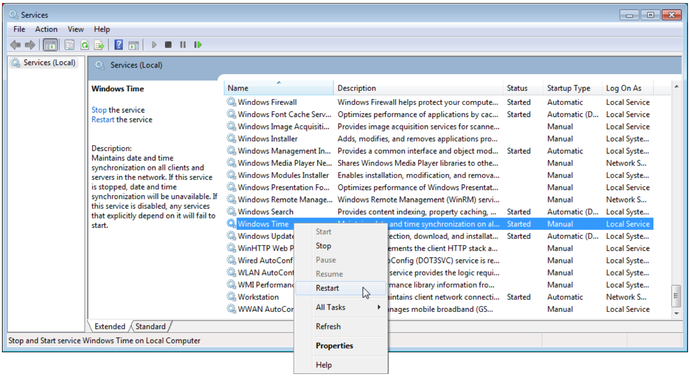
`http://192.168.40.14/tools/NTP/NTP-LocalParameter.reg`

This configures the Windows service "ws32time" – the server must be rebooted after the keys have been modified.

1. Network Time Protocol

To reboot the NTP server

1. Select **Control Panel** → **Administrative Tools** → **Services**.
2. Select the "Windows Time" service.
3. Open the shortcut menu by pressing the right-hand mouse button.
4. Select **Restart**.



> **NOTE**
>
> Please note that when the service has been rebooted, it can take a few minutes until the time is precise enough to be made available to other clients.

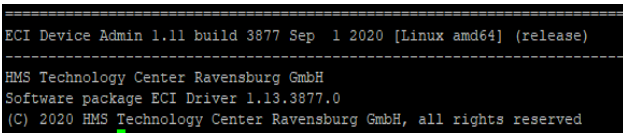
In the web interface, it is possible to carry out an ad hoc synchronization of the real-time PC clock and host clock (see "**Clock Synchronization**" on page 59).

3.4 PCI CAN and FlexRay Boards

For communication between the model and the ECU, you can purchase CAN and FlexRay boards for the ETAS real-time PC.

3.5 CAN/CAN-FD and LIN Boards

ETAS RTPC V6.5.2 uses the version 1.13.3877 of IXXAT® drivers for CAN/CAN-FD and LIN support.



This is a new version and therefore the firmware of the installed IXXAT IB200/ IB600/ IB640 boards has to be updated once. To perform the update please use the RTPC web interface and navigate to
Main Page >> System Info >> LABCAR I/O Hardware.

In the section **Firmware version check for IXXAT Boards** you will find a list of the installed IB200/ IB600/ IB640 boards and the installed firmware. The related button **Update Firmware** is placed below this list. This button will be disabled if no update is required. Fig. 3-1 shows an example for an IXXAT firmware version match.

> **NOTE**
>
> After performing the firmware update of the installed IB200, IB600, and IB640 boards it may not be possible to downgrade to a previous version. In case of questions please contact ETAS.

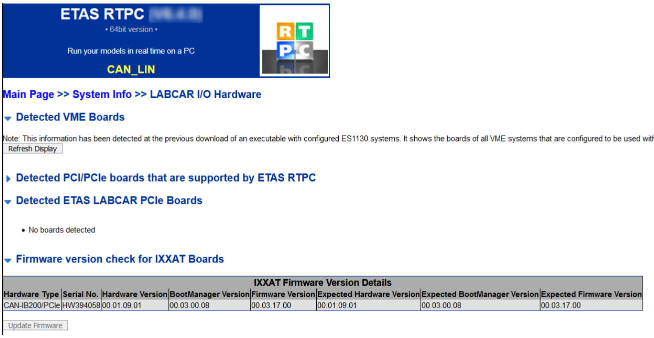


Fig. 3-1 Example for IXXAT Firmware Version match

In case of a firmware version mismatch of IXXAT IB200/ IB600 /IB640 boards, the controllers are not opened during the initialization of the simulation. An error message is written to the log file and no messages will be sent and received. This behavior is identical to the case where a configured controller is not present. Fig. 3-2 shows an example of an error message in the log file and Fig. 3-3 shows an example for a mismatch of an IXXAT firmware version.

```
error] <ixxat-can>ixxat_can_open: Firmware Version mismatch is found for board id 0. Please perform ixxat Firmware update.
error] rtos_comm_open failed for:ixxat_can
error] Error in initializing CAN card for bus 'Bus_0' with CAN-controller 0 of CAN-board 0.
notice] rtos_comm_open (ixxat_can:0/controller=1  btr0=0 btr1=28  stt=0 checkinterval=0.5 initchk=4 )
error] <ixxat-can>ixxat_can_open: Firmware Version mismatch is found for board id 0. Please perform ixxat Firmware update.
error] rtos_comm_open failed for:ixxat_can
error] Error in initializing CAN card for bus 'Bus_1' with CAN-controller 1 of CAN-board 0.
```

Fig. 3-2 Example of an Error Message in the Log File

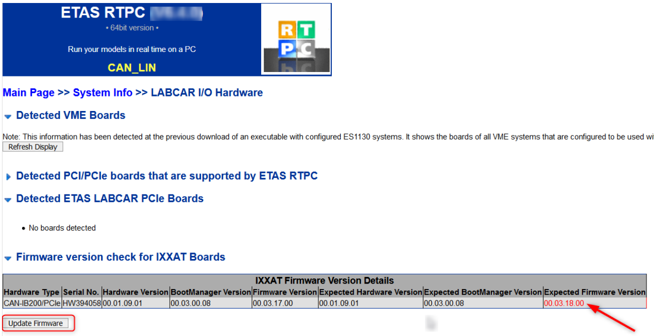


Fig. 3-3 Example for IXXAT Firmware Version Mismatch

4 Operation

This chapter includes information on working with ETAS RTPC V6.5.2.

The individual sections are:

4.1 Operating Modes

ETAS RTPC V6.5.2 has four operating modes, one of which is always active:

- Simulation Controller
- Idle/Configuration
- Ethernet Bridge
- Power Down

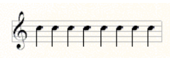
Use the Web interface to configure which operating mode is to be activated when the real-time PC is switched on (see “**RTPC_POWER_UP_MODE**” on page 56).

4.1.1 "Simulation Controller" Mode

The "Simulation Controller" mode is used to run real-time models. An ascending sequence of notes indicates that the simulation controller has been started successfully.

The simulation controller is activated automatically by default when the real-time PC is powered up.

Eight identically pitched notes indicate that the simulation controller cannot be started because of a configuration problem.



In this case, ETAS RTPC is automatically set to "Idle/Configuration" mode (see “"Idle/Configuration" Mode” on page 26).

Possible causes of error can be found in the chapter “Troubleshooting” on page 64.

NOTE

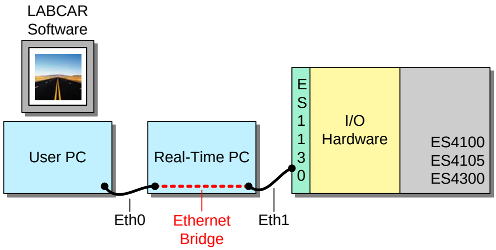
In a LABCAR system, a VMEbus reset takes place every time the simulation controller is started!

4.1.2 "Idle/Configuration" Mode

The "Idle/Configuration" mode is used to configure ETAS RTPC and to update the software (Update). ETAS RTPC cannot run real-time models in this mode.

4.1.3 "Ethernet Bridge" Mode

This mode establishes a transparent connection between the two Ethernet interfaces Eth0 and Eth*n*. This connection enables the user to access the ES1130 Simulation Controller Board directly from the user PC; manual exchanging of Ethernet cables is therefore unnecessary.



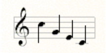
NOTE

The "Ethernet Bridge" mode can be configured separately for each "Ethn" port!

The "Ethernet Bridge" mode is activated automatically for the firmware update of the I/O hardware (HSP or LABCAR firmware update).

4.1.4 "Power Down" Mode

The "Power Down" operating mode is activated either by quickly pressing the on/off switch on the front of the real-time PC or via the Web interface. ETAS RTPC shuts down the system, plays a sequence of notes (four descending notes) and switches the real-time PC off.



The real-time PC can be disconnected from the mains.

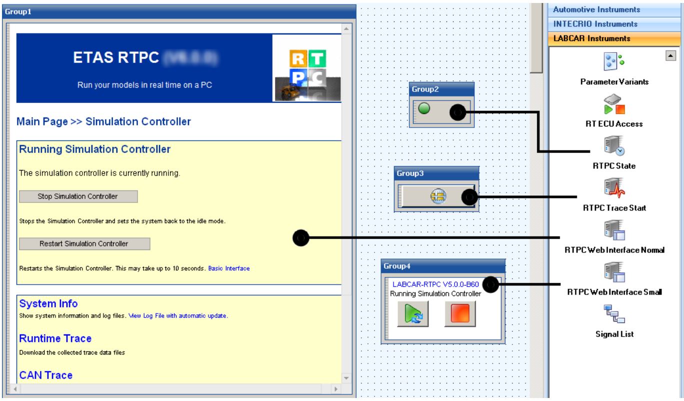
NOTE

ETAS RTPC uses a robust file system, but hardware and software can still be damaged when powered off without shutting down or during a sudden voltage drop. Thus ETAS recommends to shut down ETAS RTPC by briefly pressing the on/off switch to avoid these problems.

4.2 ETAS RTPC in ETAS Experiment Environment

The ETAS Experiment Environment contains a range of instruments that enable access to ETAS RTPC.

These are contained in the Experiment Environment window "Instruments" in the "LABCAR Instruments" group. They can be dragged into the main workspace from there in the usual manner ("Instrumentation" tab).



These are:

- RTPC State

  A running real-time PC is indicated by a green LED
- RTPC Web Interface Normal

  This instrument contains the complete representation of the web interface as in the HTML browser
- RTPC Web Interface Small

  This instrument enables the simulation controller to be started (or restarted) and stopped
- RTPC Trace Start

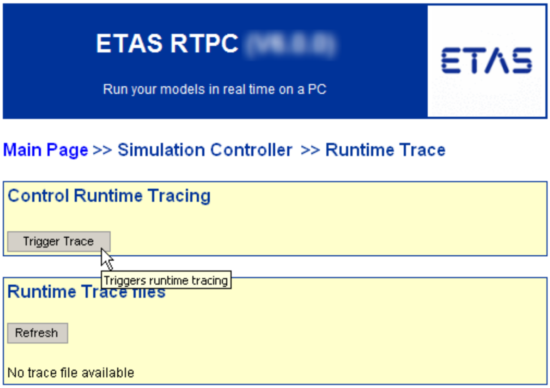
  This is used to start an RTPC trace. The files (and the access to them) are described in the section “Runtime Traces” on page 28.

4.3 Runtime Traces

A runtime trace logs processes and tasks on the real-time PC and saves them in files with the ending .rtt which can be analyzed with RTA-TRACE and saved as a summary in the form of xml, html and csv files.

To run a runtime trace

1. Select **Main Page >> Simulation Controller >> Runtime Trace**.
2. Under "Control Runtime Tracing", click **Trigger Trace**.



> The runtime trace is launched and recorded in a file.

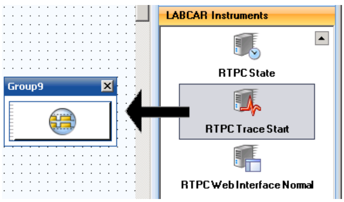
NOTE

The scope of a runtime trace (= the number of recorded events) is determined in the variable "RTOS_TRACE_BUFFERSIZE" (see “**RTPC Configuration**” on page 54).

To start a runtime trace in the Experiment Environment

You can also start runtime traces from the Experiment Environment.

1. To do this, select the instrument "RTPC Trace Start" (from the "LABCAR Instruments" in the "Instruments" window) and drag it into a layer.

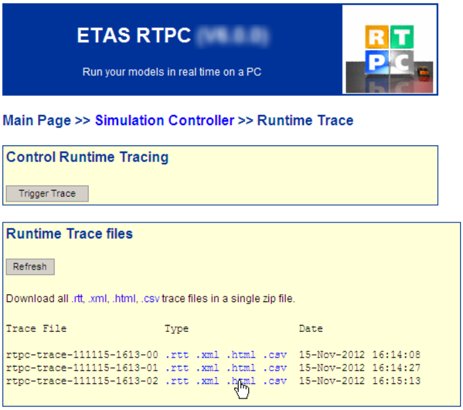


2. To start the runtime trace, click the button.

To open a file with runtime trace

The files with the results of a runtime trace are accessible at the same point:

1. Select a format next to the entry for the relevant file.



   Depending on the format selected, the file is opened or a dialog box for saving the file is opened.

2. To display the HTML file of an RTPC trace, click the file ending in the "Type" column.

   The file opens in the browser.

Process Runtime Statistic 2 - 15-Nov-2012 16:15:13

Task "Config" (CPU core: 0)

Process Name	Id	Calls	RTmin	RTavg	RTmax
[Task "Config"]	**4**	**81**	**6**	**7**	**11**
Es4408loadchassis_ConfigCode_Config_HWCF	10	81	0	0	1

All runtimes in microseconds.

Task "TaskDVEModel" (CPU core: 0)

Process Name	Id	Calls	RTmin	RTavg	RTmax
[Task "TaskDVEModel"]	**8**	**812**	**13**	**28**	**138**
Icrt_OneStep_Outputs_IdleController	17	812	2	4	82
Icrt_OneStep_States_IdleController	19	812	2	3	41
ConversionModule_Execute	16	812	1	3	16
Es113x_SyncReceiveCode_TaskDVEModel_HWCF	11	813	1	3	15
Es4408ctrlStatus_TaskDVEModel_HWCF	12	813	0	1	5
Es4408ctrlPresence_TaskDVEModel_HWCF	13	813	0	1	4
Es4456rbpiezoloadStatus_TaskDVEModel_HWCF	14	813	0	0	4
Es4456rbpiezoloadPresence_TaskDVEModel_HWCF	15	812	0	0	3
Es113x_SyncSendCode_TaskDVEModel_HWCF	18	812	0	0	3

All runtimes in microseconds.

The columns have the following meaning:

- Id: the internal ID of the process
- Calls: number of calls (while the trace was active)
- RTmin, RTavg, RTmax: minimum, average and maximum runtime of the relevant process.

3. You can also download all displayed files of a specific type as a ZIP archive. Click the relevant link in the "Download all ..." line.

4.4 CAN Trace

The CAN trace functionality enables CAN traffic to and from the IXXAT CAN cards to be monitored – the content of the CAN messages (send and receive) is transferred to a host PC as a UDP/IP stream.

For this purpose, you need a UDP monitor on your host PC, such as Netcat for Windows. The CAN trace is then started in the web interface (see “To start CAN trace in the web interface” on page 30).

Another possibility is to use the perl script provided, `labcar-rtpc-can-trace.pl` (select **CAN Trace** and click **Help**), with which the CAN trace is activated and the data output in a window of the command prompt.

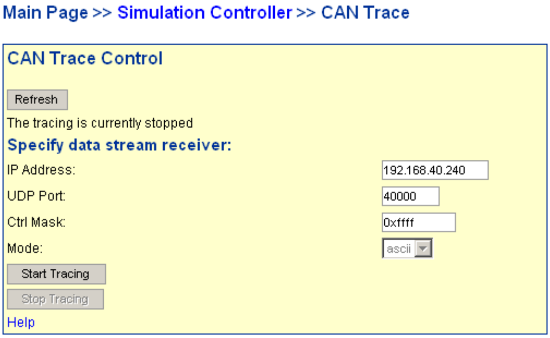
To start CAN trace in the web interface

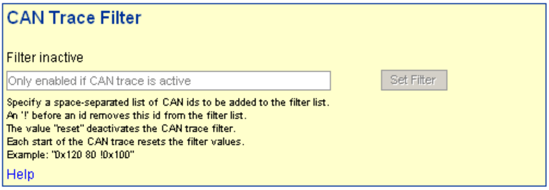
1. Launch the simulation controller in the web interface of ETAS RTPC.
2. Download your project (in the Experiment Environment).
3. Start the simulation.
4. Toggle to the ETAS RTPC web interface.

   When the simulation controller is running, the **Main Page >> Simulation Controller** page is displayed.
5. Select **CAN Trace**.
6. Enter the following data:
   - IP Address
     The IP address to which the stream is to be sent.
   - UDP Port
     The UDP port to which the stream is to be sent (see “**port=n**” on page 32).
   - Ctrl Mask
     The bit mask for selecting the board and the relevant controller (see “**ctrlmask=n**” on page 32).

7. Stream output mode

   Currently, only "ascii" is possible.





8. Click **Start Tracing** to send the HTTP request created with these arguments

```
http://192.168.40.14/cgi-bin/can-trace?
cmd=start&ipaddr=192.168.40.240&port=4000&ctrl-
mask=0xffff
```

   to the real-time PC.

9. If you would like to filter specific CAN-IDs, you can do so under "CAN-Trace Filter".

10. To end, click **Stop Tracing**.

The Arguments of the CGI Script

This section describes the arguments possible for the "can-trace" CGI script for running an HTTP request.

- cmd=[start|stop|status]

  The command to be run – this argument is mandatory.

  - cmd=start

    Starts the CAN trace – other arguments must be specified.
  - cmd=stop

    Stops the CAN trace – other arguments are not necessary.
  - cmd=status

    Outputs the current status of the CAN trace as a text.

- format=[html|text]

  The output format – "HTML" (default) or "Text" are possible. Specifying this argument is optional.

- ipaddr=a.b.c.d

  The IP address (in dotted-decimal notation) to which the UDP messages are to be sent. If no value is specified, the IP address of the caller is used.

- port=n

  The UDP port to which the UDP messages are to be sent. This port must be queried from the UDP monitor (= the counterpart). A port must be specified.

- ctrlmask=n

  The bit mask for selecting boards (and the controller on them), whose traffic is to be transferred. A unique bit is assigned to every controller on each CAN board; calculation takes place according to the formula:

  ```
  1 << (2 * board number + controller number)
  ```

  where "board number" can have values between 0 and n,
  "controller number" 0 or 1.

  The following table once more illustrates the assignment between board and controller and the relevant bits.

Bit	..	6	5	4	3	2	1	0
Board	..	3	2	2	1	1	0	0
Controller	..	0	1	0	1	0	1	0

  The width of the register is 32 bits; 8 CAN boards (= 16 controllers) are currently supported.

  The value for "ctrlmask" must be specified as a decimal or hexadecimal number (starting with "0x"). If no value is specified, the traffic on every controller is recorded.

  Example: If the first and second controllers of the third board are to be monitored, bits 4 and 5 of the register are set - the value of the bit mask is then $2^4 + 2^5 = 16 + 32 = 48$ (0x00030).

Description of the Answer

The CAN messages are transferred as pure ASCII code in UDP frames; each individual message is terminated by the character for a line feed \n (ASCII: 0x0A).

A typical output thus has the following format:

```
....
4101 1373.0778 1355.6203  E:0 CAN:0,0 RX ID: 7d DLC:8 DATA: 40 a3
02 03 00 e0 00 00
4102 1373.0789    -.----  E:0 CAN:0,1 TX ID: 7b DLC:8 DATA: 70 96
04 08 04 f0 ff 08
....
```

The meaning of the elements of an answer line is as follows:

- Line number

  Using the consecutive line numbering, the completeness of the recording (= no streaming errors, see "E:n" below) can be verified.

- Two time stamps

  The first value in the line specifies the time at which the CAN message was registered by the simulation model (on the real-time PC) in seconds.

  The second value is a hardware time stamp of the CAN board and indicates when a receive message was received. Naturally, this value does not occur with send messages.
- E:n

  "E:0" means: no error.

  "E:1" shows a streaming error resulting from too high a load on the CAN-bus which can lead to partial data loss.
- CAN:x,y

  Contains the number of the CAN board x (0...n) and the number of the controller y on it (0,1)
- RX/TX

  Describes the direction of a message from the point of view of the real-time PC (TX = send message, RX = receive message)
- ID: nn

  The hex identifier of the CAN message
- DLC: n

  The data length code of the CAN message
- DATA: XX XX XX XX XX XX XX XX

  The 8 data bytes of the CAN message (in hex notation)

4.5 Step-by-Step Execution of the Simulation

The "Model Step" function enables you to pause the automatic simulation and then run the simulation as individual steps (in the form of tasks and processes).

To do this, proceed as follows:

1. Start the simulation.
2. In the web interface, select the **Model Step** link.

   The "Model Step" page is displayed.

To pause automatic simulation

1. To pause the simulation, click the **Pause** icon.

   Automatic simulation execution stops.

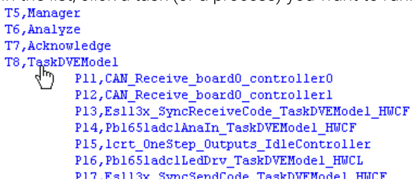
To select the acquisition task for the display in the Experiment Environment

1. If you follow the step-by-step execution of the simulation in one or more instruments, select the task that is also used for updating the instruments under "Acquisition Task". "Acquisition" is selected here by default.

   Regardless of the model step executed manually, the Acquisition Task is always run after it and the display instruments in the Experiment Environment thus updated.

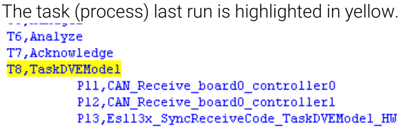
To run an individual step (task or process)

1. In the list, click a task (or a process) you want to run.

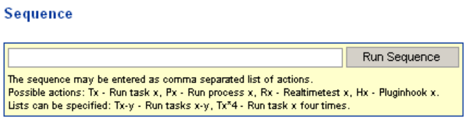
   The task (process) is run.

   The task (process) last run is highlighted in yellow.

To run a sequence of tasks or processes

1. You can run a sequence of tasks or processes you enter in the "Sequence" box.

   Individual components of the sequence are separated by commas.

   Components of a sequence are:

   - Tx: run task x
   - Px: run process x
   - Tx-y: run task x to task y
   - Tx*4: run task x four times in succession.
2. To run the sequence, click **Run Sequence**.

   The sequence is run.

To continue automatic simulation

1. To continue the automatic running of the simulation, click the **Continue** icon.

   Simulation is continued.

4.6 Real-Time Plugins (RT Plugins)

RT plugins can be used to load code dynamically at runtime. This allows the user to extend real-time models dynamically.

This section contains information on:

- “Preparation of the LABCAR-OPERATOR or the COSYM Project” on page 36
- “The Plugin Files” on page 37
- “Special Preprocessor Directives” on page 38
- “Templates and Examples” on page 39
- “Procedure” on page 39

4.6.1 Preparation of the LABCAR-OPERATOR or the COSYM Project

The project has to be prepared to work with real-time plugins. "Hooks" have to be placed at suitable positions within the process list. These hooks define the calling context used for executing the dynamically loaded code. The hooks are typically defined within a C-Code model.

Hooks API

All functions are defined in the C file header `rtos_realtime_test.h`, so please include this header with

```
#include "rtos_realtime_test.h"
```

Types:

`rtos_handle_t`

A generic handle type that is used to address the hooks.

Example:

```
static rtos_handle_t rttest_hook;
```

Init Code

```
rtos_handle_t realtime_test_hook_create(
                             const char *hookname);
```

This function creates a named hook and returns a handle.

Example:

```
rttest_hook = realtime_test_hook_create("SGPlugin");
```

Exit Code

```
int realtime_test_hook_delete(
                          const rtos_handle_t hook);
```

This function deletes a previously created hook.

Example:

```
realtime_test_hook_delete(rttest_hook);
```

Real-time Code

```
void realtime_test_hook_execute(
                                const rtos_handle_t hook);
```

This function calls the plugin functions that are attached to the hook. It defines the calling context and the exact calling position. The call to this function should be placed within a suitable periodic task.

Example:

```
realtime_test_hook_execute(rttest_hook);
```

4.6.2 The Plugin Files

The plugins to be loaded have to be defined as C-Code and header files on the user PC. All files have to be transferred to ETAS RTPC using a suitable tool (see "Preparing to Download Plugins" on page 39).

Basic Plugin Initialization

Each plugin must implement the following four basic plugin functions

```
int on_load(void);
int on_initialize(void);
void on_terminate(void);
void on_unload(void);
```

The functions `on_load` and `on_initialize` are called while the plugin is loaded. Both functions have to return 0 when successful. The functions `on_terminate` and `on_unload` are called when the plugin is unloaded.

The function `on_initialize` is typically used to call API functions that attach plugin functions to existing plugin hooks.

For this, a static interface object of type "I_realtime_test" has to be implemented. This object holds pointers to the callback functions that are called in a real-time context.

The type "I_realtime_test" is defined as follows:

```
typedef struct I_realtime_test_struct {
  int  (*init)(realtime_test_obj *obj);
  void (*exit)(realtime_test_obj *obj);
  void (*execute)(realtime_test_obj *obj, rtos_time_t
  t_ns);
  void (*background)(realtime_test_obj *obj);
}
I_realtime_test;
```

`init` is called at the initialization time of the real-time test object; `exit` is called when the real-time test object is to be deleted.

`execute` is called from the hook in a real-time context, and `background` is called periodically (approx. every 0.5 seconds).

Creating the Real-Time Test Object

The interface object (type `I_realtime_test`) is used to create a real-time test object. For this, the pointer to the interface object as well as an (optional) name and an (optional) user pointer might be used.

The function

```
realtime_test_obj *realtime_test_create(
const char *name, const I_realtime_test *interface,
void *data, const size_t resultstring_size);
```

is used to create a real-time test object.

Attaching a Real-Time Test Object to a Hook

The real-time test object can be attached to an existing hook. This is done by calling

```
int realtime_test_hook_add(
     const char *hookname, realtime_test_obj *obj);
```

Detaching a Real-Time Test Object from a Hook

To remove a real-time test object from a hook, use the

```
int realtime_test_hook_remove(
     const char *hookname, realtime_test_obj *obj);
```

function.

Deleting a Real-Time Test Object

The function

```
int realtime_test_hook_delete(
                    const rtos_handle_t hook);
```

deletes an existing real-time object.

4.6.3 Special Preprocessor Directives

A number of additional preprocessor directives are available with the plugin files. They are intended to ease the interaction with the currently running real-time executable.

The following directives can be used:

#label

- `#label name path`

  Generates a define "name" that represents the access to the element specified by "path".

#labelstruct

- `#labelstruct name path`

  Generates a define "name" that represents the base type (the final .xxx omitted) of the element specified by "path".

#labeltype

- `#labeltype name path`

  Generates a define "name" that represents the (scalar) C base type of the element specified by "path".

#al2addr

- `#a2laddr name path`

  Generates a define "name" that holds the A2L pseudo address as a string. This string can be used directly with the function `rtpc_a2l_address_value_ptr()` that is defined in the header file `a2l_address_resolution.h`.

4.6.4 Templates and Examples

There is an example of a C-Code file for implementing hooks (`PluginHook.c`) on your real-time PC under `http://192.168.40.14/tools/rtplugin/templates/`.

You will also find a template for a plugin there (`plugin-template.c`).

4.6.5 Procedure

Preparing to Download Plugins

A tool you can use to download the plugins to the real-time PC is available in the following directory:

`http://192.168.40.14/tools/rtplugin/RTPCRuntimeAccess/`

It contains two DLLs and two executables:

- `EE.RealTimePlugin.Core.Interfaces.dll`
- `EE.RealTimePlugin.Core.dll`
- `RTPC-Plugin-GUI.exe` (version with GUI)
- `RTPC-Plugin.exe` (instruction line version)

To be able to use this tool, you must copy both DLLs and one of the executables to a directory on your user PC.

To integrate hooks in a project (LABCAR-IP)

The file named in “Templates and Examples” on page 39 (`PluginHook.c`)enables you to generate a hook using a template.

1. Generate a new C-Code model in LABCAR-IP.
2. Add the relevant code.
3. Add a process.
4. Generate the code (**Project** → **Build**).
5. Toggle to the "OS Configuration" tab and assign the new process to the desired task.

To integrate hooks in a COSYM project

The file named in “Templates and Examples” on page 39 (`PluginHook.c`)enables you to generate a hook using a template.

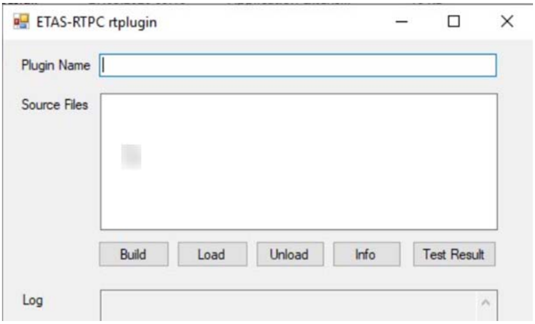
1. Generate a new C-Code model in COSYM.
2. Add the relevant code.

3. Add a process.
4. Open a system.
5. Add the model to the system.
6. Toggle to the "Deployment" view in the system and assign the new process to the desired task.
7. Generate the code (**Project** → **Build**).

To run the plugin (Experiment Environment)

1. Open your project in the Experiment Environment and run it.
2. Start the `RTPC-Plugin-GUI.exe` file named in “Preparing to Download Plugins” on page 39.

   The following GUI opens.



3. Under "Plugin Name" enter a name for the subsequent list of source files.
4. Enter the source files you want to belong to this plugin under "Source Files".
5. Alternatively you can drag files to this field using Drag & Drop.
6. Select **Build** to load the files to the real-time PC and to generate code there.
7. Select **Load** to load the plugin into the running experiment.
8. Select **Unload** to stop running the plugin.

4.7 CPU Core Load

The distribution of the process load to the processor cores can be issued in an XML file using

`http://192.168.40.14/cgi-bin/cpu-load[?detail=n]`

Depending on the degree of detail, the XML file contains different information:

- `detail=0` (default)

  This only issues the CPU load (in %):

```
<?xml version="1.0" encoding="UTF-8" ?>
- <load>
    <cpuload cpu="0" load="73.9" loadunit="%" />
  </load>
```

- `detail=1`

  The total load and the distribution to the individual simulation tasks are issued.

- `detail=2`

  In addition, all real-time processes are issued.

- `detail=3`

  In addition, all non-real-time processes are issued.

- `detail=4`

  In addition, all commands are issued.

4.8 Benchmark Indicator

To compare the simulation performance of ETAS RTPC on different PCs, a standardized calculating routine can be used.

Include the header file (in `/opt/etas/include`)

```
rtpc-benchmark.h
```

in any C-Code model and run one of the following routines:

```
double rtpc_benchmark_standard1(double salt);
double rtpc_benchmark_standard2(double salt);
```

The first routine calculates 2000 variables with 8000 instructions, the second 50 variables with 400 instructions – arguments and return values have no significance.

4.9 Testing Hardware Performance with "cyclictest"

The "cyclictest" application is used to test how well specific PC hardware is suited for real-time applications. This involves the measuring of the interrupt latency between the activation of the timer and the running of the relevant code.

To launch Cyclictest

1. In the web interface, select **Main Page >> System Info** and scroll to **Cycle Test - Test Program**.

2. Click **Cyclictest Control / Results**.

   The program can be launched on the following page. Results of tests previously executed are shown below this.

3. Select "Duration" and click **Start Cyclictest**.

Example:

A timer is triggered periodically every 100 ms and then a test is carried out to see when the relevant Software routine is actually run: The timestamp from the start of the routine is compared to the timestamp when the timer was triggered.

Results of executed Cyclictests on this RTPC

0: Wed Sep 30 06:19:19 2015

```
# /dev/cpu_dma_latency set to 0us
T: 0 ( 7098) P:99 I:100 C:  99982 Min:      1 Act:    1 Avg:    1 Max:      26
T: 1 ( 7099) P:99 I:110 C:  90910 Min:      1 Act:    1 Avg:    1 Max:       4
T: 2 ( 7100) P:99 I:120 C:  83334 Min:      1 Act:    1 Avg:    1 Max:       5
T: 3 ( 7101) P:99 I:130 C:  76924 Min:      1 Act:    1 Avg:    1 Max:       6
T: 4 ( 7102) P:99 I:140 C:  71413 Min:      1 Act:    2 Avg:    1 Max:       5
T: 5 ( 7103) P:99 I:150 C:  66633 Min:      1 Act:    2 Avg:    1 Max:       3
T: 6 ( 7104) P:99 I:160 C:  62468 Min:      1 Act:    2 Avg:    1 Max:       2
T: 7 ( 7105) P:99 I:170 C:  58794 Min:      1 Act:    2 Avg:    1 Max:       4
```

1: Wed Sep 30 05:56:48 2015

```
# /dev/cpu_dma_latency set to 0us
T: 0 ( 6106) P:99 I:100 C: 599981 Min:      1 Act:    1 Avg:    1 Max:      34
T: 1 ( 6107) P:99 I:110 C: 545455 Min:      1 Act:    1 Avg:    1 Max:       6
T: 2 ( 6108) P:99 I:120 C: 499993 Min:      1 Act:    2 Avg:    1 Max:       6
T: 3 ( 6109) P:99 I:130 C: 461524 Min:      1 Act:    1 Avg:    1 Max:       3
T: 4 ( 6110) P:99 I:140 C: 428553 Min:      1 Act:    2 Avg:    1 Max:      55
T: 5 ( 6111) P:99 I:150 C: 399972 Min:      1 Act:    2 Avg:    1 Max:       3
T: 6 ( 6112) P:99 I:160 C: 374966 Min:      1 Act:    1 Avg:    1 Max:       3
T: 7 ( 6113) P:99 I:170 C: 352910 Min:      1 Act:    2 Avg:    1 Max:       3
```

The meaning of the data issued is as follows:

- T

  Cyclictest starts a periodical task T*n* per core *n*, i.e. task T0 runs on core 0, task T1 on core 1 etc. The number in brackets represents the process ID of the relevant task.

- P

  The priority of the task (here: fixed 99)

- I

  The invoke interval of the task in µs (here: starting with 100 µs, becoming larger in increments of 10 µs)

- C

  The number of task calls during the test run

- Min, Act, Avg, Max

  The values are the latencies to the timer trigger point in µs. "Act" is the current value - statistics are also created (Min/Avg/Max).

  A value of 15 for "Max" means that it took 15 µs for the software to react to a timer event. When "Max" is 45 µs it is thus hardly possible to use this PC to calculate a 50 µs task.

  The "Max" values are shown in color (with increased latencies):

  – Max <= 30 µs: no color
  – 30 µs < Max <= 50 µs: yellow
  – Max > 50 µs: red

4.10 CAN/CANFD-Abstraction and LIN-Abstraction

4.10.1 General Functional Principle

If a project is used on LABCAR systems with differently developed CAN/CANFD- and LIN interfaces without the CAN/CANFD and LIN configuration having to be changed, the feature of the CAN/CANFD and LIN abstraction (General bus abstraction) can be used. This feature is used in conjunction with the RTPC with the CAN and LIN editors or the network model.

The assignment of combinations from board-ID and controller-ID to channels is essential for the bus abstraction. This mapping is shown on the RTPC in the form of ini-files. These files are divided into sections. The sections contain pairs of a variable name and a value, which are connected by the equal sign.

At the bus abstraction the section describes the bus type as well as the number of the canal. Valid identifiers for bus types are `can`, `canfd` and `lin.` Channels are defined by any natural numbers. The order of the sections in an ini-file is arbitrary. Combinations of board-ID and controller-ID are formed by the variables `boardid` and `controllerid` within the associated section.The values of these variables are natural numbers.

Examples:

```
[can0]
boardid=0
controllerid=1

[canfd12]
boardid=1
controllerid=0

[lin4]
boardid=0
controllerid=0
```

The first example describes the assignment of the second CAN controller, of the first board to channel 0. The board with the ID 1 in the second example is CANFD capable. For this reason, the first controller is assigned on the CANFD channel 12. The first LIN controller of the first board is referred to as LIN channel 4.

NOTE

The numbering of the LIN controllers starts with "0" for each board regardless of how many CAN controllers are present on the board.

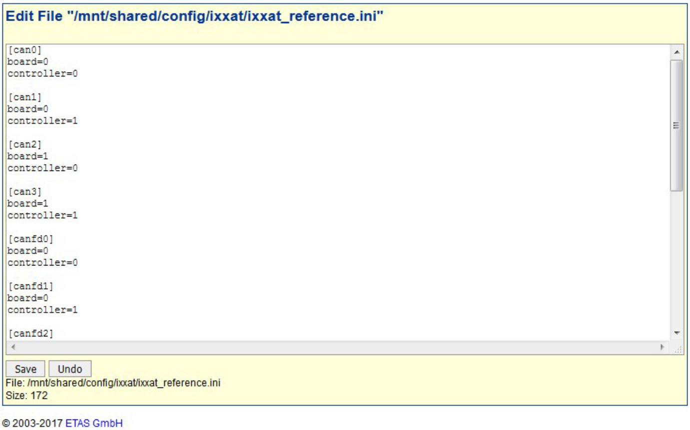
The bus abstraction is exclusively performed on the RTPC. The code generated by the CAN and LIN editors or the network model is modified before the build process. The combinations of board ID and controller ID that are configured in the editors or the network model are assigned to the corresponding channels. Then the built project is neutral with respect to the CAN/CANFD and LIN interfaces as it internally uses the channels. At runtime, concrete interfaces are addressed via the appropriate combination of board ID and controller ID from the information about the channels.

4.10.2 Abstraction of the Bus Interfaces During the Build Process

During the build process, an existing configuration of CAN/CANFD and LIN interfaces is converted into a general configuration. This general configuration can then be used on different LABCAR systems without the project having to be adjusted. The transformation is based on a reference configuration that must be present on the RTPC on which the project is being built. The reference configuration can be edited using the web interface. All CAN/CANFD and LIN interfaces that are listed in the reference configuration will be replaced by the respective channels when building the project.

Missing combinations of board ID and controller ID are ignored and not replaced by channels.

The editor for the reference configuration is available in the web interface of the RTPC as follows **Configuration** → **CAN/LIN Abstraction Configuration** → **Expert Edit.**



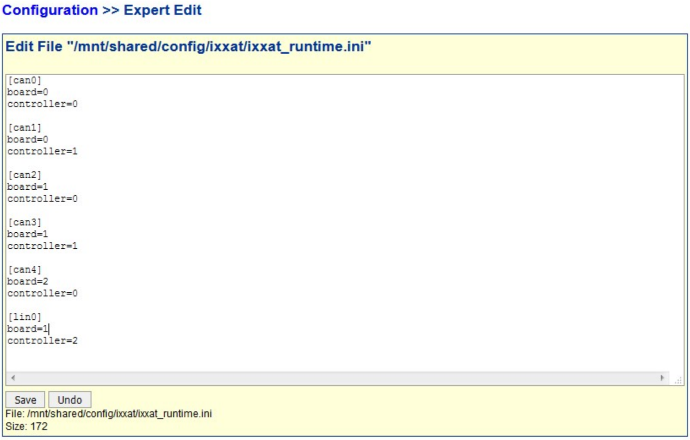
The bus interfaces are still described with the board ID and the controller ID in the CAN and LIN editors or the network model. The conversion into an abstract description with channels takes place during the building of the project on the basis of the reference configuration.

4.10.3 Converting the Channel Description into Concrete Bus Interfaces at Runtime

After downloading the project to the simulation RTPC, the back transformation of the abstract CAN/CANFD and LIN channels into the corresponding combinations of board ID and controller ID takes place. The runtime configuration is used for this. The runtime configuration can be edited using the web interface.

Combinations of controller ID and board ID that were not replaced by channels when the project was built remain in place and are addressed directly at runtime.

The editor for the runtime configuration is available in the web interface of the RTPC as follows **Configuration** → **CAN/LIN Abstraction Configuration** → **Expert Edit**



4.10.4 Template for a Channel Configuration

Each time the RTPC boots, a template for the configuration files is generated. This template does not overwrite the existing configuration files. It can be used as an overview of the installed CAN/CANFD and LIN hardware.

The template of the configuration file can be downloaded in the web interface as follows **Configuration** → **CAN/LIN Abstraction Configuration** → **Download template**

An example for a configuration:

```
; 00000000000000: iPC-I XC16/PCI, 2 CAN, 0 LIN
; HW473609: CAN-IB200/PCIe, 2 CAN, 0 LIN
; HW471605: CAN-IB600/PCIe, 2 CAN/CAN-FD, 0 LIN

[can0]
board=0
controller=0

[can1]
board=0
controller=1

[can2]
board=1
controller=0

[can3]
board=1
controller=1

[can4]
board=2
controller=0

[can5]
board=2
controller=1

[canfd0]
board=2
controller=0

[canfd1]
board=2
controller=1
```

The example shows a configuration with one XC16, one IB200 and one IB600 card each.The variable `RTOS_IXXAT_CAN_DRIVER_ORDER` is set so that the XC16 card is detected before the IB200 card and IB600 card. The controllers of the IB600 card are CAN-FD-capable, but they can also be used as standard CAN controllers. For this reason, the corresponding combinations of board ID and controller ID are listed both as CAN channels (`can4` and `can5`) as well as CANFD channels (`canfd0` and `canfd1`).

5 Configuration

This chapter describes the uses and configuration of ETAS RTPC with the web interface.

5.1 Applications

There are three applications for ETAS RTPC:

A Model-in-the-Loop (MiL)

B Hardware-in-the-Loop (HiL)

C Hardware-in-the-Loop with Measuring and Calibration Access (M&C)

Depending on the application, the „RTPC_USAGE_ETH1“ parameter has to be set in a specific way (in the web interface under **Main Page** → **Configuration** (see “Configuration” on page 54).

5.1.1 Model-in-the-Loop (MiL)

In MiL tests, the Unit-under-Test consists of a model of a controller functionality written in software. No I/O hardware is necessary.

The following setting has to be made:

Parameter	Setting
RTPC_USAGE_ETH1	Off
...	...
RTPC_USAGE_ETHn	Off

Tab. 5-1 Setting for MiL Use

5.1.2 Hardware-in-the-Loop (HiL)

HiL tests are the standard application for ETAS RTPC. In HiL operation, the Unit-under-Test exists as hardware. It is connected to the test system via I/O boards.

The real-time PC addresses the I/O hardware via an Ethernet connection. Normally, the Eth1 (HW) port of the real-time PC is used.

The following setting has to be made:

Parameter	Setting
RTPC_USAGE_ETH1	ES1130

Tab. 5-2 Setting for HiL Use

5.1.3 Hardware-in-the-Loop with Measuring and Calibration Access (M&C)

With "HiL with measuring and calibration access", it is possible to access the UuT simultaneously during test operation using serial protocols (e.g. ETK or XCP).

Measuring and calibration access can be realized using a compact ES590 ECU interface module. The ES590 and RTPC are connected directly to the user PC via an ES600 network module.

The following setting has to be made:

Parameter	Setting
RTPC_USAGE_ETH1	ES1130

Tab. 5-3 Settings for HiL Use with Measuring and Calibration

5.1.4 Hardware-in-the-Loop with Load and Error Simulation

If an ES4408 system is used to simulate loads or an ES4440 Compact Failure Simulation Module to simulate errors, these modules communicate with the real-time PC via Realtime UDP (rtudp).

The following settings then have to be made for the relevant Ethernet ports on the real-time PC:

Parameter	Setting
RTPC_USAGE_ETH*n*	rtudp_*x* (x = 0...4)

Tab. 5-4 Settings for Systems for Load or Error Simulation

5.1.5 Massive I/O

Massive I/O means that the HiL system can be assigned a large number of I/O channels. With ETAS RTPC V6.5.2, this means that up to eight VME/VXI I/O systems (ES4100, ES4105, ES4300) are supported.

Each system is controlled by a System Controller Board. To ensure real-time data transfer, there is a a point-to-point connection between the real-time PC and each I/O system. The DVE model thus has access to all I/O boards regardless of the system the individual board is contained in.

The following settings have to be made:

Parameter	Setting
RTPC_USAGE_ETHn	ES1130, ES1130_1, ES1130_2 or ES1130_3

Tab. 5-5 Settings when Using Several I/O Systems (Massive I/O)

5.2 The Web Interface

You can do the following via the ETAS RTPC web interface:

- configure ETAS RTPC
- install software updates
- display status messages and system information

The access to the web interface, navigation within the interface and the functions are described in this section.

5.2.1 Access to the Web Interface

You can access the web interface of ETAS RTPC V6.5.2 via the URL:

`http://192.168.40.14`

Simply launch a web browser on the user PC and navigate to this address.

Instead of the IP address `192.168.40.14`, you can also use the alias "rtpc" if you declare it in the Windows system file `hosts`. This file is in the following directory:'

`C:\Windows\system32\drivers\etc\hosts`

Add the following line to the existing entries:

```
192.168.40.14    rtpc
```

This makes it possible to reach the web interface of ETAS RTPC via the URL `http://rtpc`.

5.2.2 Navigation

The navigation line directly under the blue headline is an orientation guide in the web interface (see Fig. 5-1).

Fig. 5-1 The Web Interface with Running Simulation Controller

You can also return (when the simulation controller is stopped) to higher-order pages, e.g. the "Main Page", quickly using the navigation line.

Not all functions of the web interface can be invoked in the operating modes "Simulation Controller" and "Ethernet Bridge". Once the web interface has been started, you are automatically led to the relevant subpages.

The following functions are available when the simulation controller is running (see Fig. 5-1 on page 52):

- **System Info**

  This is where you obtain information on the system and find log files. For more details, refer to the section “System Info” on page 61.

- **Runtime Trace**

  This is where you can start runtime traces in which processes and tasks are logged on the real-time PC. For more details, refer to the section “Runtime Traces” on page 28.

- **CAN Trace**

  The CAN Trace functionality enables the monitoring of CAN traffic to and from the IXXAT CAN cards. This functionality is described in the section “Runtime Traces” on page 28.

- **Model Step**

  This function enables step-by-step execution of the simulation. For more details, refer to the section “Step-by-Step Execution of the Simulation” on page 34

- **Power Control**

  This is where you can shut down or reboot the system (see “Power Control” on page 54).

5.2.3 Functions

The following figure shows an overview of the ETAS RTPC web interface structure and how the different web pages can be accessed in the relevant operating modes.

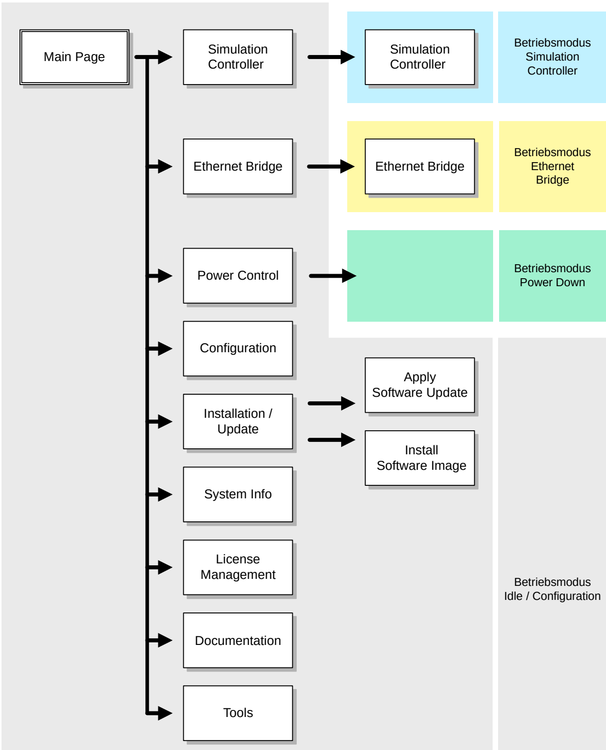


Fig. 5-2 Structure of the Web Interface

The individual pages are described in detail below.

5.2.3.1 Main Page

All functions of the ETAS RTPC web interface can be selected via the Main Page.

5.2.3.2 Simulation Controller, Ethernet Bridge

These menu items are used to start the relevant operating modes "Simulation Controller" and "Ethernet Bridge". The operating modes have already been described in the section “Operating Modes” on page 25.

Once an operating mode has been activated, only the relevant subpages can be accessed.

5.2.3.3 Power Control

If the on/off switch on the real-time PC is not accessible or does not exist, ETAS RTPC can be shut down or rebooted using this web interface link.

To boot ETAS RTPC via the network ("Wake-On-LAN"), you require a corresponding tool on the relevant network PC to send the "magic packets" (e.g. the free tool "WOL - Magic Packet Sender 2007"). Data on how to configure this tool can be found under "Wake On LAN Settings".

5.2.3.4 Configuration

This is the configuration dialog of ETAS RTPC which consists of four parts:

- Configuration Support
  - Automatic ES1130 Configuration

    Automatically configures the Ethernet connection(s) (Eth1...Ethx). The real-time Ethernet connection is checked and – if an ES1130 is connected – the "RTPC_USAGE_ETH1“ parameter is set to the value "es1130".

    NOTE

    The "RTPC_USAGE_ETHn" parameter is only displayed if the relevant Ethernet interface has also been configured.

    Refer to “To run the configuration” on page 18 for a description of autoconfiguration.
  - ES1130 Detection

    Checks whether the connection to the ES1130 works.
- RTPC Configuration

  The "RTPC Configuration" section is used for setting the configuration mode and activating the debug options.

  In section „Host Ethernet Configuration (ETH0)“ the user PC (connected to Ethernet adapter ETH0) ist configured.
  - IP Address

    This is where you can specify the IP address (IPv4) of the Ethernet adapter used to connect to the host PC (Default: 192.168.40.14). We seriously recommend that you use an IP address for a private Class-C network
    (Address: 192.168.x.y, netmask: 255.255.255.0).

- Netmask

  The netmask for ETH0
- DHCP

  This is where you can specify whether the IP address of ETH0 should be obtained via DHCP or not.
- Ethernet Negotiate

  This is where you can specify whether the controllers involved can independently negotiate and configure the maximum possible transfer speed and the duplex procedure with each other ("auto") or manually select certain settings.

In section "Realtime Ethernet Configuration" additional realtime Ethernet connections are configured.

> **NOTE**
>
> To identify the card assigned to the Ethernet interface "Eth_n", use the mouse to click *Blink*.

- Usage

  "ETH1", "ETH2" etc. can be used to connect an ES1130. If an ES1130 Simulation Controller Board is connected, select "es1130".

  The settings "es1130_*n*" are intended for additional ES1130 Simulation Controller Boards that can be connected; the setting "rtudp_n" is for real-time communication with appropriate ETAS hardware (such as the ES4440 Compact Failure Simulation Module or the ES4408 Load Chassis). Details on the real-time UDP (rtudp) driver API can be found on the web interface at http://192.168.40.14/api/rtudp.html.

  Individual Ethernet connections can be configured in Bridge mode here (see section 4.1.3 on page 26).
- IP Address

  This is where you can specify the IP addresses (IPv4) of additional Ethernet adapters (ETH1, ETH2, ...) which connect the Real-Time PC with other devices (e.g. 192.168.100.23).

  We seriously recommend that you use an IP address for a private Class-C network

  (Address: 192.168.x.y, netmask: 255.255.255.0).

– Ethernet Negotiate

  This is where you can specify whether the controllers involved can independently negotiate and configure the maximum possible transfer speed and the duplex procedure with each other ("auto") or manually select certain settings.

  NOTE

  To change the designation "ETHn" of a board, activate the option "Show options to control...". Then you can assign (column "Ethernet Order") a certain board (using the boards MAC address).

  NOTE

  If the option "auto" has been selected for "Ethernet Order", a new installation can result in a change of the "MAC Address:ETH*n*" assignment. To freeze the current assignment, select *Freeze current Ethernet Order.*

Section "General Parameters":

– RTPC_POWER_UP_MODE

  This parameter specifies the operating mode for ETAS RTPC after start.
  The following values can be set:
  idle: ETAS RTPC starts in "idle" mode. This mode is used for configuration and system update.
  simulate: ETAS RTPC starts in simulation mode. ETAS RTPC cannot be configured in this mode.
  bridge: ETAS RTPC starts in "Ethernet Bridge" mode, i.e. the Ethernet interfaces of ETAS RTPC are connected logically. ETAS RTPC cannot be configured in this mode.

– RTPC_LOG_LEVEL

  Events are logged and written to a file during ETAS RTPC operation. The level for the events to be logged can be set here – "debug" logs all events, "emerg" only logs those that make the system unusable. A useful setting for normal operation is "warning" or "error".

– RTPC_NAME

  This is where the particular real-time PC can be assigned a name. This name is helpful if the user is working with several ETAS RTPC systems - it is shown both on all web interface pages and in the browser title.

– RTPC_TIMEZONE

  Defines the time zone

– RTPC_PARALLEL_MAKE

  Enables the parallel execution of the GNU make process (i.e. simultaneously on several CPU cores).

  The feature is disabled with "no" and GNU make is only called up for one job. The feature is enabled with "auto" providing the real-time PC permits. When "force" is set, this feature is always used.

– RTPC_TASK_MONITOR_0/1

The task monitor makes it possible to monitor two tasks during run-time. The task to be monitored can be selected here by its number. While the task is active, the relevant bit (0 or1) at the LPT interface is set to "High". This can be monitored using an oscilloscope. Note that using the task monitor causes a processor load on the ETAS RTPC. To deactivate the task monitor, set the values to "-1".
The following table lists the assignment of the relevant pins of the 25-pin DSUB connector, LPT.

Pin	Function
2	Data bit 0
3	Data bit 1
18 - 25	Ground

– RTPC_TASK_TIMING_STATISTIC

This parameter defines the recording frequency (in seconds) for the statistics of the task runtimes. "0" deactivates recording.
The statistics contain the following information:
- minimum/maximum time dt (in ns) between two task activations
- minimum/average/maximum runtime (in ns) of the task

NOTE

The statistics are not output in real-time mode to prevent any disturbances of real-time behavior. The measuring functions are not implemented atomically. Individual events may therefore go missing.

NOTE

The recorded events depend on the value of the "RTPC_LOG_LEVEL" parameter (see Page 56).

– RTPC_INVALID_CPU_CORE_HANDLING

This parameter determines how to handle an invalid specification of CPU cores (in the "OS Configuration" tab in LABCAR-IP). Possible values are "warn" (default) and "stop".
"warn": If, in the OS Configuration, non-existent cores are specified, a warning is issued in the ETAS RTPC log file and all real-time tasks are run on a single CPU core.
"stop": In the case of an invalid specification of CPU cores, the real-time model does not start and an error message is issued in the ETAS RTPC log file.

- RTOS_TASK_STACKSIZE

  ETAS RTPC uses a separate stack for each running task. The size of this stack (in kB = 1024 Byte) can be defined using this parameter. A useful value is 1024 kB.

  NOTE

  If models with lots of tasks are used, the stack size may have to be reduced due to the limited size of the RAM.

- RTOS_TRACE_BUFFERSIZE

  ETAS RTPC has an automatic internal serial protocol which is useful when, for example, tests are carried out to assess the performance of simulation models.
  This parameter is used to determine the size of the buffer (in the unit "number of events") before the data is written to a protocol file (*.rtt). This file can be accessed using **Main Page** → **Simulation Controller** → **Trace File Access** and then displayed and analyzed with RTA-TRACE.

- RTPC_COMPILE_OPTIMIZATION

  ETAS RTPC uses the GNU C Compiler "gcc" – code generation optimization can be set in 4 steps (0 ... 3) which correspond to the gcc compiler options -O0 ... -O3. The larger the number, the better the optimization - this does, however, also mean that more time is required for compilation.
  If you are not sure about this setting, select "1" or "2".

- RTPC_COMPILE_LINK_DEBUG

  If you set this parameter to "yes", the compiled object files and the executable file are assigned debug information. This can be useful when detecting errors in the simulation model (see "**Debug/ Disassemble**" on page 61).

  NOTE

  Only set this option if it is absolutely necessary as the executable file becomes very large as a result.

- RTPC_COMPILE_SWAPSPACE_MB

  When using large Simulink models, the C files generated can be very large. This can lead to the computer not having enough disk space available.
  This parameter enables you to make additional disk space available for the compilation phase. This space is in the `compile.swapfile` file and is only used for the compilation of the model, not for the simulation.
  Possible values are 0, 256, 512, 768 and 1024 - these figures specify the size of the swap file on the hard disk in megabytes.
  If the value is set to 0, any swap file created previously is deleted. The value is set to 0 by default - if there are no problems during compilation, continue to use this setting.

The selected settings are saved using the **Save RTPC Configuration** button – **Discard Changes** discards changed settings.

- Clock Synchronization

  Click **Synchronize** to compare and synchronize real-time PCclock and host clock.
- System Configuration

  This part is used to configure the PC hardware. The options are automatically set correctly the first time ETAS RTPC is booted. Modification is therefore not necessary.

  If an Ethernet interface is installed at a later date, the "RTPC_SYS_ETH*n*" (n = 1 ... 4) option has to be changed from "None" to "present".

  Save your selection by clicking the **System Configuration** button.
- System Boot Settings

  ETAS RTPC saves two complete software versions on separate partitions of the real-time PC. This enables a risk-free return to a familiar software version.

  You can select which software version is to be used when the system is next booted in the selection box. This selection is saved using the **Set Boot Version** button.
- Reboot System

  Reboot System restarts ETAS RTPC with the selected software version.

5.2.3.5 Installation/Update

- Apply Update

  The current ETAS RTPC version can be updated with software updates via the web interface. These software updates can be obtained directly from ETAS.

  - Under "Upload Update File" click the **Browse...** button to select the file with the update (extension `*.tgz`).
  - Select **Upload** to transfer the file to the real-time PC.

    The uploaded file is now selected under "Select Update to Apply".
  - If several updates have been uploaded, select the relevant update to be activated.
  - Under "Select System to Update" select the software version to be updated. The software version currently active is shown ("Active").
  - Click the **Apply Update** button.

    Updates carried out to date are listed under "Applied Updates".

    Under "Delete Installed Update Files" you can delete files no longer required.
  - Activate the files to be deleted and click **Delete**.
- Install New Version

  The "Install New Version" function is used to execute a complete new installation. ETAS RTPC saves all software versions installed to date from CD on the hard disk of the real-time PC. These versions can be written back to any partition.

  There are two ways of making installation files available on the hard disk of the real-time PC:

- Upload Image file

  An image (on your local system or on a network resource) is transferred to the real-time PC.
- Check CD/DVD-ROM disk

  This is where the optical drive of the user PC is checked for the presence of a ETAS RTPC installation disk. If one is found, the image file is transferred to the real-time PC.

The images provided in this way can now be selected using "Select Software Image" (see below).

- Select the partition installation should take place in under "Install Partition".

  Every partition except the active one can be overwritten.
- Select which version is to be installed under "Select Software Image".
- Use the "Use the existing configuration...." option to specify whether the existing configuration of ETAS RTPC is to be used for this version or whether standard values are to be used.
- If you leave the "Skip implicit updates" option disabled, all available updates for the relevant software version are also automatically installed.
- Click **Install**.

The version to be booted is selected in "Configuration" (see “**System Boot Settings**” on page 59).

Images on the hard disk of the real-time PC which are no longer needed can be marked under "Delete Installed Software Images" and then deleted using **Delete**.

- Clone Installed Version

  You can clone versions of ETAS RTPC that have already been installed by generating an installable file (*.tgz) and downloading this file to your operating PC. For a detailed description of this procedure, refer to the section “Cloning Installed Versions” on page 19.

5.2.3.6 System Info

The "System Info" function shows various information on the status of ETAS RTPC. If an error occurs, this page can be saved and used for analysis purposes.

- Log File messages

  The messages recorded are shown in this window. It is also possible
  - to have the complete file displayed (**View complete log file**),
  - to generate a ZIP archive which contains the log file as well as other files with system information (**Generate ZIP file report**) or
  - to generate an encrypted ZIP archive (password: etas) (**Encrypted ZIP**).

  NOTE

  As the ETAS firewall only accepts encrypted ZIP files, you must select this option if you want to send a report archive to ETAS by e-mail.

- Realtime Process Report

  This provides detailed information on running real-time processes.
- System Information

  System information of the real-time PC is shown here.
- Cyclictest - Test Program

  This is where the test program "cyclictest" can be launched - the results are also shown on this page. For a detailed description, please refer to “Testing Hardware Performance with "cyclictest"” on page 41.
- CPU Performance Monitor

  Uses the "CPU Performance Monitor" registers to analyze CPU performance.
- I/O Hardware Information

  This provides detailed information on the connected I/O hardware (VME-bus and PCI/PCIe boards).
- Applied Updates

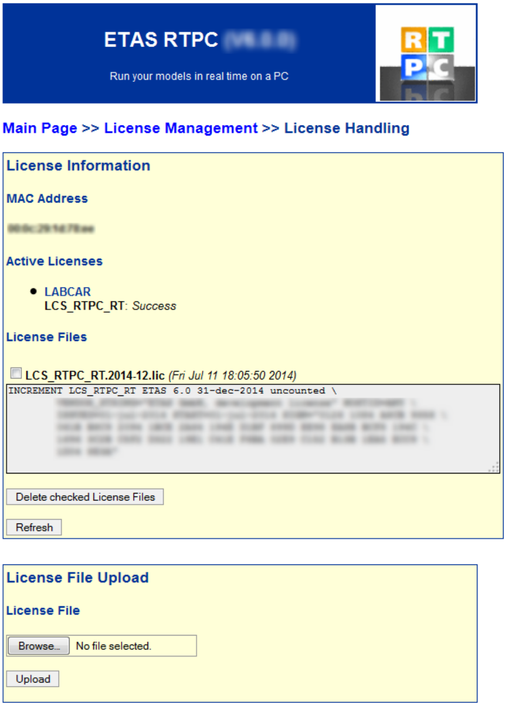
  Contains information on installed updates.
- Debug/Disassemble

  If the "RTPC_COMPILE_LINK_DEBUG" parameter has been set (see Page 58), the model can be examined here for errors.

5.2.3.7 License Management

This is where the licenses for ETAS RTPC and the FlexRay boards are managed.

- ETAS RTPC Runtime Licenses
  - License Information
    Shows the MAC address to which the license is linked, the names of the active licenses and the license file itself.
  - License File Upload
    This is where you can upload a license file on the host PC to the real-time PC (**Upload**).



- Elektrobit Device Licenses

  This is where the licenses for the FlexRay board from Elektrobit can be managed.

  The serial numbers of the connected are determined – then you can assign a license file to each board.

- To transfer a license file from the user PC to a EB_5X00 board, select the board serial number.
- Select the file with **Browse** and click **Write**.
- To transfer a license file from a board to the user PC, select the board serial number and click **Read**.

5.2.3.8 Documentation

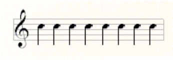
The "Documentation" link leads you to a directory containing the following documents:

- The manual on ETAS RTPC V6.5.2
- The current release notes
- A list of PCs approved by ETAS
- Two documents with information on the BIOS settings of the PCs from ETAS

6 Troubleshooting

This chapter presents the description of some typical problems with possible solutions.

A **When ETAS RTPC is started up, the sequence of error notes is played.**



- Cause:

  The Simulation Controller could not be started.
- Remedy:

  See “**Simulation Controller mode cannot be activated.**” on page 65.

B The ETAS RTPC web interface cannot be accessed.

- Cause:

  Ethernet connection does not exist, is defective or the settings are incorrect.
- Remedy 1:

  Execute the command `ping 192.168.40.14`. If you receive an answer from ETAS RTPC, check the settings of your Web browser (e.g. proxy settings, see below “Remedy 4:” on page 64).
- Remedy 2:

  If you do not receive an answer from ETAS RTPC, check the Ethernet cable. Make sure that you are using a crossover cable.
- Remedy 3:

  If the cable is OK, check the configuration of the Ethernet interface (IP address: 192.168.40.240, net mask: 255.255.255.0)
- Remedy 4:

  In your Internet configuration, a proxy server is configured for the LAN. To be able to access the ETAS RTPC web interface, the IP address of the real-time PC (192.168.40.14) must be included in the list of exceptions that should not use the proxy server.

  To do so, proceed as follows:

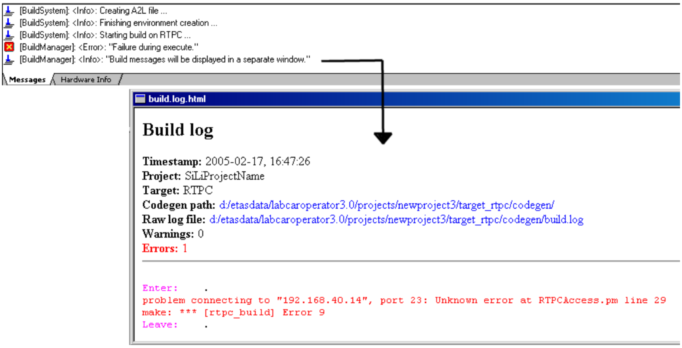
  1. In the Windows Start menu, select **Control Panel** → **Network and Internet** → **Internet Options**.

     The "Internet Properties" window opens.
  2. Select the "Connections" tab and click **LAN settings**.

     The "Local Area Network (LAN) Settings" window opens.
  3. Click **Advanced** in the "Proxy server" field.

     The "Proxy Settings" window opens.
  4. Enter the IP address of the real-time PC in the "Exceptions" field (192.168.40.14).
  5. Close one window after the other with **OK**.

C **Compiling models does not work.**



– Cause:

The real-time PC is not connected to the user PC or is switched off.

– Remedy:

Connect the real-time PC or switch it on. Check communication by calling the Web interface (http://192.168.40.14 or http://rtpc).

D **The project download ("Open Experiment") in LABCAR-OPERATOR does not work.**

– Cause 1:

Simulation Controller has not been started

– Remedy 1:

Use the Web interface to check whether the Simulation Controller has been started. Activate Simulation Controller mode if necessary.

– Cause 2:

ES1130 firmware is not compatible with ETAS RTPC.

– Remedy 2:

Check whether the firmware which is compatible with ETAS RTPC has been programmed on the ES1130. See the configuration page in the web interface for more details.

E Simulation Controller mode cannot be activated.

– Cause 1:

The ES1130 has not been connected or is switched off. (HiL mode).

– Remedy 1:

Check whether the ES1130 has been connected and is switched on. Call the "ES1130 Detection" function under "Configuration Support". The window below shows the result of the hardware scan. The following line should be visible:

```
Ethernet Controller [eth1]: Connected to Target ES1130
```

– Cause 2:

The settings for RTPC_USAGE_ETH1 are not correct

- Remedy 2:

  Check the settings for RTPC_USAGE_ETH1 in the configuration dialog and correct these (see “Applications” on page 49).
- Cause 3:

  There is an interrupt conflict in the Ethernet interface.
- Remedy 3:

  If an interrupt conflict has been registered on the configuration page of the web interface, insert the board into another slot or deactivate unnecessary peripheral devices in the PC BIOS (e.g. USB or sound support).

7 Glossary

This chapter explains technical terms used in the manual. The terms are listed in alphabetic order.

API

Application Programmer's Interface

BIOS

Basic Input Output System - the firmware of the PC

CAN

Controller Area Network (ISO 15765), a multi-master broadcast serial bus standard for connecting ECUs

CPU

Central Processing Unit

DSUB

A common hardware connector type

DVD

Digital Versatile Disc

DVE

Driver-Vehicle-Environment

EVE

ETAS Virtual ECU

Eth0

Designation of the Ethernet port for connecting the host PC

Eth1

Designation of the Ethernet port for connecting hardware like ES1130 or ES4440

GB

Gigabyte

GNU

Open source software community

HiL

Hardware-in-the-Loop

HTTP

Hypertext Transfer Protocol

ID

Identification key

IDE

Integrated Drive Electronics

IP

Internet Protocol

INCA

Measurement, Calibration, and Diagnosis tool available from ETAS

ISO

International Organization for Standardization

LAN

Local Area Network

LPT

Parallel port (IEEE-1284)

MB

Megabyte

NTP

Network Time Protocol

PC

Personal computer (desktop or laptop)

PCIe

PCI Express (Peripheral Component Interconnect Express)

PS/2

Serial input for PCs

RAM

Random Access Memory

RTPC

Real Time PC

UDP

User Datagram Protocol – a network protocol

URL

Uniform Resource Locator

VECU

Virtual electronic control unit

VME

A common bus standard

VT-d

Intel®'s Virtualization Technology for Directed I/O

XP

A Microsoft Windows release

ZIP

Compressed archive file format

8 Contact Information

ETAS Headquarters

ETAS GmbH

Borsigstraße 24	Phone:	+49 711 3423-0
70469 Stuttgart	Fax:	+49 711 3423-2106
Germany	Internet:	www.etas.com

ETAS Subsidiaries and Technical Support

For details of your local sales office as well as your local technical support team and product hotlines, take a look at the ETAS website:

ETAS subsidiaries	Internet:	www.etas.com/en/contact.php
ETAS technical support	Internet:	www.etas.com/en/hotlines.php

Index