



DRIVING EMBEDDED EXCELLENCE

# ETAS RTPC V6.5.2

## Benutzerhandbuch

## **Copyright**

---

Die Angaben in diesem Schriftstück dürfen nicht ohne gesonderte Mitteilung der ETAS GmbH geändert werden. Des Weiteren geht die ETAS GmbH mit diesem Schriftstück keine weiteren Verpflichtungen ein. Die darin dargestellte Software wird auf Basis eines allgemeinen Lizenzvertrages oder einer Einzellizenz geliefert. Benutzung und Vervielfältigung ist nur in Übereinstimmung mit den vertraglichen Abmachungen gestattet.

Unter keinen Umständen darf ein Teil dieser Veröffentlichung in irgendeiner Form ohne schriftliche Genehmigung der ETAS GmbH kopiert, vervielfältigt, in einem Retrievalsystem gespeichert oder in eine andere Sprache übersetzt werden.

© **Copyright 2022** ETAS GmbH, Stuttgart

Die verwendeten Bezeichnungen und Namen sind Warenzeichen oder Handelsnamen ihrer entsprechenden Eigentümer

ETAS RTPC V6.5.2 - Benutzerhandbuch R01 DE - 03.2022

# Inhalt

---

<b>1</b>	<b>Über dieses Dokument</b>	<b>5</b>
1.1	Klassifizierung von Warnhinweisen	5
1.2	Darstellung von Handlungsanweisungen	6
1.3	Typografische Konventionen	6
1.4	Darstellung unterstützender Informationen	6
<b>2</b>	<b>Einleitung</b>	<b>7</b>
2.1	Systemkonzept	7
2.1.1	Anwendungsfall „LABCAR Testsysteme“	7
2.1.2	Anwendungsfall „EVE - Validation of AUTOSAR Software“	7
2.2	Zielgruppe	8
2.3	Datenschutzerklärung	8
2.3.1	Datenverarbeitung durch ETAS RTPC	8
2.3.2	Daten und Datenkategorien	8
2.3.3	Technische und organisatorische Maßnahmen	9
<b>3</b>	<b>Installation</b>	<b>10</b>
3.1	Lieferumfang	10
3.2	Systemvoraussetzungen	10
3.2.1	Hardwareanforderungen	11
3.2.2	Hardwareunterstützung	11
3.2.3	Spezielle BIOS-Einstellungen	12
3.3	Installation	13
3.3.1	Installation von ETAS RTPC auf dem Real-Time PC	13
3.3.2	Installation auf EFI-Hardware	15
3.3.3	Installation von ETAS RTPC in einer virtuellen Maschine	15
3.3.4	Hardware-Integration	15
3.3.5	Systemkonfiguration	16
3.3.6	Lizenzierung	17
3.3.7	Zurücksetzen der Booteinstellung	18
3.3.8	Klonen von installierten Versionen	18
3.3.9	RTIO-Konfiguration der ES1130	20
3.3.10	Konfiguration des NTP-Servers	21
3.3.11	Clock Synchronization	21
3.4	PCI CAN- und FlexRay-Karten	22
3.5	CAN/CAN-FD- und LIN-Karten	22
<b>4</b>	<b>Betrieb</b>	<b>24</b>
4.1	Betriebsmodi	24
4.1.1	Modus „Simulation Controller“	24
4.1.2	Modus „Idle/Configuration“	25
4.1.3	Modus „Ethernet Bridge“	25
4.1.4	Modus „Power Down“	25
4.2	ETAS RTPC in der ETAS Experimentierumgebung	26
4.3	Runtime Traces	27
4.4	CAN Trace	29
4.5	Schrittweise Ausführung der Simulation	33

4.6	Real-Time Plugins (RT-Plugins) .....	35
4.6.1	Vorbereiten des LABCAR-OPERATOR-Projekts .....	36
4.6.2	Die Plugin-Dateien .....	37
4.6.3	Spezielle Präprozessor-Anweisungen .....	38
4.6.4	Templates und Beispiele .....	39
4.6.5	Vorgehensweise .....	39
4.7	CPU Core Load .....	40
4.8	Benchmark Indicator .....	41
4.9	Testen der Hardwareperformance mit „cyclictest“ .....	41
4.10	CAN/CANFD- und LIN-Abstraktion .....	43
4.10.1	Allgemeine Funktionsweise .....	43
4.10.2	Abstraktion der Busschnittstellen während des Buildvorgangs .....	44
4.10.3	Umwandeln der Kanalbeschreibung in konkrete Busschnittstellen zur Laufzeit .....	45
4.10.4	Vorlage für eine Kanalkonfiguration .....	46
<b>5</b>	<b>Konfiguration .....</b>	<b>48</b>
5.1	Einsatzgebiete .....	48
5.1.1	Model-in-the-Loop (MiL) .....	48
5.1.2	Hardware-in-the-Loop (HiL) .....	48
5.1.3	Hardware-in-the-Loop mit Mess- und Verstellzugang (M&C) .....	48
5.1.4	Hardware-in-the-Loop mit Last- und Fehlersimulation .....	49
5.1.5	Massive I/O .....	49
5.2	Das Web-Interface .....	49
5.2.1	Zugang zum Web-Interface .....	49
5.2.2	Navigation .....	50
5.2.3	Funktionen .....	52
<b>6</b>	<b>Fehlerbehebung .....</b>	<b>63</b>
<b>7</b>	<b>Glossar .....</b>	<b>66</b>
<b>8</b>	<b>Kontaktinformationen .....</b>	<b>68</b>
	<b>Index .....</b>	<b>69</b>

# 1 Über dieses Dokument

---

Dieses Handbuch besteht aus den folgenden Kapiteln:

- "Über dieses Dokument" auf Seite 5  
Dieses Kapitel
- "Einleitung" auf Seite 7  
In diesem Kapitel finden Sie das Systemkonzept, eine Beschreibung der Zielgruppe und die Datenschutzerklärung.
- "Installation" auf Seite 10  
Hier finden Informationen zu den Systemvoraussetzungen und zur Installation von ETAS RTPC einschließlich der Konfiguration der Umgebung.
- "Betrieb" auf Seite 24  
In diesem Kapitel werden die Betriebsmodi von ETAS RTPC, das Erstellen und die Migration von Projekten für ETAS RTPC beschrieben.
- "Konfiguration" auf Seite 48  
In diesem Kapitel werden die Einsatzmöglichkeiten und die Konfiguration von ETAS RTPC mit dem Web-Interface beschrieben.
- "Fehlerbehebung" auf Seite 63  
In diesem Kapitel finden Sie eine Beschreibung typischer Probleme und möglicher Lösungen dafür.
- "Glossar" auf Seite 66

## 1.1 Klassifizierung von Warnhinweisen

---

Die hier verwendeten Warnhinweise warnen vor Gefahren, die zu Personen- oder Sachschäden führen können:



### **GEFAHR**

kennzeichnet eine gefährliche Situation mit hohem Risiko, die zum Tod oder zu schweren Verletzungen führt, wenn sie nicht vermieden wird.



### **WARNUNG**

kennzeichnet eine gefährliche Situation mit mittlerem Risiko, die zum Tod oder zu schweren Verletzungen führen kann, wenn sie nicht vermieden wird.



### **VORSICHT**

kennzeichnet eine gefährliche Situation mit geringem Risiko, die zu leichten oder mittelschweren Verletzungen führen kann, wenn sie nicht vermieden wird.

**HINWEIS**

kennzeichnet eine Situation, die zu Sachschäden führen kann, wenn sie nicht vermieden wird.

## 1.2 Darstellung von Handlungsanweisungen

Das zu erreichende Ziel wird in der Überschrift definiert. Die dafür notwendigen Handlungsschritte werden in einer Schritt-für-Schritt-Anleitung aufgeführt:

**Zieldefinition**

1. Schritt 1
  2. Schritt 2
  3. Schritt 3
- > Resultat

## 1.3 Typografische Konventionen

## Software

<code>OCI_CANTxMessage msg0 =</code>	Code-Ausschnitte werden in der Schriftart Courier dargestellt. Die Bedeutung und Verwendung von Befehlen erfolgen durch die Erklärungen in den Kommentaren. Die Kommentare werden dabei in der entsprechenden Syntax dargestellt.
Wählen Sie <b>Datei</b> → <b>Öffnen</b> .	Menübefehle werden fett dargestellt.
Klicken Sie <b>OK</b> .	Schaltflächen werden fett dargestellt.
Drücken Sie <EINGABE>.	Tastaturbefehle werden in spitzen Klammern, in Kapitälchen dargestellt.
Das Dialogfenster „Datei öffnen“ erscheint.	Namen von Programmfenstern, Dialogfenstern, Feldern u.ä. werden in Anführungszeichen gesetzt.
Wählen Sie die Datei <code>setup.exe</code> aus.	Text in Auswahllisten, Programmcode sowie Pfad- und Dateinamen werden in der Schriftart Courier dargestellt.
Eine Konvertierung zwischen den Datentypen logisch und arithmetisch ist <i>nicht</i> möglich.	Inhaltliche Hervorhebungen und neu eingeführte Begriffe werden <i>kursiv</i> gesetzt.

## 1.4 Darstellung unterstützender Informationen

**INFO**

Beinhaltet zusätzliche unterstützende Informationen.

## 2 Einleitung

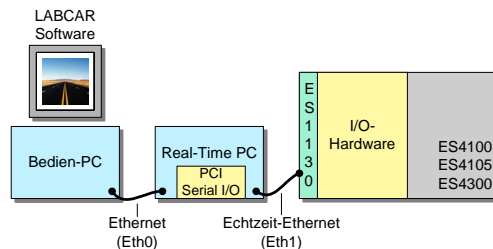
ETAS RTPC verwandelt Ihren Standard-PC in ein Hochleistungs-Simulationstarget für LABCAR Testsysteme. Indem ETAS RTPC konsequent auf kommerzielle Standardkomponenten setzt, profitieren Sie von der stetigen Weiterentwicklung der PC-Technologie hinsichtlich verfügbarer Rechenleistung und Kommunikationsbandbreite. ETAS RTPC nutzt alle Prozessorkerne moderner Intel® Mehrkernprozessoren.

### 2.1 Systemkonzept

#### 2.1.1 Anwendungsfall „LABCAR Testsysteme“

Nach Installation der ETAS RTPC-Software wird aus einem Standard-PC ein Echtzeit-PC (Real-Time PC) für LABCAR-Testsysteme.

Abb. 2-1 zeigt den Hardware-Aufbau eines ETAS RTPC-Systems. Der Real-Time PC wird über Ethernet an den Bedien-PC angeschlossen und kann in der LABCAR-Software als Simulationstarget ausgewählt werden.



**Abb. 2-1** RTPC System

Der Real-Time PC wird vollständig über ein intuitives Web-Interface vom Bedien-PC aus gesteuert. Tastatur und Monitor werden (außer bei der Installation) nicht benötigt.

Die I/O-Hardware kommuniziert über Ethernet mit dem Real-Time PC – durch den Einsatz einer hochoptimierten Punkt-zu-Punkt-Verbindung kommen die I/O-Daten deterministisch und sicher zum Ziel.

Ein PC mit ETAS RTPC wird auch als Simulationsrechner im offenen und skalierbaren ETAS HiL-System PT-LABCAR eingesetzt, das für Closed-Loop-Tests von Steuergeräten im Bereich Antriebstrang konzipiert ist.

#### 2.1.2 Anwendungsfall „EVE - Validation of AUTOSAR Software“

ISOLAR-EVE ermöglicht die Validierung AUTOSAR-konformer Software auf unterschiedlichen Plattformen – ETAS RTPC ist eine dieser Plattformen.

Mit ETAS RTPC kann ein sogenanntes VECU (virtual electronic control unit) erstellt und betrieben werden mit Zugriff auf externe Hardware wie CAN, DIO, ADC, PWM.

ISOLAR-EVE läuft auf einem Anwender-PC und steuert den Real-Time PC hauptsächlich über Automatisierungsschnittstellen zur Erstellung von Executables und deren Steuerung.

Allgemeine Konfigurationsaufgaben wie Lizenzierung und Überwachung können mit dem intuitiven Web-Interface von ETAS RTPC durchgeführt werden.

## 2.2 Zielgruppe

Dieses Handbuch richtet sich an qualifiziertes Personal aus dem Bereich „Entwicklung und Applikation von Automotive-Steuergeräten“. Spezielle Kenntnisse auf dem Gebiet der Mess- und Steuertechnik sind erforderlich.

Als Anwender sollten Sie mit dem Microsoft Windows XP, Windows Vista, oder Windows 10 Betriebssystem vertraut sein. Darüber hinaus sollten alle Anwender mit dem Windows-Dateisystem vertraut sein, insbesondere mit dem Zusammenhang von Dateien und Verzeichnissen.

Als Anwender müssen Sie die Basisfunktionen des Windows Dateimanagers und Programmmanagers bzw. des Windows Explorer kennen und mit der Drag&Drop-Funktionalität vertraut sein.

ISOLAR-EVE-Anwender, die noch nicht mit auf dem AUTOSAR-Framework basierenden Softwarebeschreibungen vertraut sind, sollten zuvor entsprechend geschult werden.

Kenntnisse über allgemeine Eigenschaften und den Betrieb von Eclipse-basierter Software sind von Nutzen, aber nicht zwingend erforderlich.

## 2.3 Datenschutzerklärung

### 2.3.1 Datenverarbeitung durch ETAS RTPC

Bitte beachten Sie, dass bei Nutzung des vorliegenden Produkts ETAS RTPC personenbezogene oder personenbeziehbare Daten verarbeitet werden.



#### INFO

Der Erwerber von ETAS RTPC ist gemäß Art. 4 Nr. 7 der Datenschutzgrundverordnung (DSGVO) für die rechtskonforme Verarbeitung der Daten verantwortlich.

Die ETAS GmbH haftet nicht als Hersteller für etwaiges Fehlverhalten im Umgang mit diesen Daten.

### 2.3.2 Daten und Datenkategorien

Bei Verwendung des ETAS License Managers in Kombination mit User-based-Licenses werden insbesondere folgende personenbezogene oder personenbeziehbare Daten oder Datenkategorien zu Zwecken der Lizenzverwaltung erfasst:

**Kommunikationsdaten:** IP-Adresse

**Benutzerdaten:** UserID



### 2.3.3 Technische und organisatorische Maßnahmen

Das vorliegende Produkt ETAS RTPC führt keine Verschlüsselung der erfassten personenbezogenen oder personenbezieharen Daten durch.

Bitte stellen Sie die Datensicherheit der aufgezeichneten Daten durch geeignete technische oder organisatorische Maßnahmen ihres IT-Systems sicher, z.B. durch klassischen Diebstahl- und Zugangsschutz auf die Hardware.

## 3 Installation

---

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- "Lieferumfang" auf Seite 10
- "Systemvoraussetzungen" auf Seite 10
  - "Hardwareanforderungen" auf Seite 11
  - "Spezielle BIOS-Einstellungen" auf Seite 12
- "Installation" auf Seite 13
  - "Installation von ETAS RTPC auf dem Real-Time PC" auf Seite 13
  - "Installation auf EFI-Hardware" auf Seite 15
  - "Installation von ETAS RTPC in einer virtuellen Maschine" auf Seite 15
  - "Hardware-Integration" auf Seite 15
  - "Systemkonfiguration" auf Seite 16
  - "Lizenzierung" auf Seite 17
  - "Zurücksetzen der Booteinstellung" auf Seite 18
  - "Klonen von installierten Versionen" auf Seite 18
  - "RTIO-Konfiguration der ES1130" auf Seite 20
  - "Clock Synchronization" auf Seite 21
- "PCI CAN- und FlexRay-Karten" auf Seite 22
- "CAN/CAN-FD- und LIN-Karten" auf Seite 22

### 3.1 Lieferumfang

---

Zum Lieferumfang von ETAS RTPC V6.5.2 gehören

- ETAS RTPC Software DVD
- Dieses Benutzerhandbuch
- Eventuell PC von ETAS

### 3.2 Systemvoraussetzungen

---

Zur Installation und Betrieb der ETAS RTPC-Software benötigen Sie Folgendes:

- ETAS RTPC Software CD
- Eine ETAS RTPC-Softwarelizenz
- Geeignete PC-Hardware (siehe "Hardwareanforderungen" auf Seite 11)
- Ethernet-Crossover-Kabel (2x RJ45-Buchse) für die Verbindung zwischen Real-Time PC und Bedien-PC

### 3.2.1 Hardwareanforderungen

ETAS RTPC ist kompatibel zu den meisten Standard-PCs. Zum Betrieb der Software müssen folgende Anforderungen erfüllt sein:

Prozessor	Intel® Xeon® Familie Intel® Core™ i Familie Intel® Core™ 2 Familie (Duo und Quad)
Ethernet-Adapter für Verbindung zum Bedien-PC	Alle vom Linux-Kernel 3.18 unterstützten Ethernet-Adapter
Ethernet-Adapter für Verbindung zu ES1130	Alle vom Linux-Kernel 3.18 unterstützten Ethernet-Adapter, z.B. - Intel® PRO/100 card S, revision 12 (0C) oder höher - Intel® PRO/1000 family - Intel® PRO/1000 PT Quad Port Server Adapter - Intel® Gigabit CT Desktop Adapter
RAM	Min. 4 GB, empfohlen 8 GB oder mehr
Festplatte	IDE oder SATA, min. 250 GB
Grafik	VGA-kompatibel
Monitor	Nur für die Grundinstallation benötigt
Tastatur	PS/2 oder USB (nur für Grundinstallation benötigt)
CD-ROM Laufwerk	IDE, SATA oder USB, wird nur für die Grundinstallation benötigt (bootfähig)
Betriebssystem	Nicht erforderlich

#### INFO

Während des Betriebs von ETAS RTPC werden bestimmte Erfolgs- oder Fehlermeldungen in Form von Tonfolgen ausgegeben. Sollte Ihr PC kein Ausgabegerät für akustische Signale besitzen, können Sie den aktuellen Status auch über das Web-Interface (siehe Abschnitt "Das Web-Interface" auf Seite 49) feststellen.

Genauere Abgaben zur Kompatibilität von PCs verschiedener Hersteller finden Sie in der Kompatibilitätsliste (<http://192.168.40.14/documentation/>).

### 3.2.2 Hardwareunterstützung

ETAS RTPC unterstützt eine maximale Anzahl an Hardwareanbindungen in Verbindung mit ES5300 Gehäusen.

#### INFO

Bevor Sie ein System mit der vorgegebenen maximalen Anzahl an Hardwareanbindungen planen oder aufbauen, kontaktieren Sie Ihren ETAS Ansprechpartner. Der ETAS Ansprechpartner unterstützt Sie bei der individuellen Hardwareauswahl.

In folgender Tabelle ist die maximale Anzahl der Hardwareanbindungen aufgelistet:


Hardware	Maximale Anzahl
CAN-Karten	32
Netzwerkports	32
FlexRay-Karten	8
RS232-Ports/RS485-Ports	16

Durch die Erweiterung der Anzahl von RS232-Ports/RS485-Ports auf 16 kann sich die Reihenfolge der Ports ändern. Wenn Sie diese Ports verwenden, überprüfen Sie die Hardwarekonfiguration und passen Sie diese entsprechend an.

### 3.2.3 Spezielle BIOS-Einstellungen

#### 3.2.3.1 Boot-Einstellungen

Damit die Installation von CD-ROM funktioniert, muss das BIOS des Rechners so konfiguriert werden, dass der PC von der CD-ROM bootet („primary boot device = CD-ROM“). Wenn Sie den PC später ohne Tastatur betreiben wollen, müssen Sie auch diese Einstellung vornehmen („no halt on keyboard“). Die Namen der Einstellungen in den Klammern können variieren - Details dazu finden Sie in der Bedienungsanleitung des PCs.

 **INFO**

Eine Dokumentation der BIOS-Einstellungen der ETAS PC-Hardware finden Sie unter <http://192.168.40.14/documentation/>.

#### 3.2.3.2 Hyper-Threading

Die Verwendung von Hyper-Threading hat insofern Auswirkungen auf die Echtzeit-Performance des Real-Time PC, als dass bestimmte Ressourcen wie FPU und ALU von zwei verknüpften Prozessorkernen gemeinsam genutzt werden können. Dies beeinflusst die Ausführung von Code auf den verknüpften Kernen, weil u.U. ein Kern auf die Freigabe der von einem anderen Kern verwendeten Ressource warten muss.

Wenn es zu Problemen in Form von Echtzeit-Verletzungen kommt, kann es hilfreich sein, das Hyper-Threading in den BIOS-Einstellungen zu deaktivieren. Insbesondere wenn die Modelle mit sehr schnellen Zykluszeiten (< 100 µs) ausgeführt werden, ist eine Deaktivierung des Hyper-Threading empfehlenswert.

Es kann aber durchaus Situationen geben, bei denen die Verwendung von Hyper-Threading zu zusätzlicher Rechenleistung führt, ohne dass es zu Echtzeit-Verletzungen kommt. Typische Szenarien dafür sind Simulationsmodelle, die mit relativ großen Zykluszeiten (> 500 µs) ausgeführt werden oder Real-Time PCs, die ausschließlich zu Build-Zwecken verwendet werden.

### 3.3 Installation

Die Installation von ETAS RTPC besteht aus vier Teilen, die in den folgenden Abschnitten beschrieben werden:

- "Installation von ETAS RTPC auf dem Real-Time PC" auf Seite 13
- "Installation von ETAS RTPC in einer virtuellen Maschine" auf Seite 15
- "Hardware-Integration" auf Seite 15
- "Systemkonfiguration" auf Seite 16
- "Lizenzierung" auf Seite 17
- "Zurücksetzen der Booteinstellung" auf Seite 18
- "Klonen von installierten Versionen" auf Seite 18
- "RTIO-Konfiguration der ES1130" auf Seite 20
- "Clock Synchronization" auf Seite 21



#### INFO

ETAS bietet ETAS RTPC auch vorinstalliert auf von ETAS empfohlener und getesteter PC-Hardware an. In diesem Fall beginnen Sie bitte gleich mit dem Abschnitt "Hardware-Integration" auf Seite 15.

#### 3.3.1 Installation von ETAS RTPC auf dem Real-Time PC

Falls die Installation von ETAS RTPC von ETAS durchgeführt wurde, fahren Sie bitte mit dem nächsten Abschnitt "Hardware-Integration" auf Seite 15 fort.

1. Schließen Sie Monitor und Tastatur an die PC-Hardware an.
2. Schalten Sie den PC und den Monitor ein.
3. Legen Sie die ETAS RTPC Software DVD in das DVD-Laufwerk des Real-Time PCs ein.
4. Wählen Sie im Startbildschirm die gewünschte Option (Voreinstellung:  
`ETAS RTPC Vx.y.z - Standard Install`).



#### INFO

Bei dieser Option wird das Dateisystem „ext4“ installiert – wenn Sie mit ETAS RTPC Versionen vor V5.1.0 kompatibel sein müssen (Dateisystem „ReiserFS“), wählen Sie die Option „Install for parallel use with RTPC versions before V5.1.0“.

5. Warten Sie, bis der Installationsdialog von ETAS RTPC erscheint.
6. Wählen Sie mit den Cursortasten die Option [`Install RTPC software`].

7. Drücken Sie <ENTER>.



### INFO

Alle auf der Festplatte befindlichen Daten werden durch die Installation von ETAS RTPC gelöscht!

8. Bestätigen Sie die Warnung durch Auswahl der Option [Yes, continue with installation].
9. Warten Sie, bis ETAS RTPC fertig installiert ist (Dauer etwa 5 Minuten).
10. Entnehmen Sie die ETAS RTPC Software DVD.
11. Wählen Sie die Option [Reboot] aus.  
Der PC startet neu.



12. Warten Sie, bis der PC eine aufsteigende Tonfolge spielt.  
ETAS RTPC wurde erfolgreich installiert.



13. Fahren Sie den PC durch kurzes Drücken des Ein/Aus-Tasters herunter.

Nach wenigen Sekunden spielt der PC eine absteigende Tonfolge und schaltet sich ab.

Dies zeigt an, dass ETAS RTPC ordnungsgemäß heruntergefahren wurde und von der Netzspannung getrennt werden darf.

Genauere Informationen über diese akustischen Signale finden Sie im Kapitel "Betrieb" auf Seite 24.



### INFO

ETAS RTPC verwendet zwar ein robustes Dateisystem, trotzdem können Hardware und Software durch ein Ausschalten ohne Herunterfahren oder einen plötzlichen Spannungsverlust beschädigt werden. ETAS empfiehlt deshalb, ETAS RTPC durch kurzes Drücken des Ein/Aus-Tasters herunterzufahren, um diese Probleme zu vermeiden.

14. Entfernen Sie Monitor und Tastatur (siehe auch "Spezielle BIOS-Einstellungen" auf Seite 12).

#### Erreichbarkeit von ETAS RTPC prüfen

1. Wechseln Sie zum Host.
2. Wählen Sie **Ausführen** aus dem Startmenü.
3. Geben Sie „cmd“ ein.

Die Eingabeaufforderung wird geöffnet.

4. Geben Sie Folgendes ein:

```

ping 192.168.40.14
U:\>ping 192.168.40.14

Ping wird ausgeführt für 192.168.40.14 mit 32 Bytes Daten:

Antwort von 192.168.40.14: Bytes=32 Zeit=4ms TTL=64
Antwort von 192.168.40.14: Bytes=32 Zeit<10ms TTL=64
Antwort von 192.168.40.14: Bytes=32 Zeit<10ms TTL=64
Antwort von 192.168.40.14: Bytes=32 Zeit<10ms TTL=64

Ping-Statistik für 192.168.40.14:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
    Ca. Zeitangaben in Millisek.:
        Minimum = 0ms, Maximum = 4ms, Mittelwert = 1ms

```

5. Alternativ können Sie auch in Ihrem Browser das Web Interface (<http://192.168.40.14>) öffnen.

### 3.3.2 Installation auf EFI-Hardware

ETAS RTPC V6.5.2, V6.5.1, V6.5.0, V6.4.1 und V6.4.0 unterstützen das Booten auf EFI-Hardware. Bei der Installation auf einer EFI-Hardware ist es nicht empfehlenswert, ältere Versionen als ETAS RTPC V6.4.0 parallel zu installieren. Von ETAS RTPC V6.4.0 und höher aus funktioniert die Installation. Allerdings kennen die älteren Versionen den EFI-Boot nicht.

#### **HINWEIS**

Die Installation einer anderen älteren Version von ETAS RTPC aus einer Version vor ETAS RTPC V6.4.0 führt zur Beschädigung des RTPC.

### 3.3.3 Installation von ETAS RTPC in einer virtuellen Maschine

Die ETAS RTPC Software kann in einer virtuellen Maschine installiert werden. Verwenden Sie dafür das Installations-Image. Die virtuelle Maschine muss entsprechend der Systemanforderungen des ETAS RTPC konfiguriert sein (siehe "Systemvoraussetzungen" auf Seite 10). Für die Konfiguration und Installation der virtuellen Maschine konsultieren Sie bitte die Anleitung des entsprechenden Tools.

Für den Betrieb des ETAS RTPC in einer virtuellen Maschine empfiehlt ETAS die Verwendung eines "Host-only" Interfaces als Netzwerkschnittstelle eth0. Die Schritte zur Konfiguration eines solchen Netzwerk-Interfaces entnehmen Sie bitte der Dokumentation des entsprechenden Tools.

### 3.3.4 Hardware-Integration

Dieses Kapitel beschreibt die Integration des Real-Time PC in ein Hardware-in-the-Loop-Test-System (HiL). Weitere Konfigurationen wie z.B. Model-in-the-Loop- oder HiL-System mit Mess- und Verstellzugang sind in Kapitel "Einsatzgebiete" auf Seite 48 erklärt.

ETAS RTPC verwendet für die Onboard-Ethernet-Schnittstelle die Bezeichnung „Eth0 (Host)" - die Ethernet-Karte trägt die Bezeichnung „Eth1 (HW)".

### Bedien-PC und ES1130 anschließen

1. Schließen Sie den Real-Time PC über die Onboard-Ethernet-Schnittstelle (Eth0) mit einem Cross-Over-Kabel an die Experimental-Target-Schnittstelle des Bedien-PCs.



#### INFO

Cross-Over-Kabel haben überkreuzte Verbindungen und sind oft an einem oder beiden Enden mit gelben oder orange-farbenen Steckverbindern gekennzeichnet.

2. Verbinden Sie den Real-Time PC über die Ethernet-Schnittstelle (Eth1) mit dem ES1130 Simulation Controller Board. Für diese Verbindung kann das ETAS Kabel CBE100.1 (für ES1130.2 und ES1130.3) oder K107 (für ES1130.1) verwendet werden.

## 3.3.5 Systemkonfiguration

ETAS RTPC lässt sich bequem vom Bedien-PC aus über einen Web-Browser (z.B. Internet Explorer oder Mozilla Firefox) konfigurieren – der Konfigurationsdialog ist über die IP-Adresse 192.168.40.14 erreichbar.

Mit den im Folgenden beschriebenen Schritten konfigurieren Sie ETAS RTPC für Hardware-in-the-Loop-Tests (HiL). Weitere Informationen zu den Konfigurationsmodi finden Sie im Kapitel "Einsatzgebiete" auf Seite 48.

### Konfigurationsdialog öffnen

1. Starten Sie den Real-Time PC durch Druck auf den Ein/Aus-Taster an der Vorderseite.
2. Schalten Sie das Chassis mit der I/O-Hardware ein.
3. Warten Sie einige Sekunden, bis Sie die aufsteigende Tonfolge hören.
4. Starten Sie auf dem Bedien-PC einen Web-Browser.
5. Rufen Sie die URL <http://192.168.40.14> auf.



#### INFO

Sie können für diese IP-Adresse auch einen leichter zu merkenden Alias vergeben - Näheres dazu finden Sie in Abschnitt "Zugang zum Web-Interface" auf Seite 49.

Der Konfigurationsdialog von ETAS RTPC wird geöffnet.

Zur Konfiguration der ETAS RTPC-Software muss der Simulation Controller gestoppt werden (siehe auch "Modus „Simulation Controller“" auf Seite 24.



#### INFO

Solange der Simulationscontroller läuft, ist der Link *Main Page* inaktiv!



**Simulationscontroller anhalten**

1. Klicken Sie die Schaltfläche **Stop Simulation Controller**.
2. Klicken Sie auf den Link **Main Page**.

**Konfiguration durchführen**

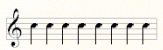
1. Klicken Sie auf den Link **Configure ETAS RTPC**.
2. Klicken Sie die Schaltfläche **Autoconfigure**.

Die Autokonfiguration wird durchgeführt und das Ergebnis im Feld „RTPC\_USAGE\_ETH1“ eingestellt.

**Simulationscontroller wieder starten**

1. Klicken Sie auf den Link **Main Page**.
2. Klicken Sie auf den Link **Simulation Controller**.
3. Klicken Sie die Schaltfläche **Start Simulation Controller**.

Wenn Sie die aufsteigende Tonfolge hören, ist der Simulationscontroller wieder aktiv.


 **INFO**

Wenn statt dessen die Fehlerntonfolge erklingt, konnte der Simulationscontroller nicht gestartet werden. Nähere Hinweise dazu finden Sie im Kapitel "Fehlerbehebung" auf Seite 63

Damit ist die Konfiguration für den HiL-Modus erfolgreich durchgeführt. Informationen zu den anderen Konfigurationsmodi finden Sie im Kapitel "Fehlerbehebung" auf Seite 63.

### 3.3.6 Lizenzierung

Zum Betrieb von ETAS RTPC V6.5.2 benötigen Sie eine Lizenz, die Sie in Form einer Lizenzdatei im [ETAS License Portal](#) erhalten können. Einzelheiten dazu finden Sie in den Handbüchern „LABCAR-OPERATOR V5.4.13 - Schnelleinstieg" oder „ETAS ISOLAR-EVE Vx.y<sup>1</sup> - Installation Guide" oder „COSYM V2.5.0 - Installation Guide".

 **INFO**

Die Installation der Lizenz für ETAS RTPC erfolgt aber nicht - wie oben beschrieben - über den ETAS License Manager, sondern im Web-Interface (siehe "License Management" auf Seite 60).

**Lizenzversionen**


- RTPC\_EVE

Für die Ausführung von ISOLAR-EVE-Projekten

---

1. Die entsprechende Version von ISOLAR-EVE war zum Zeitpunkt der Freigabe von ETAS RTPC V6.5.2 noch nicht festgelegt.

- LCS\_RTPC\_RT  
Für die Ausführung von LABCAR-OPERATOR-Projekten in Echtzeit
- COSYM\_RT  
Für den Build-Vorgang eines COSYM-Projekts und dessen Ausführung in Echtzeit
- LCS\_RTPC\_OLC  
Bei der Ausführung von LABCAR-OPERATOR-Projekten und von COSYM-Projekten beträgt die minimale Periodendauer 200 ms.

 **INFO**

Für den Build-Vorgang eines LABCAR-OPERATOR-Projektes wird keine Lizenz benötigt!

### 3.3.7 Zurücksetzen der Booteinstellung


Wenn es bei der Änderung von Boot-Einstellungen Probleme gibt (z.B. nach einer Repartitionierung), können diese Einstellungen zurückgesetzt werden.

#### Booteinstellungen zurücksetzen

1. Wählen Sie **Main Page >> Configuration**.
2. Gehen Sie zum Abschnitt „System Boot Settings“.
3. Wählen Sie eine Version.
4. Klicken Sie **Set Boot Version**.

**System Boot Settings**

**Caution:** Modification of the values below requires a fundamental knowledge of RTPC.

<p><b>Select version to boot</b></p> <p>This version will be used with the next system boot.</p> <p>Info: The current system runs from partition sda7</p>	<div style="border: 1px solid gray; padding: 2px;">sda7: ETAS-RTPC </div>
---	--

[Expert Edit](#) [Reset Boot Settings](#)

**Reboot System**

Reboot RTPC using the selected boot version.

5. Führen Sie mit **Reboot System** einen Neustart durch.

### 3.3.8 Klonen von installierten Versionen

Sie können bereits installierte Versionen von ETAS RTPC klonen, indem Sie eine installierbare Datei (\*.tgz) erzeugen und diese auf Ihren Bedien-PC herunterladen.

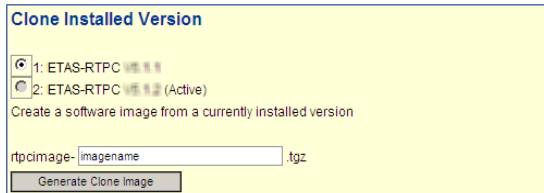
#### Version klonen

1. Wählen Sie **Main Page >> Installation / Update >> Clone Installed Version**.

- Wählen Sie eine der installierten Versionen.



- Geben Sie einen Namen für den zu erstellenden Klon ein.
- Klicken Sie **Generate Clone Image**.



Der Vorgang wird gestartet und der Fortschritt im Web-Interface angezeigt.



### Cloning finished

- [Select Boot Version](#)
- [Return to Installation / Update](#)

### Cloning-Progress

```
Cloning Partition '/dev/sda5' to 'rtpcimage-6.5.2.0.tgz'
action=creating-tar
Creating tar file from partition '/dev/sda5' ...
tar: ./var/run/proftpd/proftpd.sock: socket ignored
Special processing of some files ...
/etc/fstab
action=compressing-tar
Compressing the tar file (this takes a while) ...
--- finished ---
```

Nach Beendigung des Vorgangs können Sie das Abbild zur Weiterverwendung auf Ihren Bedien-PC herunterladen.

### Klon herunterladen

- Klicken Sie den Link **Return to Installation / Update**.
- Gehen Sie zu „Download Software Image“.
- Wählen Sie die eben erstellte Version.

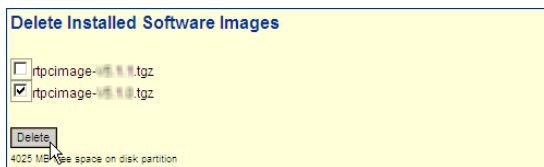
4. Klicken Sie **Download**.



Es wird ein Dialogfenster zum Herunterladen geöffnet, mit dem Sie diese Datei speichern können.

5. Wenn Sie diese Datei von Ihrem Real-Time PC löschen wollen, wählen Sie das Abbild unter „Delete Installed Software Images“.

6. Klicken Sie **Delete**.

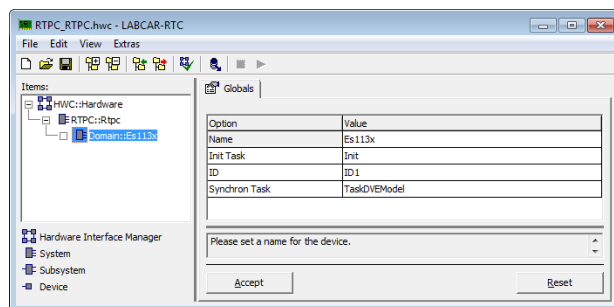


Die Datei wird vom Real-Time PC gelöscht.

### 3.3.9 RTIO-Konfiguration der ES1130

Bei einer ES1130, die logisch hinter einem Real-Time PC angeschlossen ist, wird die Kommunikation zwischen ETAS RTPC und ES1130 synchron zu einer wählbaren Task durchgeführt. Diese Auswahl erfolgt in LABCAR-RTC (RTIO) in der Registerkarte „Globals“ der ES1130 ("Synchron Task").

Die besten Ergebnisse werden erzielt, wenn das Simulationsmodell und die periodischen Hardwarezugriffe in genau einer Task laufen (Task „TaskDVE-Model“).



### 3.3.10 Konfiguration des NTP-Servers

Um einen NTP<sup>1</sup>-Server verwenden zu können, muss die Umgebungsvariable `RTPC_NTP_SERVER_ADDR` in die Adresse des NTP-Servers eingetragen werden, der über den RTPC erreichbar ist (meist der Host-PC). Verwenden Sie die Ansicht "expert edit" des Web-Interface, um diese Variable einzutragen.

### 3.3.11 Clock Synchronization

Zur Synchronisation der Zeit auf dem Bedien-PC und dem Real-Time PC kann der in Windows integrierte NTP-Server aktiviert werden.

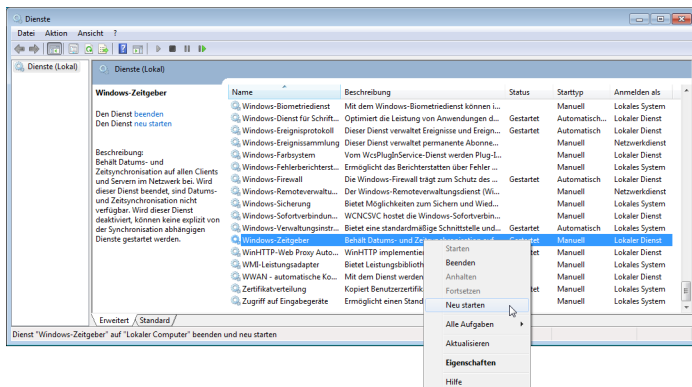
Die Aktivierung erfolgt über Einträge in die Windows Registrierdatenbank – die Datei zum Setzen der entsprechenden Schlüssel finden Sie über das Web-Interface unter

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

Damit wird der Windows-Dienst „ws32time“ konfiguriert – nach dem Ändern der Schlüssel muss der Server neu gestartet werden.

#### NTP-Server neu starten

1. Wählen Sie **Systemsteuerung** → **Verwaltung** → **Dienste**.
2. Wählen Sie den Dienst „Windows-Zeitgeber“.
3. Drücken der rechten Maustaste öffnet das Kontextmenü.
4. Wählen Sie **Neu starten**.



#### **INFO**

Bitte beachten Sie, dass es nach dem Neustart des Dienstes einige Minuten dauern kann, bis die Zeit präzise genug ist, um anderen Clients zur Verfügung gestellt zu werden.

Eine Ad-Hoc-Synchronisation der Uhren ist auch im Web-Interface möglich (siehe "**Clock Synchronization**" auf Seite 57).

1. Network Time Protocol

## 3.4 PCI CAN- und FlexRay-Karten

Zum ETAS Real-Time PC können für die Kommunikation zwischen Modell und Steuergerät CAN- und FlexRay-Boards erworben werden.

## 3.5 CAN/CAN-FD- und LIN-Karten

ETAS RTPC V6.5.2 verwendet die Version 1.13.3877 der IXXAT®-Treiber für CAN/CANFD- und LIN-Unterstützung.

```
ECI Device Admin 1.11 build 3877 Sep  1 2020 [Linux amd64] (release)
-----
HMS Technology Center Ravensburg GmbH
Software package ECI Driver 1.13.3877.0
(C) 2020 HMS Technology Center Ravensburg GmbH, all rights reserved
```

Da dies eine neue Version ist, muss die Firmware der installierten IXXAT IB200/IB600/IB640-Karten einmal aktualisiert werden. Um das Update durchzuführen, verwenden Sie bitte das ETAS RTPC-Web-Interface und navigieren Sie zu **Main Page >> System Info >> LABCAR I/O Hardware**.

Im Abschnitt **Firmware version check for IXXAT Boards** finden Sie eine Liste der installierten IB200/IB600/IB640-Karten und der installierten Firmware. Die zugehörige Schaltfläche **Update Firmware** ist unter dieser Liste platziert. Diese Schaltfläche wird deaktiviert, wenn keine Aktualisierung erforderlich ist. Abb. 3-1 zeigt das Beispiel einer Übereinstimmung der IXXAT-Firmwareversion.

**INFO**

Eventuell ist es nach Durchführung des Firmware-Updates der installierten IB200/IB600/IB640-Karten nicht mehr möglich, auf eine vorherige Version zurückzugreifen. Bitte kontaktieren Sie Ihren ETAS Ansprechpartner bei Fragen.

**ETAS RTPC**  
\* Global version \*

Run your models in real time on a PC

**CAN LIN**

[Main Page >> System Info >> LABCAR I/O Hardware](#)

▼ **Detected VME Boards**

Note: This information has been detected at the previous download of an executable with configured ES1130 systems. It shows the boards of all VME systems that are configured to be used with the ETAS RTPC.  
[Refresh Display](#)

▶ **Detected PCI/PCle boards that are supported by ETAS RTPC**

▼ **Detected ETAS LABCAR PCle Boards**

- No boards detected

▼ **Firmware version check for IXXAT Boards**

IXXAT Firmware Version Details							
Hardware Type	Serial No.	Hardware Version	BootManager Version	Firmware Version	Expected Hardware Version	Expected BootManager Version	Expected Firmware Version
CAN-IB200/PCe	HV289409	00.01.09.01	00.03.00.09	00.03.17.00	00.01.09.01	00.03.00.09	00.03.17.00

[Update Firmware](#)

**Abb. 3-1** Beispiel einer Übereinstimmung der IXXAT-Firmwareversion

Stimmen die Firmwareversionen der IXXAT IB200/IB600/IB640-Karten nicht überein, werden die Controller während der Initialisierung der Simulation nicht geöffnet. Eine Fehlermeldung wird in die Protokolldatei geschrieben und es werden weder Nachrichten gesendet noch empfangen. Dieses Verhalten ist identisch mit dem Fall, dass ein konfigurierter Controller nicht vorhanden ist. Abb. 2-2 zeigt ein Beispiel für eine Fehlermeldung in der Protokolldatei und Abb. 2-3 zeigt das Beispiel für eine Nichtübereinstimmung einer IXXAT-Firmwareversion.

```
[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.
```

Abb. 3-2 Beispiel für eine Fehlermeldung in der Protokolldatei

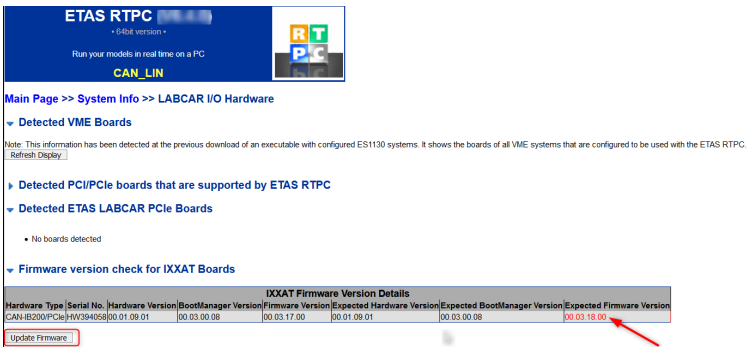


Abb. 3-3 Beispiel für eine Nichtübereinstimmung einer IXXAT-Firmwareversion

## 4 Betrieb

---

In diesem Kapitel finden Sie Informationen zum Arbeiten mit ETAS RTPC V6.5.2.

Im Einzelnen sind dies:

- "Betriebsmodi" auf Seite 24
- "ETAS RTPC in der ETAS Experimentierumgebung" auf Seite 26
- "Runtime Traces" auf Seite 27
- "CAN Trace" auf Seite 29
- "Schrittweise Ausführung der Simulation" auf Seite 33
- "Real-Time Plugins (RT-Plugins)" auf Seite 35
- "CPU Core Load" auf Seite 40
- "Benchmark Indicator" auf Seite 41
- "Testen der Hardwareperformance mit „cyclictest“" auf Seite 41

### 4.1 Betriebsmodi

---

ETAS RTPC V6.5.2 besitzt vier Betriebsmodi, von denen jeweils ein Modus aktiv ist:

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

Über das Web-Interface können Sie konfigurieren, welcher Betriebsmodus nach dem Einschalten des Real-Time PCs aktiviert werden soll (siehe "**RTPC\_POWER\_UP\_MODE**" auf Seite 55).

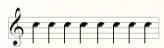
#### 4.1.1 Modus „Simulation Controller“

Der Simulationscontroller-Modus dient zum Ausführen von Echtzeit-Modellen. Eine aufsteigende Tonfolge zeigt an, dass der Simulationscontroller erfolgreich gestartet wurde.



Standardmäßig wird der Simulationscontroller nach dem Hochfahren des Real-Time PC automatisch aktiviert.

Kann der Simulationscontroller aufgrund eines Konfigurationsproblems nicht gestartet werden, wird dies durch acht gleich hohe Töne angezeigt.



In diesem Fall geht ETAS RTPC automatisch in den Zustand „Idle/Configuration“ (siehe "Modus „Idle/Configuration“" auf Seite 25) über.



Mögliche Fehlerursachen finden Sie im Kapitel "Fehlerbehebung" auf Seite 63.

### INFO

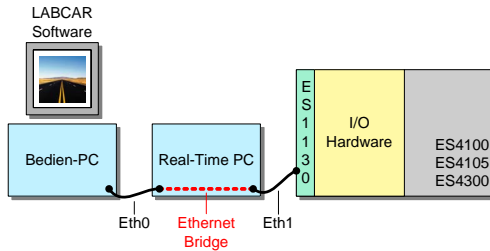
Bei jedem Start des Simulationscontrollers wird ein VMEbus-Reset durchgeführt!

#### 4.1.2 Modus „Idle/Configuration“

Der Modus „Idle/Configuration“ dient zur Konfiguration von ETAS RTPC und zur Aktualisierung der Software (Update). ETAS RTPC kann in diesem Modus keine Echtzeitmodelle ausführen.

#### 4.1.3 Modus „Ethernet Bridge“

Dieser Modus stellt eine transparente Verbindung zwischen den beiden Ethernet-Schnittstellen Eth0 und Ethn her. Hiermit kann direkt vom Bedien-PC auf das ES1130 Simulation Controller Board zugegriffen werden – ein manuelles Umstecken der Ethernet-Kabel entfällt damit.



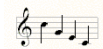
### INFO

Der Modus „Ethernet Bridge“ kann für jeden Anschluss „Ethn“ separat konfiguriert werden!

Zum Firmware-Update der I/O-Hardware (HSP oder LABCAR-Firmware-Update) wird der Ethernet-Bridge-Modus automatisch aktiviert.

#### 4.1.4 Modus „Power Down“

Der „Power Down“ Betriebsmodus wird entweder durch kurzen Druck auf den Ein/Aus-Taster an der Vorderseite des Real-Time PC oder über das Web-Interface aktiviert. ETAS RTPC fährt das System herunter, spielt eine Tonfolge (vier absteigende Töne) und schaltet den Real-Time PC ab.



Der Real-Time PC kann danach von der Netzspannung getrennt werden.

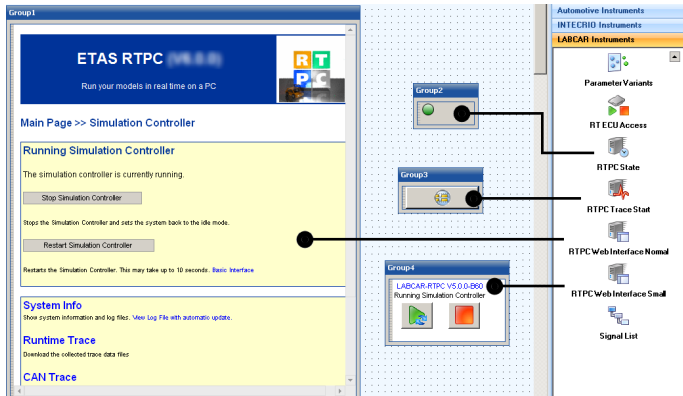
### INFO

ETAS RTPC verwendet zwar ein robustes Dateisystem, trotzdem können Hardware und Software durch ein Ausschalten ohne Herunterfahren oder einen plötzlichen Spannungsverlust beschädigt werden. ETAS empfiehlt deshalb, ETAS RTPC durch kurzes Drücken des Ein/Aus-Tasters herunterzufahren, um diese Probleme zu vermeiden.

## 4.2 ETAS RTPC in der ETAS Experimentierumgebung

In der ETAS Experimentierumgebung gibt es eine Reihe von Instrumenten, die den Zugriff auf ETAS RTPC ermöglichen.

Diese finden Sie in der Experimentierumgebung im Fenster „Instruments“ in der Gruppe „LABCAR Instruments“. Von dort können Sie wie gewohnt in den Hauptarbeitsbereich (Register „Instrumentation“) gezogen werden.



Im Einzelnen sind dies:

- **RTPC State**  
Ein laufender Real-Time PC wird durch eine grüne LED symbolisiert.
- **RTPC Web Interface Normal**  
Dieses Instrument enthält die volle Darstellung des Web-Interfaces wie im HTML-Browser.
- **RTPC Web Interface Small**  
Dieses Instrument ermöglicht das Starten (bzw. Neustarten) und Stoppen des Simulationscontrollers.
- **RTPC Trace Start**  
Damit können Sie einen RTPC Trace starten. Die Dateien (und der Zugriff darauf) sind im Abschnitt "Runtime Traces" auf Seite 27 beschrieben.

## 4.3 Runtime Traces

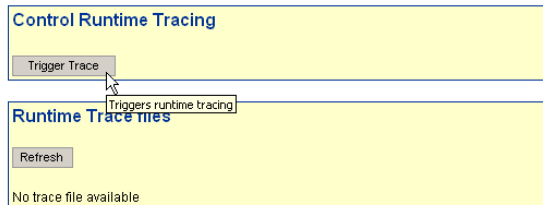
Ein Runtime Trace protokolliert Prozesse und Tasks auf dem Real-Time PC und speichert diese in Dateien mit der Endung .rtt, die mit RTA-TRACE analysiert und als Zusammenfassung in Form von xml-, html- und csv-Datei gespeichert werden können.

### Einen Runtime Trace ausführen

1. Wählen Sie **Main Page >> Simulation Controller >> Runtime Trace**.
2. Unter „Control Runtime Tracing“, klicken Sie **Trigger Trace**.



**Main Page >> Simulation Controller >> Runtime Trace**



Der Runtime Trace wird gestartet und in einer Datei aufgezeichnet.

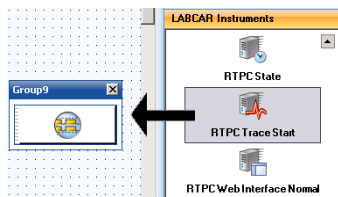
### **i** INFO

Der Umfang eines Runtime Trace (= die Zahl der aufgezeichneten Ereignisse) wird mit der Größe „RTOS\_TRACE\_BUFFERSIZE“ (siehe **“RTPC Configuration“** auf Seite 53) festgelegt.

### Einen Runtime Trace in der Experimentierumgebung starten

Sie können Runtime Traces auch aus der Experimentierumgebung starten.

1. Wählen Sie dazu das Instrument „RTPC Trace Start“ (aus den „LABCAR Instruments“ im Fenster „Instruments“) und ziehen Sie es in einen Layer.



- Zum Starten der Runtime Traces klicken Sie auf die Schaltfläche.



### Datei mit Runtime Trace öffnen

Die Dateien mit den Resultaten eines Runtime Trace sind an derselben Stelle zugänglich:

- Wählen Sie ein Format neben dem Eintrag für die jeweilige Datei.

The screenshot shows the ETAS RTPC web interface. At the top, there is a blue banner with the text "ETAS RTPC" and "Run your models in real time on a PC". Below the banner, there is a navigation path: "Main Page >> Simulation Controller >> Runtime Trace". The main content area is divided into two sections: "Control Runtime Tracing" and "Runtime Trace files".

**Control Runtime Tracing**

Trigger Trace

**Runtime Trace files**

Refresh

Download all `rtt`, `xml`, `html`, `csv` trace files in a single zip file.

Trace File	Type	Date
rttc-trace-111115-1613-00	.rttc .xml .html .csv	15-Nov-2012 16:14:08
rttc-trace-111115-1613-01	.rttc .xml .html .csv	15-Nov-2012 16:14:27
rttc-trace-111115-1613-02	.rttc .xml .html .csv	15-Nov-2012 16:15:13

Je nach gewähltem Format wird die Datei geöffnet oder es wird ein Dialog zum Speichern der Datei geöffnet.

- Um die HTML-Datei eines RTPC-Trace darzustellen, klicken Sie auf die Dateiendung in der Spalte „Type“.

Die Datei wird im Browser geöffnet.

### 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
lctrl_OneStep_Outputs_IdleController	17	812	2	4	82
lctrl_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.

Die Bedeutung der Spalten ist wie folgt:

- Id: Die interne ID des Prozesses
  - Calls: Anzahl der Aufrufe (während der Trace aktiv war)
  - RTmin, RTavg, RTmax: Minimale, durchschnittliche und maximale Laufzeit des jeweiligen Prozesses.
- Sie können auch alle dargestellten Dateien eines bestimmte Typs in Form eines ZIP-Archivs herunterladen. Klicken Sie hierzu den jeweiligen Link in der Zeile „Download all ...“.

## 4.4 CAN Trace

Die CAN Trace Funktionalität ermöglicht die Überwachung des CAN-Verkehrs von und zu den IXXAT CAN-Karten – dabei wird der Inhalt der CAN-Messages (Send und Receive) als UDP/IP-Stream zum einem Host-PC übermittelt.

Dafür benötigen Sie auf Ihrem Host-PC einen UDP-Monitor wie z.B. Netcat for Windows – gestartet wird der CAN Trace dann im Web-Interface (siehe "CAN Trace im Web-Interface starten" auf Seite 30).

Eine andere Möglichkeit ist die Verwendung des mitgelieferten Perl-Skripts `labcar-rtpc-can-trace.pl` (wählen Sie **CAN Trace** und klicken Sie **Help**), mit dem sowohl der CAN Trace aktiviert wird als auch die Daten in ein Fenster der Eingabeaufforderung ausgegeben werden.

### CAN Trace im Web-Interface starten

1. Starten Sie den Simulationscontroller im Web-Interface von ETAS RTPC.
2. Laden Sie (in der Experimentierumgebung) Ihr Projekt herunter.
3. Starten Sie die Simulation.
4. Wechseln Sie wieder in das Web-Interface von ETAS RTPC.

Bei laufendem Simulationscontroller wird Ihnen die Seite **Main Page >> Simulation Controller** angezeigt.

5. Wählen Sie **CAN Trace**.
6. Geben Sie folgende Daten ein:
  - IP Address  
Die IP-Adresse, an die der Stream gesendet werden soll.
  - UDP Port  
Der UDP-Port, an den der Stream gesendet werden soll (siehe "**port=n**" auf Seite 31).
  - Ctrl Mask  
Die Bitmaske zur Auswahl des Boards und des jeweiligen Controllers (siehe "**ctrlmask=n**" auf Seite 32).
  - Ausgabemodus des Streams  
Momentan nur „ascii“ möglich.

### **Main Page >> Simulation Controller >> CAN Trace**

#### CAN Trace Control

The tracing is currently stopped

**Specify data stream receiver:**

IP Address:

UDP Port:

Ctrl Mask:

Mode:

[Help](#)

#### CAN Trace Filter

Filter inactive

Specify a space-separated list of CAN ids to be added to the filter list.  
An '!' before an id removes this id from the filter list.  
The value 'reset' deactivates the CAN trace filter.  
Each start of the CAN trace resets the filter values.  
Example: '0x120 60 !0x100'

[Help](#)

7. Klicken Sie **Start Tracing**, um die mit diesen Argumenten gebildete HTTP-Anfrage:

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

an den Real-Time PC zu senden.

8. Wenn Sie bestimmte CAN-IDs filtern möchten, können Sie dies unter „CAN-Trace Filter“ tun.
9. Zum Beenden klicken Sie **Stop Tracing**.

### Die Argumente des CGI-Skripts

Im Folgenden werden die beim CGI-Skript „can-trace“ möglichen Argumente zur Durchführung einer HTTP-Anfrage beschrieben.

- **cmd=[start|stop|status]**

Der auszuführende Befehl – dieses Argument ist zwingend erforderlich.

- **cmd=start**

Startet den CAN Trace – weitere Argumente müssen angegeben werden.

- **cmd=stop**

Hält den CAN Trace an – weitere Argumente sind nicht erforderlich.

- **cmd=status**

Gibt den aktuellen Status des CAN Trace als Text aus

- **format=[html|text]**

Das Ausgabeformat – möglich sind „HTML“ (Default) oder „Text“. Die Angabe dieses Arguments ist optional.

- **ipaddr=a.b.c.d**

Die IP-Adresse (in Dezimalzahlen-Punkt-Darstellung), an die die UDP-Messages gesendet werden sollen. Wird kein Wert spezifiziert, so wird die IP-Adresse verwendet, von der der Aufruf stammt.

- **port=n**

Der UDP-Port, an den die UDP-Messages gesendet werden sollen. Dieser Port muss vom UDP-Monitor (= der Gegenstelle) erfragt werden. Die Angabe eines Ports ist unbedingt erforderlich.

- **ctrlmask=n**

Die Bitmaske zur Auswahl der Boards (und der Controller darauf), deren Verkehr übermittelt werden soll. Jedem Controller auf jedem CAN Board ist dabei ein eindeutiges Bit zugeordnet, die Berechnung erfolgt nach der Formel:

$$1 \ll (2 * \text{Boardnummer} + \text{Controllernummer})$$

wobei „Boardnummer“ Werte zwischen 0 und n annehmen kann, „Controllernummer“ 0 oder 1.

Die folgende Tabelle illustriert noch einmal die Zuordnung zwischen Board und Controller und den jeweiligen 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

Die Breite des Registers beträgt 32 Bit, unterstützt werden momentan 8 CAN Boards (= 16 Controller).

Der Wert für „ctrlmask“ muss als Dezimalzahl oder als Hexadezimalzahl (beginnend mit „0x“) angegeben werden. Wird kein Wert spezifiziert, so wird der Verkehr aller Controller aufgezeichnet.

**Beispiel:** Sollen der erste und zweite Controller des dritten Boards überwacht werden, so werden Bit 4 und Bit 5 des Registers gesetzt - der Wert der Bitmaske ergibt sich dann zu  $2^4 + 2^5 = 16 + 32 = 48$  (0x00030).

## Beschreibung der Antwort

Die CAN-Messages werden in den UDP-Frames als reiner ASCII-Code übertragen, jede einzelne Message wird durch das Zeichen für einen Zeilenvorschub \n (ASCII: 0x0A) abgeschlossen.

Eine typische Ausgabe sieht wie folgt aus:

```
....
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
....
```

Die Bedeutung der Elemente einer Antwortzeile ist wie folgt:

- **Zeilennummer**

Anhand der fortlaufenden Zeilennummerierung kann die Vollständigkeit der Aufzeichnung (= keine Streamingfehler, siehe „E.n“ weiter unten) verifiziert werden.

- **Zwei Zeitstempel**

Der erste Wert in der Zeile gibt die Zeit an, zur der die CAN-Message vom Simulationsmodell (auf dem Real-Time PC) erfasst wurde (in Sekunden).

Der zweite Wert ist ein Hardware-Zeitstempel des CAN-Boards und stellt die Empfangszeit einer Receive-Message dar. Bei Send-Message entfällt dieser Wert natürlich.



- **E:n**  
„E:0“ bedeutet: Kein Fehler.  
„E:1“ zeigt einen Streamingfehler durch zu hohe Last auf dem CANBus an, der in einem teilweisen Datenverlust führen kann.
- **CAN:x,y**  
enthält die Nummer des CAN-Boards x (0...n) und die Nummer des Controllers y auf diesen (0,1).
- **RX/TX**  
beschreibt die Richtung einer Message aus Sicht des Real-Time PC (TX = Send-Message, RX = Receive-Message).
- **ID: nn**  
Der Hex-Identifizier der CAN-Message
- **DLC: n**  
Der Data Length Code der CAN-Message
- **DATA: XX XX XX XX XX XX XX XX**  
Die 8 Daten-Bytes der CAN-Message (in Hex-Darstellung)

## 4.5 Schrittweise Ausführung der Simulation

---

Die Funktion „Model Step“ ermöglicht das Anhalten der automatischen Simulation und die anschließende Ausführung in Einzelschritten (in Form von Tasks und Prozessen).

Gehen Sie dazu wie folgt vor:

1. Starten Sie die Simulation.

- Wählen Sie im Web-Interface den Link **Model Step**.  
Die Seite „Model Step“ wird angezeigt.



**ETAS RTPC**  
Run your models in real time on a PC

**ETAS**

[Main page >> Model Step](#)

Acquisition Task: T3,Acquisition

```

T1,Init
  P0,CAN_Init
  P1,lcrt_Start_IdleController
  P2,Es113x_InitCode_Init_HWCF
  P3,Es1651sch_InitCode_Init_HWCF
  P4,Pb1651adcl_InitCode_Init_HWCF

T2,Exit
  P5,Pb1651adcl_ExitCode_Exit_HWCL
  P6,Es1651sch_ExitCode_Exit_HWCL
  P7,CAN_Exit
  P8,Es113x_ExitCode_Exit_HWCL
  P9,lcrt_Terminate_IdleController

T3,Acquisition
T4,Config
  P10,Pb1651adcl_ConfigCode_Config_HWCF

T5,Manager
T6,Analyse
T7,Acknowledge
T8,TaskDVEModel
  P11,CAN_Receive_board0_controller0
  P12,CAN_Receive_board0_controller1
  P13,Es113x_SyncReceiveCode_TaskDVEModel_HWCF
  P14,Pb1651adclAnnIn_TaskDVEModel_HWCF
  P15,lcrt_OneStep_Outputs_IdleController
  P16,Pb1651asclieAdrv_TaskDVEModel_HWCL
  P17,Es113x_SyncSendCode_TaskDVEModel_HWCF
  P18,lcrt_OneStep_States_IdleController
T9,Module_IdleCtrl_CAN_Send
  P19,CAN_Send
  
```

**Sequence**

The sequence may be entered as comma separated list of actions.  
Possible actions: Tx - Run task x, Px - Run process x, Rx - Realtimeest x, Hx - Pluginhook x.  
Lists can be specified: Tx,y - Run tasks x,y, Tx\*N - Run task x four times.

### Automatische Simulation anhalten



- Um die Simulation anzuhalten, klicken Sie das Symbol **Pause**.  
Die automatische Ausführung der Simulation wird gestoppt.

### Acquisition Task für die Darstellung in LABCAR-EE wählen

- Wenn Sie die schrittweise Ausführung der Simulation in einem oder mehreren Instrumenten verfolgen, wählen Sie unter „Acquisition Task“ diejenige Task, die auch für die Aktualisierung der Instrumente verwendet wird. Standardmäßig ist hier „Acquisition“ gewählt.

Unabhängig vom manuell durchgeführten Modellschritt wird danach immer die Acquisition Task ausgeführt und damit die Anzeigeelemente in der Experimentierumgebung aktualisiert.

### Einzelschritt ausführen (Task oder Prozess)

1. Klicken Sie in der Liste eine Task (oder einen Prozess), die Sie ausführen möchten.

```

T5,Manager
T6,Analyze
T7,Acknowledge
T8,TaskDVEModel
  P11,CAN_Receive_board0_controller0
  P12,CAN_Receive_board0_controller1
  P13,Es113x_SyncReceiveCode_TaskDVEModel_HWCF
  P14,Pb1651adclAnaIn_TaskDVEModel_HWCF
  P15,lcrt_OneStep_Outputs_IdleController
  P16,Pb1651adclLedDrv_TaskDVEModel_HWCL

```

Die Task (der Prozess) wird ausgeführt.

Die zuletzt ausgeführte Task (Prozess) wird gelb hinterlegt.

```

T5,Manager
T6,Analyze
T7,Acknowledge
T8,TaskDVEModel
  P11,CAN_Receive_board0_controller0
  P12,CAN_Receive_board0_controller1
  P13,Es113x_SyncReceiveCode_TaskDVEModel_HW

```

### Abfolge von Tasks oder Prozessen ausführen

1. Sie können auch eine Abfolge von Tasks oder Prozessen ausführen, die Sie in das Feld „Sequence“ eingeben.

#### Sequence

	Run Sequence
<small>The sequence may be entered as comma separated list of actions.          Possible actions: Tx - Run task x, Px - Run process x, Rx - Realtime test x, Hx - Pluginhook x.          Lists can be specified: Tx-y - Run tasks x-y, Tx*4 - Run task x four times.</small>	

Einzelne Bestandteile der Sequenz werden durch Kommas getrennt.

Bestandteile einer Sequenz sind:

- Tx: Führe Task x aus.
- Px: Führe Prozess x aus.
- Tx-y: Führe Task x bis Task y aus.
- Tx\*4: Führe Task x viermal hintereinander aus.

2. Zum Ausführen der Sequenz klicken Sie **Run Sequence**.

Die Sequenz wird ausgeführt.

### Automatische Simulation fortführen



1. Um die automatische Ausführung der Simulation fortzusetzen, klicken Sie das Symbol **Continue**.

Die Simulation wird wiederaufgenommen.

## 4.6 Real-Time Plugins (RT-Plugins)

RT-Plugins werden dazu verwendet, Code zur Laufzeit des Simulationsmodells hinzu zu laden – damit hat der Anwender die Möglichkeit, Echtzeitmodelle dynamisch zu erweitern.

In diesem Abschnitt finden Sie Informationen zu:

- "Vorbereiten des LABCAR-OPERATOR-Projekts" auf Seite 36

- "Die Plugin-Dateien" auf Seite 37
- "Spezielle Präprozessor-Anweisungen" auf Seite 38
- "Templates und Beispiele" auf Seite 39
- "Vorgehensweise" auf Seite 39

#### 4.6.1 Vorbereiten des LABCAR-OPERATOR-Projekts

Das Projekt muss einmalig vorbereitet werden, dass es mit RT-Plugins arbeiten kann, indem Hooks an passende Stellen in der Prozessliste eingefügt werden. Diese Hooks definieren den Aufrufkontext für die Ausführung von dynamisch geladenem Code. Spezifiziert werden diese Hooks in einem C-Code-Modul.

##### Hooks API

Alle Funktionen werden in der Headerdatei `rtos_realtime_test.h` definiert, den Sie mit

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

##### Typen:

```
rtos_handle_t
```

Typ eines generischen Handles zur Adressierung der Hooks

Beispiel:

```
static rtos_handle_t rttest_hook;
```

##### Init Code

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

Diese Funktion erzeugt einen „Named Hook“ und gibt ein Handle zurück.

Beispiel:

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

##### Exit Code

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

Diese Funktion entfernt einen zuvor erstellten Hook.

Beispiel:

```
realtime_test_hook_delete(rttest_hook);
```

##### Real-Time Code

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

Diese Funktion ruft die dem Hook zugewiesenen Plugin-Funktionen auf – der Hook definiert dabei den Kontext und die genaue Position des Aufrufs. Diese Funktion sollte in eine passende periodische Task eingehängt werden.

Beispiel:

```
realtime_test_hook_execute(rttest_hook);
```

## 4.6.2 Die Plugin-Dateien

Die zu ladenden Plugins werden auf dem Bedien-PC in Form von C-Code- und Header-Dateien spezifiziert. Alle diese Dateien müssen mit einem geeigneten Werkzeug zu LABCAR-RTPC übertragen werden (siehe "Vorbereitungen zum Herunterladen der Plugins" auf Seite 39).

### Grundlegende Plugin-Initialisierung

Für die grundlegende Initialisierung eines Plugins auf dem Real-Time PC müssen vier Funktionen implementiert werden:

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

Die Funktionen `on_load` und `on_initialize` werden beim Laden des Plugins aufgerufen – sie müssen im Erfolgsfall den Wert 0 zurückgeben. Die Funktionen `on_terminate` und `on_unload` werden beim Entladen des Plugins aufgerufen.

Die Funktion `on_initialize` wird üblicherweise zum Aufruf von API-Funktionen verwendet, die Plugin-Funktionen zu existierenden Hooks zuweisen.

Dazu muss ein statisches Interface-Objekt vom Typ „I\_realtime\_test“ implementiert werden – dieses Objekt enthält die Zeiger zu den Callback-Funktionen, die im Echtzeitkontext aufgerufen werden.

Die Interface-Definition von „I\_realtime\_test“ lautet:

```
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 ns);
    void (*background)(realtime_test_obj *obj);
}
I_realtime_test;
```

`init` wird bei Initialisierung des Realtime-Testobjekts aufgerufen, `exit` bei Beendigung des Realtime-Testobjekts.

`execute` wird vom Hook im Echtzeitkontext aufgerufen und `background` wird periodisch (etwa alle 0,5 s) aufgerufen.

### Erzeugung des Real-Time-Testobjekts

Zur Erzeugung des Real-Time-Testobjekts wird ein Interface-Objekt (Typ `I_realtime_test`) verwendet. Dabei können (neben einem Zeiger auf das Interface-Objekt) auch ein (optionaler) Name (`*name`) und ein benutzerdefinierter Zeiger (`*data`) verwendet werden.

Die Funktion

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

wird zur Erzeugung eines Real-Time-Testobjekts verwendet.

### Zuordnung des Real-Time-Testobjekts zu einem Hook

Ein Real-Time-Testobjekt kann durch den Aufruf von

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

einem vorhandenen Hook zugewiesen werden.

### Lösen des Real-Time-Testobjekts von einem Hook

Um ein Real-Time-Testobjekt von einem Hook zu lösen, verwenden Sie die Funktion

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

### Löschen eines Real-Time-Testobjekts

Die Funktion

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

entfernt ein existierendes Real-Time-Testobjekt.

## 4.6.3 Spezielle Präprozessor-Anweisungen

Für die Plugin-Dateien sind eine Reihe von zusätzlichen Präprozessor-Anweisungen verfügbar, die die Interaktion mit der laufenden Echtzeitanwendung erleichtern sollen.

Folgende Anweisungen stehen zur Verfügung:

#### #label

- #label name path  
Erzeugt ein Makro „name“, das den Zugriff auf das durch „path“ spezifizierte Element ermöglicht.

#### #labelstruct

- #labelstruct name path  
Erzeugt ein Makro „name“, das den Basistyp (ohne „.xxx“ am Ende) des durch „path“ spezifizierten Elements repräsentiert.

#### #labeltype

- #labeltype name path  
Erzeugt ein Makro „name“, das den (skalaren) C-Basistyp des durch „path“ spezifizierten Elements repräsentiert.

### #a2addr

- #a2laddr name path  
Erzeugt ein Makro „name“, das die A2L-Pseudo-Adresse als String enthält. Dieser String kann direkt mit der Funktion `rtpc_a2l_addr_value_ptr()` verwendet werden, die in `a2l_address_resolution.h` definiert ist.

## 4.6.4 Templates und Beispiele

Eine C-Datei mit einem Beispiel zur Implementierung von Hooks (`PluginHook.c`) finden Sie auf Ihrem Real-Time PC unter `http://192.168.40.14/tools/rtplugin/templates/`.

Dort finden Sie auch ein Template für ein Plugin (`plugin-template.c`).

## 4.6.5 Vorgehensweise

### Vorbereitungen zum Herunterladen der Plugins

Zum Herunterladen der Plugins auf den Real-Time PC wird steht ein Tool zur Verfügung, das Sie in folgendem Verzeichnis finden:

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

Es enthält zwei DLLs und zwei Executables:

- `EE.RealTimePlugin.Core.Interfaces.dll`
- `EE.RealTimePlugin.Core.dll`
- `RTPC-Plugin-GUI.exe` (Version mit GUI)
- `RTPC-Plugin.exe` (Kommandozeilenversion)

Um dieses Tool verwenden zu können, müssen Sie die beiden DLL und eines der Executables in ein Verzeichnis auf Ihrem Bedien-PC kopieren.

### Hooks in Projekt integrieren (LABCAR-IP)

Die unter "Templates und Beispiele" auf Seite 39 genannte Datei `PluginHook.c` ermöglicht beispielhaft die Erstellung eines Hooks.

1. Erstellen Sie dazu in LABCAR-IP ein neues C-Code-Model.
2. Fügen Sie den entsprechenden Code aus dieser Datei ein.
3. Fügen Sie einen entsprechenden Prozess hinzu.
4. Generieren Sie den Code erneut (**Project** → **Build**).
5. Wechseln Sie ins Register „OS Configuration“ und weisen Sie den neuen Prozess der gewünschten Task zu.

### Hooks in COSYM Projekt integrieren

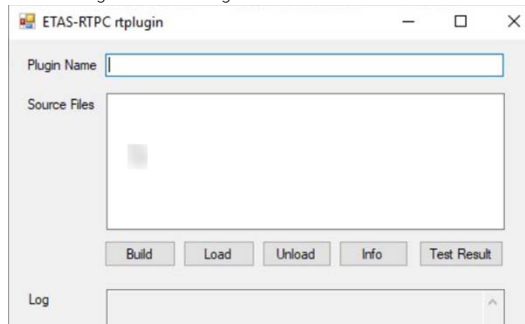
Die unter "Templates und Beispiele" auf Seite 39 genannte Datei `PluginHook.c` ermöglicht beispielhaft die Erstellung eines Hooks.

1. Erstellen Sie dazu in COSYM ein neues C-Code-Model.
2. Fügen Sie den entsprechenden Code hinzu.
3. Fügen Sie einen entsprechenden Prozess hinzu.
4. Öffnen Sie ein System.
5. Fügen Sie das C-Code-Model zum System hinzu.

6. Wechseln Sie ins Register „Deployment“ und weisen Sie den neuen Prozess der gewünschten Task zu.
7. Generieren Sie den Code erneut (**Project** → **Build**).

#### Plugin ausführen (LABCAR-EE)

1. Öffnen Sie Ihr Projekt in der Experimentierumgebung und führen Sie es aus.
2. Starten Sie die in "Vorbereitungen zum Herunterladen der Plugins" auf Seite 39 genannte Datei `RTPC-Plugin-GUI.exe`. Das folgende GUI wird geöffnet.



3. Geben Sie unter „Plugin Name“ einen Namen für die nachfolgende Liste von Quelldateien ein.
4. Geben Sie unter „Source Files“ die Quelldateien an, die zu diesem Plugin gehören sollen.
5. Alternativ können Sie Dateien auch per Drag & Drop in dieses Feld ziehen.
6. Wählen Sie **Build**, um die Dateien zum Real-Time PC zu laden und dort Code zu generieren.
7. Wählen Sie **Load**, um das Plugin zum laufenden Experiment hinzuzuladen.
8. Wählen Sie **Unload**, um die Ausführung des Plugins zu beenden.

## 4.7 CPU Core Load

Die Verteilung der Last von Prozessen auf die Prozessorkerne kann mittels `http://192.168.40.14/cgi-bin/cpu-load[?detail=n]` in eine XML-Datei ausgegeben werden.

Je nach Detaillierungsgrad enthält die XML-Datei unterschiedliche Informationen:



- `detail=0` (default)  
Hier wird lediglich die CPU-Last (in %) ausgegeben:  

```
<?xml version="1.0" encoding="UTF-8" ?>
- <load>
  <cpuload cpu="0" load="73.9" loadunit="%" />
</load>
```
- `detail=1`  
Es wird die Gesamtlast und die Verteilung auf die einzelnen Simulationstasks ausgegeben.
- `detail=2`  
Zusätzlich werden alle Echtzeitprozesse ausgegeben.
- `detail=3`  
Zusätzlich werden alle Nicht-Echtzeitprozesse ausgegeben.
- `detail=4`  
Zusätzlich werden alle Befehle ausgegeben.

## 4.8 Benchmark Indicator

---

Um die Simulationsleistung von ETAS RTPC auf unterschiedlichen PCs zu vergleichen, kann eine standardisierte Rechenroutine verwendet werden.

Schließen Sie die Headerdatei (in `/opt/etas/include`)

```
rtpc-benchmark.h
```

in ein beliebiges C-Code-Modul ein und führen Sie eine der folgenden Routinen aus:

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

Die erste Routine berechnet 2000 Variablen mit 8000 Anweisungen, die zweite 50 Variablen mit 400 Anweisungen – Argumente und Rückgabewerte sind ohne Bedeutung.

## 4.9 Testen der Hardwareperformance mit „cyclicttest“

---

Mit der Anwendung „cyclicttest“ kann geprüft werden, wie gut eine PC-Hardware für Echtzeitanwendungen geeignet ist. Hierbei wird die sogenannte Interrupt-Latenz zwischen dem Aktivieren des Timers und dem Ausführen des zugehörigen Codes gemessen.

### Cyclicttest aufrufen

1. Wählen Sie im Webinterface **Main Page >> System Info** und scrollen Sie zum Punkt **Cycle Test - Test Program**.

## 2. Klicken Sie **Cyclictest Control / Results**.

Auf der folgenden Seite kann das Programm gestartet werden. Darunter werden die Ergebnisse bereits durchgeführter Tests dargestellt



[Main Page](#) >> [System Info](#) >> [Cyclictest](#)

Duration [s]:

## 3. Wählen Sie die Testdauer („Duration“) und klicken Sie **Start Cyclictest**.

### Beispiel:

Ein Timer wird periodisch alle 100 ms gefeuert und anschließend wird gemessen, wann die zugehörige Software-Routine tatsächlich ausgeführt wird: Der Zeitstempel vom Start der Routine wird mit dem Zeitstempel verglichen, bei dem der Timer gefeuert wurde.

#### 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: 599991 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: 34
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
```

Die ausgegebenen Daten haben folgende Bedeutung:

- **T**  
Cyclictest startet pro Kern  $n$  einen periodischen Task  $T_n$ , d.h. Task T0 läuft auf Core 0, Task T1 auf Core 1 usw. Der Zahlenwert in Klammern liefert die Prozess-ID des jeweiligen Tasks.
- **P**  
Die Priorität des Tasks (hier: fest 99)
- **I**  
Das Aufrufintervall des Tasks in  $\mu\text{s}$  (hier: beginnend mit 100  $\mu\text{s}$ , größer werdend in Schritten von 10  $\mu\text{s}$ )
- **C**  
Die Anzahl der Taskaufrufe während des ausgeführten Tests

- **Min, Act, Avg, Max**

Die Werte sind die Latenzen zum Timer-Feuer-Zeitpunkt in  $\mu\text{s}$ . „Act“ ist der aktuelle Wert - zudem wird eine Statistik (Min/Avg/Max) erstellt.

Ein Wert für „Max“ von 15 bedeutet, dass es 15  $\mu\text{s}$  gedauert hat, bis die Software tatsächlich auf ein Timer-Ereignis reagiert hat. Bei einem „Max“ von 45  $\mu\text{s}$  ist es folglich kaum möglich, mit diesem PC einen 50  $\mu\text{s}$ -Task zu rechnen.

Die Max-Werte werden (bei erhöhten Latenzen) durch Farben hervorgehoben:

- Max  $\leq$  30  $\mu\text{s}$ : Keine Farbe
- 30  $\mu\text{s}$  < Max  $\leq$  50  $\mu\text{s}$ : Gelb.
- Max > 50  $\mu\text{s}$ : Rot

## 4.10 CAN/CANFD- und LIN-Abstraktion

---

### 4.10.1 Allgemeine Funktionsweise

Soll ein Projekt auf LABCAR-Systemen mit unterschiedlich ausgebauten CAN/CANFD- und LIN Schnittstellen eingesetzt werden, ohne dass die CAN/CANFD- und LIN-Konfiguration geändert werden muss, kann das Feature der CAN/CANFD- und LIN-Abstraktion (allg. Busabstraktion) verwendet werden. Dieses Feature wird in Verbindung des RTPC mit dem CAN- und LIN-Editor bzw. dem Netzwerkmodul eingesetzt.

Wesentlich für die Busabstraktion ist die Zuordnung von Kombinationen aus Board-ID und Controller-ID zu Kanälen. Diese Zuordnung wird auf dem RTPC in Form von ini-Dateien abgebildet. Diese Dateien sind in Sektionen unterteilt. Die Sektionen enthalten Paare aus einer Variablenbezeichnung und eines Wertes, welche durch das Gleichheitszeichen verbunden sind.

Bei der Busabstraktion beschreibt die Sektion den Bustypen sowie die Nummer des Kanals. Gültige Bezeichner für Bustypen sind `can`, `canfd` und `lin`. Kanäle werden durch beliebige natürliche Zahlen definiert. Die Reihenfolge der Sektionen in einer ini-Datei ist beliebig. Kombinationen aus Board-ID und Controller-ID werden durch die Variablen `boardid` und `controllerid` innerhalb der zugehörigen Sektion gebildet. Die Werte dieser Variablen sind natürliche Zahlen.

#### Beispiele

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

```
[canfd12]
boardid=1
controllerid=0
```

```
[lin4]
```

```
boardid=0  
controllerid=0
```

Das erste Beispiel beschreibt die Zuordnung des zweiten CAN-Controllers, des ersten Boards zum Kanal 0. Das Board mit der ID 1 im zweiten Beispiel ist CANFD-fähig. Aus diesem Grund ist dem ersten Controller auf dem CANFD Kanal 12 zugeordnet. Der erste LIN-Controller des ersten Boards wird als LIN-Kanal 4 bezeichnet.



#### INFO

Die Nummerierung der LIN-Controller beginnt bei jedem Board mit "0" unabhängig davon, wie viele CAN Controller auf dem Board vorhanden sind.


Die Busabstraktion wird ausschließlich auf dem RTPC ausgeführt. Dabei wird der durch die CAN- und LIN-Editoren bzw. das Netzwerkmodul erzeugte Code vor dem Build-Vorgang modifiziert. Dabei werden die in den Editoren bzw. dem Netzwerkmodul konfigurierten Kombinationen aus Board-ID und Controller-ID den entsprechenden Kanälen zugeordnet. Das gebaute Projekt ist dann neutral bezüglich der CAN/CANFD- und LIN-Schnittstellen, da es intern die Kanäle verwendet. Zur Laufzeit werden aus den Informationen über die Kanäle konkrete Schnittstellen über die entsprechende Kombination aus Board-ID und Controller-ID angesprochen.

### 4.10.2 Abstraktion der Busschnittstellen während des Buildvorgangs

Während des Buildvorgangs wird eine vorhandene Konfiguration aus CAN/CANFD- und LIN Schnittstellen in eine allgemeine Konfiguration überführt. Diese allgemeine Konfiguration kann dann auf unterschiedlichen LABCAR-Systemen ausgeführt werden, ohne dass das Projekt angepasst werden muss. Die Transformation erfolgt anhand einer Referenz-Konfiguration, welche auf dem RTPC, auf dem das Projekt gebaut wird, vorhanden sein muss. Die Referenzkonfiguration kann mit Hilfe des Web-Interfaces editiert werden. Alle in der Referenzkonfiguration aufgelisteten CAN/CANFD- und LIN-Schnittstellen werden beim Bauen des Projektes durch die jeweiligen Kanäle ersetzt.

Nicht vorhandene Kombinationen aus Board-ID und Controller-ID werden ignoriert und nicht durch Kanäle ersetzt.

Der Editor für die Referenzkonfiguration ist im Webinterface des RTPC unter **Configuration** → **CAN/LIN Abstraction Configuration** → **Expert Edit** erreichbar.



```

Edit File "/mnt/shared/config/ixcat/ixcat_reference.ini"

[can0]
board=0
controller=0

[can1]
board=0
controller=1

[can2]
board=1
controller=0

[can3]
board=1
controller=1

[canfd0]
board=0
controller=0

[canfd1]
board=0
controller=1

[canfd2]
4
  
```

Save Undo  
File: /mnt/shared/config/ixcat/ixcat\_reference.ini  
Size: 172

© 2003-2017 ETAS GmbH

Im CAN- und LIN-Editor sowie im Netzwerkmodul werden die Busschnittstellen weiterhin mit der Board-ID und der Controller-ID beschrieben. Die Umwandlung in eine abstrakte Beschreibung mit Kanälen erfolgt beim Bauen des Projektes anhand der Referenzkonfiguration.

### 4.10.3 Umwandeln der Kanalbeschreibung in konkrete Busschnittstellen zur Laufzeit

Nach dem Download des Projektes zum Simulations-RTPC erfolgt die Rücktransformation der abstrakten CAN/CANFD- und LIN-Kanäle in die entsprechenden Kombinationen aus Board-ID und Controller-ID. Dafür wird die Laufzeitkonfiguration verwendet. Diese kann mit Hilfe des Web-Interfaces editiert werden.

Kombinationen aus Controller-ID und Board-ID, welche beim Bauen des Projektes nicht durch Kanäle ersetzt wurden, bleiben bestehen und werden zur Laufzeit direkt angesprochen.

Der Editor für die Laufzeitkonfiguration ist im Webinterface des RTPC unter **Configuration** → **CAN/LIN Abstraction Configuration** → **Expert Edit** erreichbar.

### Configuration >> Expert Edit



```

Edit File "/mnt/shared/config/ixxat/ixxat_runtime.ini"

[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

[lin0]
board=1
controller=2

Save Undo

```

#### 4.10.4 Vorlage für eine Kanalkonfiguration

Bei jedem Bootvorgang des RTPC wird eine Vorlage für die Konfigurationsdateien erzeugt. Diese Vorlage überschreibt die vorhandenen Konfigurationsdateien nicht. Sie kann als Übersicht über die installierte CAN/CANFD und LIN-Hardware verwendet werden.

Die Vorlage der Konfigurationsdatei kann im Webinterface unter **Configuration** → **CAN/LIN Abstraction Configuration** → **Download template** heruntergeladen werden.



#### Beispiel für eine Konfiguration

```

; 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
```

Das Beispiel zeigt eine Konfiguration mit je einer XC16, einer IB200 und einer IB600-Karte. Dabei ist die Variable `RTOS_IXXAT_CAN_DRIVER_ORDER` so gesetzt, dass die XC16-Karte vor der IB200-Karte und der IB600-Karte erkannt wird. Die Controller der IB600-Karte sind CAN-FD-fähig, können aber auch als Standard-CAN-Controller eingesetzt werden. Aus diesem Grund werden die entsprechenden Kombinationen aus Board-ID und Controller-ID sowohl als CAN-Kanäle (`can4` und `can5`) als auch als CANFD-Kanäle (`canfd0` und `canfd1`) aufgelistet.

## 5 Konfiguration

---

In diesem Kapitel werden die Einsatzmöglichkeiten und die Konfiguration von ETAS RTPC mit dem Web-Interface beschrieben.

### 5.1 Einsatzgebiete

---

Für ETAS RTPC gibt es drei Einsatzgebiete:

- A Model-in-the-Loop (MiL)
- B Hardware-in-the-Loop (HiL)
- C Hardware-in-the-Loop mit Mess- und Verstellzugang (M&C)

Je nach Einsatzgebiet muss im Web-Interface unter **Main Page** → **Configuration** (siehe "Configuration" auf Seite 53) der Parameter „RTPC\_USAGE\_ETH1“ unterschiedlich gesetzt werden.

#### 5.1.1 Model-in-the-Loop (MiL)

Bei MiL-Tests besteht die Unit-under-Test aus einem in Software geschriebenen Modell einer Controllerfunktionalität. Es wird keine I/O-Hardware benötigt.

Folgende Einstellung ist vorzunehmen:

Parameter	Einstellung
RTPC_USAGE_ETH1	Off
...	...
RTPC_USAGE_ETHn	Off

**Tab. 5-1** Einstellung für MiL-Einsatz

#### 5.1.2 Hardware-in-the-Loop (HiL)

HiL-Tests sind der Standard-Anwendungsfall für ETAS RTPC. Im HiL-Betrieb ist die Unit-under-Test als Hardware vorhanden. Sie wird über I/O-Karten an das Test-System angeschlossen.

Der Real-Time PC steuert die I/O-Hardware über eine Ethernet-Verbindung an. Hier wird normalerweise der Anschluss Eth1 (HW) des Real-Time PC verwendet.

Folgende Einstellung ist vorzunehmen:

Parameter	Einstellung
RTPC_USAGE_ETH1	ES1130

**Tab. 5-2** Einstellung für HiL-Einsatz

#### 5.1.3 Hardware-in-the-Loop mit Mess- und Verstellzugang (M&C)

Bei der Einsatzart „HiL mit Mess- und Verstellzugang“ ist es möglich, während des Testlaufs simultan über serielle Protokolle (z.B. ETK oder XCP) auf die UuT zuzugreifen.

Der Mess- und Verstellzugang kann dabei mittels eines kompakten ES590 ECU-Schnittstellenmoduls realisiert werden. Die ES590 und der Real-Time PC werden über ein ES600 Netzwerkmodul direkt an den Bedien-PC angeschlossen.



Folgende Einstellung ist vorzunehmen:

Parameter	Einstellung
RTPC_USAGE_ETH1	ES1130

**Tab. 5-3** Einstellungen für HiL-Einsatz mit Messen und Verstellen

## 5.1.4 Hardware-in-the-Loop mit Last- und Fehlersimulation

Wenn ein ES4408-System zur Lastsimulation oder ein ES4440 Compact Failure Simulation Module zur Fehlersimulation eingesetzt wird, so findet die Kommunikation dieser mit dem Real-Time PC per Real-Time UDP (rtudp) statt.

Für die jeweiligen Ethernetanschlüsse am Real-Time-PC sind dann folgende Einstellungen vorzunehmen.:

Parameter	Einstellung
RTPC_USAGE_ETHn	rtudp_x (x = 0..4)

**Tab. 5-4** Einstellungen für Systeme zur Last- oder Fehlersimulation

## 5.1.5 Massive I/O

Unter „Massive I/O“ wird verstanden, dass das HiL-System mit einer hohen Anzahl von I/O-Kanälen versehen werden kann. Bei ETAS RTPC V6.5.2 bedeutet dies, dass bis zu acht VME/VXI I/O-Systeme (ES4100, ES4105, ES4300) unterstützt werden.

Jedes System wird von einem System Controller Board gesteuert. Um Echtzeit-Datenübertragung zu gewährleisten, gibt es zwischen Real-Time PC und jedem I/O-System eine Punkt-zu-Punkt-Verbindung. Das DVE-Modell hat damit Zugriff auf alle I/O-Karten, unabhängig davon, in welchem System sich die jeweilige Karte befindet.

Folgende Einstellungen sind vorzunehmen:

Parameter	Einstellung
RTPC_USAGE_ETHn	ES1130, ES1130_1, ES1130_2 oder ES1130_3

**Tab. 5-5** Einstellungen bei Einsatz mehrerer I/O-Systeme (Massive I/O)

## 5.2 Das Web-Interface

Über das ETAS RTPC-Web-Interface können Sie

- ETAS RTPC konfigurieren
- Software-Updates installieren
- Status-Meldungen und System-Informationen anzeigen

In diesem Abschnitt werden der Zugang zum Web-Interface, die Navigation in diesem und dessen Funktionen beschrieben.

### 5.2.1 Zugang zum Web-Interface

Das Web-Interface von ETAS RTPC V6.5.2 können Sie über die URL

`http://192.168.40.14`

erreichen. Starten Sie einfach einen Web-Browser auf dem Bedien-PC und navigieren Sie zu dieser Adresse.

Statt der IP-Adresse 192.168.40.14 können Sie auch den Alias „rtpc“ verwenden, wenn Sie diesen in der Windows-Systemdatei `hosts` deklarieren. Diese Datei finden Sie im folgenden Verzeichnis:

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

Fügen Sie zu den bereits bestehenden Einträgen die folgende Zeile hinzu:

```
192.168.40.14    rtpc
```

Damit können Sie das Web-Interface von ETAS RTPC über die URL `http://rtpc` erreichen.

## 5.2.2 Navigation

Die Navigationszeile direkt unter der blauen Kopfzeile hilft Ihnen bei der Orientierung im Web-Interface (siehe Abb. 5-1).



**Abb. 5-1** Das Web-Interface bei laufendem Simulationscontroller

Außerdem kommen Sie (wenn der Simulationscontroller angehalten wird) über die Navigationszeile schnell zu übergeordneten Seiten, z.B. der Hauptseite („Main Page“) zurück.

In den Betriebsmodi „Simulation Controller“ und „Ethernet-Bridge“ können nicht alle Funktionen des Web-Interfaces aufgerufen werden. Nach Starten des Web-Interfaces werden Sie automatisch zu den entsprechenden Unterseiten geleitet.

Folgende Funktionen stehen bei laufendem Simulationscontroller zur Verfügung (siehe Abb. 5-1 auf Seite 50):

- **System Info**

Hier erhalten Sie Informationen zum System und finden Sie Log-Dateien. Nähere Informationen finden Sie im Abschnitt "System Info" auf Seite 59.

- **Runtime Trace**

Hier können Sie Runtime Traces starten, in denen Prozesse und Tasks auf dem Real-Time PC protokolliert werden. Nähere Informationen finden Sie im Abschnitt "Runtime Traces" auf Seite 27.

- **CAN Trace**

Die CAN Trace Funktionalität ermöglicht die Überwachung des CAN-Verkehrs von und zu den IXXAT CAN-Karten. Eine Beschreibung dieser Funktionalität finden Sie im Abschnitt "CAN Trace" auf Seite 29.

- **Model Step**

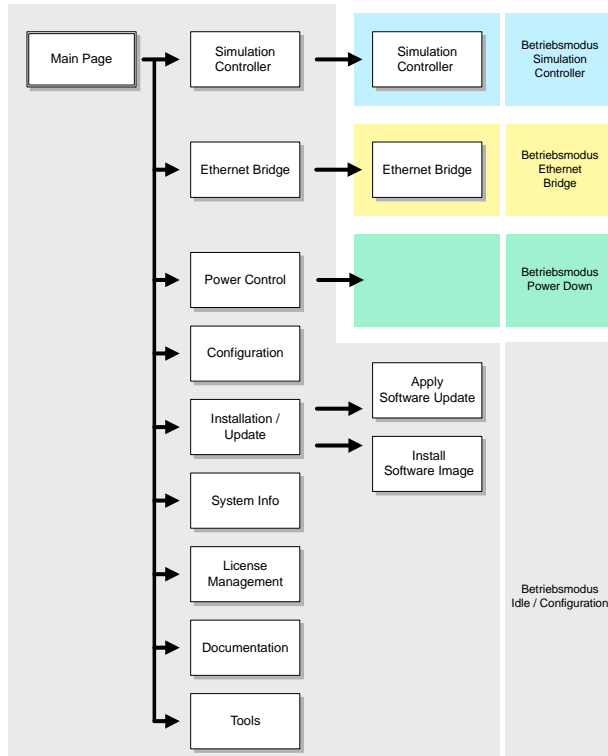
Diese Funktion ermöglicht eine Schritt-für-Schritt-Ausführung der Simulation. Nähere Informationen finden Sie im Abschnitt "Schrittweise Ausführung der Simulation" auf Seite 33.

- **Power Control**

Hier können Sie das System herunterfahren oder neu booten (siehe "Power Control" auf Seite 53).

### 5.2.3 Funktionen

Die folgende Abbildung zeigt eine Übersicht über die Struktur des ETAS RTPC Web-Interface und die Erreichbarkeit der verschiedenen Web-Seiten in den jeweiligen Betriebsmodi.



**Abb. 5-2** Struktur des Web-Interface

Die einzelnen Seiten sind im Folgenden näher beschrieben.

#### 5.2.3.1 Hauptseite (Main Page)

Über die Hauptseite lassen sich alle Funktionen des ETAS RTPC Web-Interface auswählen.

#### 5.2.3.2 Simulation Controller, Ethernet Bridge

Diese Menüpunkte dienen zum Starten der entsprechenden Betriebsmodi „Simulation Controller“ bzw. „Ethernet Bridge“. Die Betriebsmodi wurden bereits im Abschnitt „Betriebsmodi“ auf Seite 24 beschrieben.

Nach Aktivierung eines Betriebsmodus kann nur noch auf die entsprechenden Unterseiten zugegriffen werden.

### 5.2.3.3 Power Control

Falls der Ein/Aus-Taster am Real-Time PC nicht zugänglich oder nicht vorhanden ist, kann das Herunterfahren oder ein Reboot von ETAS RTPC auch über diesen Link des Web-Interface ausgelöst werden.

Um ETAS RTPC über das Netzwerk zu starten („Wake-on-LAN“), benötigen Sie auf dem jeweiligen Netzwerk-PC ein entsprechendes Tool, das die sogenannten „Magic Packets“ sendet (z.B. das freie Tool „WOL - Magic Packet Sender 2007“). Die notwendigen Daten für die Konfiguration dieses Tool finden Sie unter „Wake On LAN Settings“.

### 5.2.3.4 Configuration

Dies ist der Konfigurationsdialog von ETAS RTPC, der aus vier Teilen besteht:

- **Configuration Support**
  - **Automatic ES1130 Configuration**

Konfiguriert die Ethernet-Verbindung(en) (Eth1...Ethx) automatisch. Dabei wird die Echtzeit-Ethernetverbindung überprüft und - wenn eine ES1130 vorhanden und angeschlossen ist - der Parameter „RTPC\_USAGE\_ETH1“ auf den Wert „es1130“ gesetzt.



#### INFO

Der Parameter „RTPC\_USAGE\_ETHn“ wird nur dann angezeigt, wenn die jeweilige Ethernet-Schnittstelle auch konfiguriert wurde.

Eine Beschreibung der Autokonfiguration finden Sie in „Konfiguration durchführen“ auf Seite 17.

- **ES1130 Detection**  
Überprüft, ob die Verbindung zur ES1130 funktioniert.
- **RTPC Configuration**

Der Abschnitt „RTPC Configuration“ dient zum Einstellen des Konfigurationsmodus und zum Aktivieren von Debug-Optionen.

Unter „**Host Ethernet Configuration (ETH0)**“ wird der Bedien-PC konfiguriert, der am Ethernetadapter „ETH0“ angeschlossen ist.

- **IP Address**  
Hier kann die IP-Adresse (IPv4) des Ethernet-Adapters vorgegeben werden, der für die Verbindung zum Host-PC dient (Voreinstellung: 192.168.40.14).  
Es wird dringend empfohlen, eine IP-Adresse für ein privates Class-C-Netzwerk zu verwenden (Adresse: 192.168.x.y, Netmask: 255.255.255.0).
- **Netmask**  
Die Netmask für ETH0
- **DHCP**  
Hier können Sie festlegen, ob die IP-Adresse von ETH0 via DHCP bezogen werden soll oder nicht.

#### – Ethernet Negotiate

Hier können Sie festlegen, ob die beteiligten Controller selbständig die maximal mögliche Übertragungsgeschwindigkeit und das Duplex-Verfahren miteinander aushandeln und konfigurieren können („auto“) oder bestimmte Werte fest vorgeben.

Unter **„Realtime Ethernet Configuration“** werden weitere Echtzeit-Ethernetverbindungen konfiguriert.

#### INFO

Um die der Ethernetschnittstelle „Ethn“ zugeordnete Karte zu identifizieren, klicken Sie mit der Maus auf *Blink*.

#### – Usage

„ETH1“, „ETH2“ usw. können zum Anschluss einer ES1130 verwendet werden. Wenn ein ES1130 Simulation Controller Board angeschlossen wird, wählen Sie die Einstellung „es1130“.

Die Einstellungen „es1130\_n“ sind für weitere anschließbare ES1130 Simulation Controller Boards vorgesehen, die Einstellung „rtudp\_n“ ist für die Echtzeitkommunikation mit dafür geeigneter ETAS-Hardware (wie z.B. ES4440 Compact Failure Simulation Module oder das ES4408 Load Chassis) bestimmt. Details zur Real-Time UDP (rtudp) Driver API finden Sie im Web Interface unter <http://192.168.40.14/api/rtudp.html>. Des weiteren können hier einzelne Ethernetanschlüsse im Bridge-Modus (siehe Abschnitt 4.1.3 auf Seite 25) konfiguriert werden.

#### – IP Address

Hier können die IP-Adressen (IPv4) zusätzlicher Ethernet-Adapter (eth1, eth2, ...) vorgegeben werden, die den Real-Time PC mit weiteren Geräten verbinden (z.B. 192.168.100.23).

Es wird dringend empfohlen, eine IP-Adresse für ein privates Class-C-Netzwerk zu verwenden  
(Adresse: 192.168.x.y, Netmask: 255.255.255.0).

#### – Ethernet Negotiate

Hier können Sie festlegen, ob die beteiligten Controller selbständig die maximal mögliche Übertragungsgeschwindigkeit und das Duplex-Verfahren miteinander aushandeln und konfigurieren können („auto“) oder bestimmte Werte fest vorgeben.

#### INFO

Um die Benennung „ETHn“ für die vorhandenen Karten zu ändern, wählen Sie die Option „Show options to control...“ und ordnen Sie anschließend in der Spalte „Ethernet Order“ die gewünschte Karte zu (anhand der MAC-Adresse).

#### INFO

Ist für „Ethernet Order“ die Option „auto“ gewählt, so kann es bei einer Neuinstallation zur Änderung der Zuordnung „MAC-Adresse:ETHn“ kommen. Um die aktuelle Zuordnung zu fixieren, wählen Sie *Freeze current Ethernet Order*.

## Abschnitt „General Parameters“:

– **RTPC\_POWER\_UP\_MODE**

Dieser Parameter legt den Betriebsmodus fest, in dem sich ETAS RTPC nach dem Hochfahren befindet.

Folgenden Werte sind einstellbar:

**idle:** ETAS RTPC startet im Modus „idle“. Diese Modus wird zur Konfiguration und zum Systemupdate verwendet.

**simulate:** ETAS RTPC startet im Simulationsmodus. In diesem Modus kann ETAS RTPC nicht konfiguriert werden.

**bridge:** ETAS RTPC startet im Modus „Ethernet Bridge, d.h. die Ethernet-Schnittstellen von ETAS RTPC sind logisch verbunden. In diesem Modus kann ETAS RTPC nicht konfiguriert werden.

– **RTPC\_LOG\_LEVEL**

Beim Betrieb von ETAS RTPC werden Ereignisse protokolliert und in eine Datei geschrieben. Hier kann der Level für die zu protokollierenden Ereignisse festgelegt werden - „debug“ protokolliert alle Ereignisse, „emerg“ nur diejenigen, die das System unbrauchbar machen. Sinnvolle Einstellung für den Normalbetrieb sind „warning“ oder „error“.

– **RTPC\_NAME**

Hier kann für den jeweiligen Real-Time PC ein Name vergeben werden. Dieser Name kann hilfreich sein, wenn der Anwender mehrere ETAS RTPC-Systeme verwalten muss - er wird sowohl auf allen Seiten des Web-Interfaces als auch im Browser-Titel angezeigt.

– **RTPC\_TIMEZONE**

Zur Einstellung einer Zeitzone

– **RTPC\_PARALLEL\_MAKE**

Ermöglicht die parallele (d.h. auf mehreren CPU-Kernen gleichzeitige) Ausführung des GNU Make-Prozesses.

Mit „no“ wird das Feature deaktiviert und GNU Make wird nur für einen Job aufgerufen. Mit „auto“ wird das Feature aktiviert, wenn der Real-Time PC dies erlaubt. Bei der Einstellung „force“ wird dieses Feature immer verwendet.

– **RTPC\_TASK\_MONITOR\_0/1**

Der Taskmonitor ermöglicht eine Laufzeitüberwachung von zwei Tasks. Die zu überwachende Task kann hier über ihre Nummer ausgewählt werden.

Während die Task aktiv ist, wird das entsprechende Bit (0 oder 1) an der LPT-Schnittstelle auf „High“ gesetzt. Dies kann mit einem Oszilloskop beobachtet werden. Beachten Sie, dass der Einsatz des Taskmonitors einige Prozessorlast auf dem ETAS RTPC erzeugt. Um den Taskmonitor zu deaktivieren, setzen Sie die Werte auf „-1“.

Die folgende Tabelle enthält die Belegung der entsprechenden Anschlüsse des 25-poligen DSUB-Verbinders LPT.

Pin	Funktion
2	Datenbit 0
3	Datenbit 1
18 - 25	Masse

#### – RTPC\_TASK\_TIMING\_STATISTIC

Dieser Parameter definiert die Aufzeichnungsfrequenz (in Sekunden) für die Statistik der Tasklaufzeiten. Der Wert „0“ deaktiviert die Aufzeichnung.

Die Statistik beinhaltet folgende Informationen:

- Minimale/maximale Zeit dt (in ns) zwischen zwei Taskaktivierungen
- Minimale/durchschnittliche/maximale Laufzeit (in ns) der Task

#### INFO

Um störende Einflüsse auf das Echtzeitverhalten auszuschließen, erfolgt die Ausgabe der Statistik nicht im Echtzeitmodus. Außerdem sind die Messfunktionen nicht atomar implementiert. Es kann daher vorkommen, dass einzelne Ereignisse verlorengehen.

#### INFO

Die aufgezeichneten Ereignisse sind abhängig vom Wert des Parameters „RTPC\_LOG\_LEVEL“ (siehe Seite 55).

#### – RTPC\_INVALID\_CPU\_CORE\_HANDLING

Mit diesem Parameter wird die Behandlung einer ungültigen Spezifikation von CPU-Kernen (im Register „OS Configuration“ in LABCAR-IP) festgelegt. Mögliche Werte sind „warn“ (Default) und „stop“.

„warn“: Werden in der OS Configuration nicht-vorhandene Kerne spezifiziert, wird im Log-File von ETAS RTPC eine Warnung ausgegeben und alle Echtzeit-Tasks werden auf einem einzigen CPU-Kern ausgeführt.

„stop“: Im Falle einer ungültigen Spezifikation von CPU-Kernen wird das Echtzeit-Modell nicht starten und im Log-File von ETAS RTPC wird eine Fehlermeldung ausgegeben.

#### – RTOS\_TASK\_STACKSIZE

ETAS RTPC verwendet für jede laufende Task einen separaten Stack. Über diesen Parameter kann die Größe dieses Stacks (in kB = 1024 Byte) definiert werden. Ein sinnvoller Wert ist 1024 kB.

#### INFO

Wenn Modelle mit vielen Tasks verwendet werden, ist es aufgrund der limitierten Größe des RAM eventuell erforderlich, die Stackgröße zu verringern.

#### – RTOS\_TRACE\_BUFFER\_SIZE

ETAS RTPC besitzt eine automatische interne Ablaufprotokollierung, die beispielsweise bei Untersuchungen zur Performanz von Simulationsmodellen nützlich ist.

Mit diesem Parameter wird die Größe des Puffers (in der Einheit „Zahl der Ereignisse“) festgelegt, bevor die Daten in eine Protokolldatei (\*.rt) geschrieben werden. Diese Datei kann über die Seite **Main Page** → **Simulation Controller** → **Trace File Access** aufgerufen werden und anschließend mit RTA-TRACE dargestellt und analysiert werden.



#### – RTPC\_COMPILE\_OPTIMIZATION

ETAS RTPC verwendet den GNU C Compiler „gcc“ – die Optimierung der Codegenerierung kann in 4 Schritten eingestellt werden (0 ... 3), die den gcc Compileroptionen -O0 ... -O3 entsprechen. Je größer die Zahl, desto besser die Optimierung - auf der anderen Seite nimmt aber auch die Zeit für den Kompilervorgang zu.

Wenn Sie sich über diese Einstellung nicht sicher sind, wählen Sie „1“ oder „2“ für diese Einstellung.

#### – RTPC\_COMPILE\_LINK\_DEBUG

Wenn Sie diesen Parameter auf „yes“ setzen, werden die kompilierten Objektdateien und die ausführbare Datei mit Debuginformationen versehen. Dies kann bei der Fehlersuche im Simulationsmodell hilfreich sein (siehe **„Debug/Disassemble“** auf Seite 60).



#### INFO

Setzen Sie diese Option nur, wenn sie unbedingt erforderlich ist, da die ausführbare Datei dadurch sehr groß wird.

#### – RTPC\_COMPILE\_SWAPSPACE\_MB

Bei Verwendung großer Simulink-Modelle können die generierten C-Dateien sehr groß werden, was dazu führen kann, dass dem Compiler zu wenig Speicherplatz zur Verfügung steht.

Mit diesem Parameter können Sie für die Kompilationsphase zusätzlichen Speicherplatz zur Verfügung stellen. Dieser Speicher befindet sich in der Datei `compile.swapfile` und wird nur für die Kompilierung des Modells, nicht aber für die Simulation verwendet.

Mögliche Werte sind 0, 256, 512, 768 und 1024 - diese Zahlen geben die Größe der Swapdatei auf der Festplatte in Megabyte an. Wenn der Wert zu 0 gesetzt wird, dann wird eine zuvor angelegte Swapdatei wieder gelöscht. Die Voreinstellung ist 0 - wenn Sie beim Kompilieren keine Probleme haben, sollten Sie diese Einstellung beibehalten.

Die gewählten Einstellungen werden mit der Schaltfläche **Save RTPC Configuration** gespeichert. Mit der Schaltfläche **Discard Changes** können die wieder verworfen werden.

#### • Clock Synchronization

Klicken Sie **Synchronize**, um die Uhren des Real-Time PC und des Bedien-PC zu vergleichen und ggf. zu synchronisieren.

#### • System Configuration

Dieser Teil dient zur Konfiguration der PC-Hardware. Die Optionen werden beim ersten Booten von ETAS RTPC automatisch korrekt eingestellt. Eine Modifikation ist daher nicht notwendig.

Beim nachträglichen Einbau einer Ethernet-Schnittstelle muss die Option **„RTPC\_SYS\_ETHn“** (n = 1 ... 4) von „None“ auf „present“ umgestellt werden.

Durch Klicken der Schaltfläche **Save System Configuration** können Sie Ihre Auswahl speichern.

- **System Boot Settings**

ETAS RTPC speichert zwei vollständige Softwarestände auf getrennten Partitionen des Real-Time PC. Dies ermöglicht risikoloses Zurückkehren zu einer bekannten Softwareversion.

In der Auswahlbox können Sie auswählen, welcher Softwarestand beim nächsten Booten aktiviert werden soll. Mit der Schaltfläche **Set Boot Version** wird diese Auswahl gespeichert.

- **Reboot System**

Mit **Reboot System** startet ETAS RTPC mit dem ausgewählten Softwarestand neu.

### 5.2.3.5 Installation/Update

- **Apply Update**

Die aktuelle ETAS RTPC Version kann über das Web-Interface mit Software-Updates aktualisiert werden. Diese Software-Updates erhalten Sie bei Bedarf direkt von ETAS.

1. Klicken Sie unter „Upload Update File“ **Durchsuchen...**, um die Datei mit dem Update (Erweiterung \*.tgz) auszuwählen.
2. Wählen Sie **Upload**, um die Datei zum Real-Time PC zu übertragen.  
Die hochgeladene Datei ist jetzt unter „Select Update to Apply“ ausgewählt.
3. Falls mehrere Updates hochgeladen wurden, wählen Sie das entsprechende Update aus, welches aktiviert werden soll.
4. Unter „Select System to Update“ wählen Sie aus, welcher Softwarestand mit dem Update aktualisiert werden soll. Der gerade aktive Software-Stand ist gekennzeichnet („Active“).
5. Klicken Sie **Apply Update**.  
Unter „Applied Updates“ sind die bis dato durchgeführten Updates aufgelistet.  
Unter „Delete Installed Update Files“ können Sie nicht mehr benötigte Dateien wieder löschen.
6. Markieren Sie die zu löschenden Dateien und klicken Sie **Delete**.

- **Install New Version**

Mit der Funktion „Install New Version“ kann eine vollständige Neuinstallation durchgeführt werden. ETAS RTPC speichert alle bisher von CD installierten Software-Versionen auf der Festplatte des Real-Time PC. Diese Versionen können auf eine beliebige Partition zurückgeschrieben werden.

Für die Bereitstellung der Installationsdateien auf der Festplatte des Real-Time PC gibt es zwei Möglichkeiten:

- **Upload Image file**

Hiermit wird ein Image (auf ihrem lokalen Rechner oder auf einer Netzwerkressource) auf den Real-Time PC übertragen.

#### – Check CD/DVD-ROM disk

Hier wird das optische Laufwerk des Bedienrechners auf das Vorhandensein einer ETAS RTPC-Installationsdisk überprüft. Wird eine solche gefunden, wird die Image-Datei zum Real-Time PC übertragen. Die auf diese Weise bereitgestellten Images können jetzt unter „Select Software Image“ (s.u.) ausgewählt werden.

1. Wählen Sie unter „Install Partition“ aus, auf welche Partition neu installiert werden soll.  
Bis auf die aktive Partition kann dabei jede Partition beschrieben werden.
2. Wählen Sie unter „Select Software Image“ aus, welche Version installiert werden soll.
3. Wählen Sie über die Option „Use the existing configuration...“, ob die existierende Konfiguration von ETAS RTPC für diese Version übernommen wird oder Standardwerte verwendet werden.
4. Wenn Sie die Option „Skip implicit updates“ deaktiviert lassen, werden alle für die jeweilige Softwareversion verfügbaren Updates automatisch mit installiert.
5. Klicken Sie auf **Install**.

Die Auswahl der zu bootenden Version erfolgt in „Configure LABCAR-RTPC“ (siehe **“System Boot Settings“** auf Seite 58).

Nicht mehr benötigte Images auf der Festplatte des Real-Time PC können unter „Delete installed Software Images“ markiert und mit **Delete** gelöscht werden.

#### • Clone Installed Version

Sie können bereits installierte Versionen von ETAS RTPC klonen, indem Sie eine installierbare Datei (\*.tgz) erzeugen und diese auf Ihren Bedien-PC herunterladen. Eine ausführliche Beschreibung dieses Vorgangs finden Sie im Abschnitt “Klonen von installierten Versionen“ auf Seite 18.

### 5.2.3.6 System Info

Die Funktion „System Info“ zeigt verschiedene Informationen zum Status von ETAS RTPC. Im Fehlerfall kann diese Seite gespeichert werden und zu Analyse Zwecken verwendet werden.

#### • Log File messages

In diesem Fenster werden die aufgezeichneten Meldungen dargestellt. Zudem ist es möglich,

- sich die vollständige Datei anzeigen zu lassen (**View complete log file**),
- ein ZIP-Archiv zu erzeugen, das neben der Logdatei weitere Dateien mit Systeminformationen enthält (**Generate ZIP file report**) oder

- ein verschlüsseltes ZIP-Archiv (Kennwort: etas) zu erzeugen (**Encrypted ZIP**).



#### INFO

Da die ETAS Firewall nur verschlüsselte ZIP-Dateien passieren lässt, müssen Sie diese Option wählen, wenn Sie ein Reportarchiv per E-Mail an ETAS senden wollen.

- **Realtime Process Report**

Hier erhalten Sie ausführliche Informationen über laufende Echtzeit-Prozesse.

- **System Information**

In diesem Feld werden Systeminformationen des Real-Time PC dargestellt.

- **Cycticest - Test Program**

Hier kann das Testprogramm „cycticest“ aufgerufen werden - die Ergebnisse werden ebenfalls auf dieser Seite dargestellt. Eine genaue Beschreibung finden Sie im Abschnitt "Testen der Hardwareperformance mit „cycticest“" auf Seite 41.

- **CPU Performance Monitor**

Damit können Sie mit Hilfe der „CPU Performance Monitor“ Register eine Analyse der CPU-Performance durchführen.

- **LABCAR I/O Hardware Information**

Hier erhalten Sie detaillierte Informationen über die angeschlossene I/O-Hardware (VMEbus- und PCI/PCIe-Karten).

- **Applied Updates**

Informationen über durchgeführte Updates

- **Debug/Disassemble**

Wenn der Parameter „RTPC\_COMPILE\_LINK\_DEBUG“ (siehe Seite 57) gesetzt war, kann hier das Modell auf Fehler untersucht werden.

### 5.2.3.7 License Management

Hier werden die Lizenzen für ETAS RTPC und die FlexRay-Boards verwaltet.

- **ETAS RTPC Runtime Licenses**

- **License Information**

Zeigt die MAC-Adresse, an die die Lizenz gebunden ist, die Namen der aktiven Lizenzen und die Lizenzdatei selbst.



Hier können die Lizenzen für das FlexRay-Board von Elektrobit verwaltet werden.

Die Seriennummer der angeschlossenen Boards werden ermittelt – jedem Board kann dann eine Lizenzdatei zugeordnet werden.

[Main Page](#) >> [License Management](#) >> [Elektrobit License Manager](#)

#### Device List

```
INFO: =====
INFO: Searching for known PCI devices...
INFO:      Index 0: EB 5100 with Serial Number 249
```

#### Device License (Serial Number '249')

```
Target ready
Available licenses:
EB_tresoa_Buamirror_FlexRay (05/2011), 2009.a:OK, 2009.a.sr1:OK, 2010.a:OK, 3.3:OK, :
```

**Write License File to Elektrobit Device**

Serial Number

License File

**Read License File from Elektrobit Device**

Serial Number

1. Um eine Lizenzdatei vom Bedien-PC auf eine EB 5X00 zu transferieren, wählen Sie deren Seriennummer.
2. Wählen Sie die Datei mit **Durchsuchen** und klicken Sie **Write**.
3. Um eine Lizenzdatei von einer Karte auf den Bedien-PC zu transferieren, wählen Sie deren Seriennummer und klicken Sie **Read**.

### 5.2.3.8 Documentation

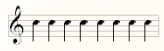
Der Link „Documentation“ führt Sie zu einem Verzeichnis, in dem Sie folgende Dokumente finden:

- Das Handbuch zu ETAS RTPC V6.5.2
- Die aktuellen Release Notes
- Eine Liste der von ETAS freigegebenen PCs
- Zwei Dokumente mit Informationen zu den BIOS-Einstellungen der PCs von ETAS

## 6 Fehlerbehebung

In diesem Kapitel finden Sie eine Beschreibung typischer Probleme und möglicher Lösungen dafür.

### A Nach dem Hochfahren von ETAS RTPC ertönt die Fehlertonfolge.



- Ursache:  
Der Simulationscontroller konnte nicht gestartet werden.
- Abhilfe:  
Siehe unter **"Der Simulationscontroller-Modus lässt sich nicht aktivieren."** auf Seite 65.

### B Das Web-Interface von ETAS RTPC ist nicht erreichbar.

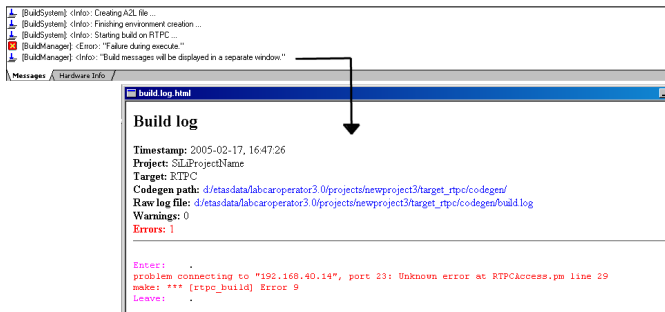
- Ursache:  
Ethernet-Verbindung ist nicht vorhanden, defekt oder Einstellungen sind fehlerhaft.
- Abhilfe 1:  
Führen Sie den Befehl `ping 192.168.40.14` aus. Falls Sie eine Antwort von ETAS RTPC erhalten, überprüfen Sie die Einstellungen Ihres Web-Browsers (z.B. Proxy-Einstellungen, siehe unten"Abhilfe 4:" auf Seite 63 ).
- Abhilfe 2:  
Falls Sie keine Antwort von ETAS RTPC erhalten, überprüfen Sie das Ethernet-Kabel. Stellen Sie insbesondere sicher, dass es sich um ein Crossover-Kabel handelt.
- Abhilfe 3:  
Wenn das Kabel in Ordnung ist, überprüfen Sie die Konfiguration des Ethernet-Interfaces (IP-Adresse: 192.168.40.240, Netzmaske: 255.255.255.0).
- Abhilfe 4:  
In Ihrer Internet-Konfiguration ist für das LAN ein Proxyserver konfiguriert. Um auf das Web-Interface von ETAS RTPC zugreifen zu können, muss die IP-Adresse des Real-Time PCs (192.168.40.14) in die Liste der Ausnahmen aufgenommen werden, die den Proxyserver nicht verwenden sollen.

#### Gehen Sie dazu wie folgt vor:

1. Wählen Sie im Startmenü **Einstellungen** → **Netzwerk und Internet** → **Internetoptionen**.  
Das Fenster „Eigenschaften von Internet“ wird geöffnet.
2. Wählen Sie das Register „Verbindungen“ und klicken Sie **LAN-Einstellungen**.  
Das Fenster „LAN-Einstellungen“ wird geöffnet.
3. Klicken Sie im Feld „Proxyserver“ **Erweitert**.  
Das Fenster „Proxysteinstellungen“ wird geöffnet.

4. Tragen Sie im Feld „Ausnahmen“ die IP-Adresse des Real-Time PCs ein (192.168.40.14).
5. Schließen Sie ein Fenster nach dem anderen mit **OK**.

### C Das Kompilieren von Modellen funktioniert nicht.



- Ursache:
  - Der Real-Time PC ist nicht mit dem Bedien-PC angeschlossen oder ausgeschaltet.
- Abhilfe:
  - Schließen Sie den Real-Time PC an oder schalten Sie ihn ein. Überprüfen Sie die Kommunikation durch den Aufruf des Web-Interfaces (<http://192.168.40.14> oder <http://rtpc>).

### D Der Projekt-Download („Open Experiment“) in LABCAR-OPERATOR schlägt fehl.

- Ursache 1:
  - Simulationscontroller wurde nicht gestartet.
- Abhilfe 1
  - Überprüfen Sie mit dem Web-Interface, ob der Simulationscontroller gestartet wurde. Aktivieren Sie gegebenenfalls den Simulationscontroller-Modus.
- Ursache 2
  - ES1130 Firmware ist nicht zu ETAS RTPC kompatibel.
- Abhilfe 2
  - Überprüfen Sie, ob die zu ETAS RTPC kompatible Firmware auf die ES1130 programmiert wurde. Schauen Sie dazu in der Konfigurationsseite.



**E Der Simulationscontroller-Modus lässt sich nicht aktivieren.**

- Ursache 1:  
ES1130 ist nicht angeschlossen oder ist ausgeschaltet. (HiL-Modus).
- Abhilfe 1:  
Überprüfen Sie, ob die ES1130 angeschlossen und eingeschaltet ist. Rufen Sie dazu die Funktion „ES1130 Detection“ unter „Configure ETAS RTPC“ auf. Im Fenster darunter sehen Sie das Ergebnis des Hardware-Scans. Es sollte die folgende Zeile zu sehen sein:  
`Ethernet Controller [eth1]: Connected to Target ES1130`
- Ursache 2:  
Die Einstellungen für RTPC\_USAGE\_ETH1 sind nicht korrekt.
- Abhilfe 2:  
Überprüfen Sie die Einstellungen für RTPC\_USAGE\_ETH1 im Konfigurationsdialog und berichtigen Sie diese (siehe „Einsatzgebiete“ auf Seite 48).
- Ursache 3:  
Bei der Ethernet-Schnittstelle liegt ein Interruptkonflikt vor.
- Abhilfe 3:  
Falls in der Konfigurationsseite des Web-Interface ein Interruptkonflikt gemeldet wird, stecken Sie die Karte in einen anderen Slot und schalten Sie im BIOS des PCs nicht benötigte Peripherie-Geräte (z.B. USB oder Sound-Unterstützung) ab.

## 7

## Glossar

---

In diesem Kapitel finden Sie die Erläuterung der in diesem Handbuch verwendeten Abkürzungen (in alphabetischer Reihenfolge).

**API**

Anwendungsprogrammierschnittstelle (Application Programmer's Interface)

**BIOS**

Basic Input Output System - die Firmware des PC

**CAN**

Controller Area Network (ISO 15765), ein serieller Bus-Standard zur Verbindung von Steuergeräten

**CPU**

Die Hauptprozessoreinheit des PC (Central Processing Unit)

**DSUB**

Bauform eines Steckersystems

**DVD**

Digital Versatile Disc

**DVE**

Fahrer-Fahrzeug-Umgebung (Driver-Vehicle-Environment)

**EVE**

ETAS Virtual ECU

**Eth0**

Bezeichnung des Ethernetports zum Anschließen des Host-PC

**Eth1**

Bezeichnung des Ethernetports zum Anschluss von Hardware wie z.B. ES1130 oder ES4440

**GB**

Gigabyte

**GNU**

Open-Source-Software Community

**HiL**

Hardware-in-the-Loop

**HTTP**

Hypertext Transfer Protocol

**ID**

Identifikationsschlüssel

**IDE**

Integrated Drive Electronics

**IP**

Internet Protocol

**INCA**

Mess-, Verstell- und Diagnosewerkzeug von ETAS

<b>ISO</b>	International Organization for Standardization
<b>LAN</b>	Local Area Network
<b>LPT</b>	Parallele Schnittstelle nach IEEE-1284
<b>MB</b>	Megabyte
<b>NTP</b>	Network Time Protocol
<b>PC</b>	Personalcomputer (Desktop oder Laptop)
<b>PCIe</b>	PCI Express (Peripheral Component Interconnect Express)
<b>PS/2</b>	Serielle Schnittstelle von PCs
<b>RAM</b>	Random Access Memory
<b>RTPC</b>	Real-Time PC
<b>UDP</b>	User Datagram Protocol – ein Netzwerkprotokoll
<b>URL</b>	Uniform Resource Locator
<b>VECU</b>	Virtual Electronic Control Unit
<b>VMEbus</b>	Bussystem
<b>VT-d</b>	Intel®'s Virtualization Technology for Directed I/O
<b>XP</b>	Ein Release von Microsoft Windows
<b>ZIP</b>	Dateiformat für komprimierte Archive

## 8 Kontaktinformationen

---

### ETAS Hauptsitz

ETAS GmbH

Borsigstraße 24

70469 Stuttgart

Deutschland

Telefon: +49 711 3423-0

Fax: +49 711 3423-2106

Internet: [www.etas.com](http://www.etas.com)

### ETAS Regionalgesellschaften und Technischer Support

Informationen zu Ihrem lokalen Vertrieb und zu Ihrem lokalen Technischen Support bzw. den Produkt-Hotlines finden Sie im Internet:

ETAS Regionalgesellschaften    Internet: [www.etas.com/de/contact.php](http://www.etas.com/de/contact.php)

ETAS Technischer Support    Internet: [www.etas.com/de/hotlines.php](http://www.etas.com/de/hotlines.php)

# Index

---

<b>A</b>	
Automatische Konfiguration	53
<b>B</b>	
Benchmark Indicator	41
Betriebsmodus	
Ethernet Bridge	25
Idle/Configuration	25
Power down	25
Simulation Controller	24
BIOS	
Einstellungen	12
Boot-Einstellungen	58
<b>C</b>	
CAN Trace	29
CAN/CANFD-Abstraktion	43
CAN-Board	22
CAN-FD	22
Clock Synchronization	21, 57
CPU Core Load	40
CPU Performance Monitor	60
cyclctest	41
<b>D</b>	
Datenschutzerklärung	8
<b>E</b>	
Ethernet Order	54
<b>F</b>	
Fehlerbehebung	63
FlexRay-Board	22
<b>H</b>	
Hardwareanforderungen	11
Hardware-in-the-Loop (HiL)	48
Hyper-Threading	12
<b>I</b>	
Installation	13
<b>K</b>	
Klonen	
installierter Versionen	59
Konfiguration	48
<b>L</b>	
LABCAR I/O Hardware	
Information	60
License Management	60
Lieferumfang	10
LIN	22
LIN-Abstraktion	43
Lizensierung	17
<b>M</b>	
Massive I/O	49
Model-in-the-Loop (MiL)	48
<b>R</b>	
Real-Time Plugins	35
Realtime Process Report	60
RTOS_TASK_STACKSIZE	56
RTOS_TRACE_BUFFERSIZE	56
RTPC_COMPILE_LINK_DEBUG	57
RTPC_COMPILE_OPTIMIZATION	57
RTPC_COMPILE_SWAPSPACE_MB	57
RTPC_INVALID_CPU_CORE_HANDLING	56
RTPC_LOG_LEVEL	55
RTPC_NAME	55
RTPC_POWER_UP_MODE	55
RTPC_TASK_MONITOR_0/1	55
RTPC_TASK_TIMING_STATISTIC	56
Runtime Trace	27
<b>S</b>	
Schrittweise Ausführung	
der Simulation	33
Simulation	
schrittweise ausführen	33
System Boot Settings	58
Systemkonfiguration	
HiL-Testsystem	16
Systemkonzept	7
Systemvoraussetzungen	10
<b>V</b>	
VMEbus-Reset	25
<b>W</b>	
Web-Interface	49
Funktionen	52
Navigation	50