

ASCET Rapid Prototyping V6.3

User's Guide



Copyright

The data in this document may not be altered or amended without special notification from ETAS GmbH. ETAS GmbH undertakes no further obligation in relation to this document. The software described in it can only be used if the customer is in possession of a general license agreement or single license. Using and copying is only allowed in concurrence with the specifications stipulated in the contract.

Under no circumstances may any part of this document be copied, reproduced, transmitted, stored in a retrieval system or translated into another language without the express written permission of ETAS GmbH.

© **Copyright 2014** ETAS GmbH, Stuttgart

The names and designations used in this document are trademarks or brands belonging to the respective owners.

Document EC012401 V6.3 R01 EN - 11.2014

Contents

1	Introduction	7
1.1	Safety Advice	7
1.1.1	Correct Use	7
1.1.2	Labeling of Safety Instructions	7
1.1.3	Demands on the Technical State of the Product	8
1.2	Installation	8
1.3	User Information	9
1.3.1	Documentation Structure	9
1.3.2	How to Use this Manual	9
1.3.3	ASCET-RP Online Help – Quick Guide	11
2	Configuring Experimental Targets	12
2.1	The Hardware Options	12
2.1.1	"Hardware Configuration" Subnode	13
2.1.2	"Signal Mapping" Subnode	13
2.1.3	"HWC-Paths" Subnode	13
2.1.4	"Legacy HWC Editor (deprecated)" Subnode	13
2.1.5	"Hardware Connection" Subnode	14
2.2	Hardware Connection with the ETAS Network Manager	14
2.3	Interface Setup Without ETAS Network Manager (ES1000 Only)	14
2.4	Selecting a Compiler	14
2.4.1	Using Your Own Compiler	15
2.4.2	Changing to the GNU Cross Compiler or QCC Compiler	16
2.5	Hints on Using ASCET-RP	18
2.5.1	Preprocessing Available Data Bases	18
2.5.2	Converting Projects for ES1000.1 to a Supported Target	18
2.5.3	Using dT	19

3	Rapid-Prototyping Experiments	20
3.1	Experimenting with ASCET	20
3.1.1	The User Interface	20
3.1.2	Running Online Experiments	22
3.1.3	Standalone Mode	27
3.2	Experimenting with INCA	28
3.3	Experimenting with INTECRIO	35
4	Hardware Configuration	36
4.1	Hardware Configurator	36
4.2	Migration of Existing Projects	40
4.3	HWC Editor (Legacy)	42
5	Hardware Systems	43
6	Legacy HWC Editor	44
7	The ETK Bypass	45
7.1	ETK Bypass: Definition	45
7.2	Hardware Configuration of an ETK Bypass	46
7.3	ASCET Project for the ETK Bypass	46
7.4	How the ETK Bypass Works	47
7.5	Data Exchange Between Control Unit and ETAS Experimental System	48
7.6	Initially Required Information and Data	56
8	HWC Items	58
9	ES1000 Tutorial	59
9.1	Tutorial – Experimenting with INTECRIO	61
9.2	Tutorial – ES1222 (CAN-IO)	62
9.2.1	The ES1222 Board	63
9.2.2	Sample Project.	64
9.2.3	Creating the Hardware Configuration	65
9.2.4	Configuring the CAN Channels	69
9.2.5	Saving the Hardware Configuration and Generating Code.	76
9.2.6	Experimenting with the Sample Project	77
9.3	Tutorial – ES1303	81
9.3.1	The ES1303 Hardware	81
9.3.2	Sample Project.	82
9.3.3	Creating the Hardware Configuration	82
9.3.4	Configuring the ES1303 Signal Groups and Signals	86
9.3.5	Saving the Hardware Configuration and Generating Code.	88
9.3.6	Final Actions	89
9.4	Tutorial – ES1325 (without Trigger)	90
9.4.1	The ES1325 Board	93
9.4.2	Sample Project.	94
9.4.3	Creating the Hardware Configuration	95
9.4.4	Configuring the Devices of the ES1325	99
9.4.5	Saving the Hardware Configuration and Generating Code.	108
9.4.6	Experimenting with the Sample Project	109
9.5	Tutorial – ES1325 (with Trigger)	115

9.5.1	The ES1325 Board	116
9.5.2	Sample Project.	116
9.5.3	Creating the Hardware Configuration	117
9.5.4	Configuring the Devices of the ES1325	120
9.5.5	Saving the Hardware Configuration and Generating Code.	125
9.5.6	Experimenting with the Sample Project	126
10	RTPRO-PC Tutorial	133
10.1	Preparations	133
10.1.1	RTPRO-PC Configuration	133
10.1.2	ASCET-RP Preparations	136
10.2	Adapting the Model	137
10.3	Creating the Hardware Configuration	138
10.4	Configuring the CAN Channels	141
10.5	Saving the Hardware Configuration and Generating Code	149
10.6	Experimenting with the Sample Project	150
11	Appendix A: Bypass Concept	153
11.1	ETK Bypass Concept Description	153
11.2	Bypass Input	153
11.3	Hook-Based Bypass	154
11.4	Service-Based Bypass	154
11.5	Safety Considerations	155
11.5.1	Bypass Input Data	155
11.5.2	Bypass Calculation.	155
11.5.3	Bypass Output Data.	156
11.5.4	Message Copies	156
11.6	Service-Based Bypass Specifics	157
11.6.1	Service Processes for the SBB Implemented as Service Functions	157
11.6.2	Controlling the ECU Behavior from ASCET-RP	158
11.6.3	Summary.	159
12	Appendix B: Compiler Switches and API Functions	160
13	Appendix C: ETAS Network Manager	161
13.1	Overview	161
13.2	ETAS Hardware Addressing	161
13.3	Network Adapter Addressing	162
13.3.1	Type of Network Adapter Addressing	162
13.3.2	Addressing the Network Adapter Manually	162
13.3.3	Addressing the Network Adapter via DHCP	163
13.4	User Interface	163
13.4.1	"Network settings for ETAS hardware (Page 1)" Dialog Window	163
13.4.2	"Network settings for ETAS hardware (Page 2)" Dialog Window	165
13.4.3	"Network settings for ETAS hardware (Page 4)" Dialog Window	166
13.5	Configuring Network Addresses for ETAS Hardware	166

- 13.5.1 Adapter with Fixed IP Address 166
- 13.5.2 Adapter in DHCP Environment. 168
- 13.6 Troubleshooting Ethernet Hardware Access 170
- 14 Appendix D: Troubleshooting General Problems 171
 - 14.1 Problems and Solutions 171
 - 14.1.1 Network Adapter cannot be selected via Network Manager. 171
 - 14.1.2 Search for Ethernet Hardware fails. 172
 - 14.1.3 Personal Firewall blocks Communication 174
- 15 ETAS Contact Addresses 179
- Index 181

1 Introduction

The execution of real-time software requires experimenting hardware that is capable of real-time processing. The ASCET **R**apid **P**rototyping V6.3 (ASCET-RP V6.3) software package is used to integrate the ES1000, ES900, and RTPRO-PC experimental targets (E-Targets) in ASCET V6.3. Together with I/O periphery, powerful development systems can be built on the basis of these experimental targets.

In addition to the license for the compiler toolset used, the ASCET-RP Package also includes extensions to the ASCET development environment, such as for the homogenous integration of compiler and linker calls and the ERCOS^{EK} and RTA-OSEK operating system kernels for experimental targets.

ASCET-RP contains, in addition, the basic functions of the ASCET base software. ASCET-MD is needed in addition for the modeling functions.

1.1 Safety Advice

Please adhere to the Product Liability Disclaimer (ETAS Safety Advice) and to the following safety instructions to avoid injury to yourself and others as well as damage to the device.

1.1.1 Correct Use

ETAS GmbH cannot be made liable for damage which is caused by incorrect use and not adhering to the safety instructions.

1.1.2 Labeling of Safety Instructions

The safety instructions contained in this manual are shown with the standard danger symbol shown below:



The following safety instructions are used. They provide extremely important information. Read this information carefully.



WARNING

Indicates a possible medium-risk danger which could lead to serious or even fatal injuries if not avoided.



CAUTION

Indicates a low-risk danger which could result in minor or less serious injury or damage if not avoided.

NOTICE

Indicates behavior which could result in damage to property.

1.1.3 Demands on the Technical State of the Product

The following special requirements are made to ensure safe operation:

- Take all information on environmental conditions into consideration before setup and operation (see the documentation of your computer, hardware, etc.).

**WARNING**

Wrongly initialized NVRAM variables can lead to unpredictable behavior of a vehicle or a test bench, and thus to safety-critical situations.

ASCET projects that use the NVRAM possibilities of ASCET-RP targets expect a **user-defined** INIT process that checks whether all NV variables are valid for the current project, both individually and in combination with other NV variables. If this is not the case, all NV variables have to be initialized with their (reasonable) default values.

Due to the NVRAM saving concept, this is **absolutely necessary** when projects are used in environments where any harm to people and equipment can happen when unsuitable initialization values are used (e.g. in-vehicle-use or at test benches).

Further safety advice is given in the ASCET V6.3 safety manual (`ASCET_SafetyManual.pdf`) available at ETAS upon request.

1.2 Installation

The ASCET-RP V6.3 installation is described in the ASCET installation guide (`ASCET_V6.3_Installation.pdf` on the installation disk).

Details on licensing are also given in the ASCET installation guide.

Components

The ASCET-RP V6.3 installation includes the following components:

- Integration of the ES1130 and ES1135¹;
- Integration of the ES910;
- Integration of the RTPRO-PC;
- Integration of the I/O boards, including ETK Bypass and CAN Bypass, for the targets ES1130 and ES1135;
- Integration of the I/O interfaces for the target ES910;

¹ The term *ES113x* is used throughout this manual for an arbitrary ES1000 system controller.

- Integration of the I/O interfaces for the target RTPRO-PC;
- GNU and QCC compilers;
- Documentation and examples.

Sample Files

After the installation of ASCET-RP V6.3, the sample databases exported from ASCET are located in the `EXPORT` directory of your ASCET installation in the `Tutorial RTIO.*1` and `Tutorial INTECRIO.*2` files.

1.3 User Information

1.3.1 Documentation Structure

The *ASCET-RP documentation* consists of a user's guide which describes general features, as well as the migration of existing projects that still use the legacy HWC editor to the Hardware Configurator, and an online help which describes the operation of the Hardware Configurator.

The ASCET-RP V6.3 user's guide consists of the following main sections:

- General Section (chapters 1, 2, 3)
- Hardware Configuration (including Bypass Interface, chapters 4, 5, 6, 7, 8)
- Tutorial (chapters 9 and 10)
- Appendix (chapters 11, 12, 13, 14)

The general section is intended for all users of ASCET-RP V6.3. Here, the users find information about the structure, installation and usage of ASCET-RP V6.3.

The subsequent chapters introduce and explain the functionality and operation.

The tutorial contains lessons about experimenting with INTECRIO, as well as the configuration of several boards.

1.3.2 How to Use this Manual

The user's guide is available electronically and can be viewed on the screen at any time. Using the index, full-text search, and hypertext links, you can find references fast and conveniently (see `ASCET-RP V6.3 Manual.pdf`).

Documentation Conventions

Instructions are phrased in a task-oriented format as shown in the following example:

To reach a goal:

- [Execute operation 1.](#)
Explanations are given below an operation.
- [Execute operation 2.](#)
- [Execute operation 3.](#)

¹. * = `exp` (binary export format) or `ax1` (xml-based export format)

In this manual, an *action* is a sequence of operations that need to be executed in order to reach a certain goal. The title of an action usually expresses the result of the operations, such as "To create a new component" or "To rename an item". The action descriptions often include screenshots of the corresponding ASCET window or dialog box related to the action.

Typographic Conventions

The following typographic conventions are used throughout this manual:


Select File → Open . Click OK .	Menu options and button labels are shown in blue bold font.
Press <ENTER>.	Key commands are shown in angled brackets, in BLOCK CAPITALS.
The "Open File" dialog window opens.	Names of program windows, dialog windows, fields, etc. are enclosed in double quotes.
Select the <code>setup.exe</code> file.	Text in drop-down lists on the screen, program code, as well as path and file names are shown in the <code>Courier</code> font.
A <i>distribution</i> is always a one-dimensional table of sample points.	Content markings and newly introduced terms are shown in <i>italics</i> .
The OSEK group (see http://www.osekvd.org/) has developed certain standards.	Links to internet documents are set in <u>blue</u> font.

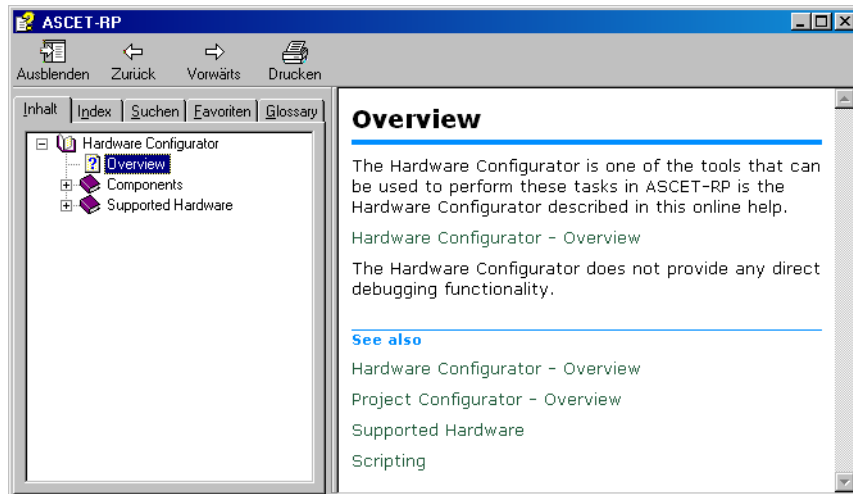
Important notes for the users are presented as follows:

Tip

Important note for the user.

1.3.3 ASCET-RP Online Help – Quick Guide

In the Hardware Configurator, use the **Help → Help** menu option or the  button to invoke the general help function. Press the <F1> function key to call context-sensitive help.



The tabs of the help window provide you with the following options:

- The "Contents" tab allows you to browse the help topics by categories.
- The "Index" tab lists all index entries. Browse the entire list, or enter a search term to limit the scope of listing.
- The "Search" tab allows you to search for individual words or terms included in a help topic. Type a search string and let the help function list the entries it has found related to this term.
- The "Favorites" tab allows you to bookmark topics.
- The "Glossary" tab contains special terms and their definitions.

2 Configuring Experimental Targets

ASCET Rapid Prototyping V6.3 (ASCET-RP V6.3) contains the compiler and linker tools required for producing executable files for a rapid prototyping target, and an extension of the ASCET development environment for the hardware integration. The target itself can be selected in the target options of the project. That way the targets are fully integrated into ASCET.

The configuration of the compiler and the linker, as well as the description of the interface to the actual target hardware is not described in ASCET directly, but either with the help of the ETAS Network Manager or with the help of *.ini files. Section 2.1 describes the hardware options of ASCET-RP, hardware connection using the ETAS Network Manager is described in section 2.2. Section 2.3 describes the configuration of the host interface via changes in target.ini, as well as the configuration of the ethernet interface.

Section 2.4 describes the selection of the compiler, and section 2.5 gives some hints on using ASCET-RP.

Structure of the E-Target Directories

Installing ASCET-RP V6.3 results in the creation of several e-target subdirectories in the ..\ASCET6.3\Target directory. These subdirectories contain E-target-specific information, configuration and library files.

The following table lists the subdirectories:

ASCET Subdirectory	E-Target	Computer Node
..\Target\ES1130	ES1000.2/ES1000.3 ^a	ES1130
..\Target\ES1135	ES1000.2/ES1000.3	ES1135
..\Target\ES910	ES910.2/ES910.3	ES910
..\Target\QNX86	RTPRO-PC	RTPRO-PC
..\Target\Prototyping	(determined in INTECRIO)	

a: The terms *ES1000* or *ES1000.x* are used in this manual, unless a particular target is meant.

The ...\Target\ES113x directory contains files used by both ES1000 simulation nodes, e.g. compiler-specific make files.

All makefiles and build scripts support paths with blanks.

- If a path containing blanks is to be used in a makefile, ASCET converts it to short Windows format (for example, c:\Documents and Settings would be converted to c:\DOCUME~1).
- If a path containing blanks is to be used in a batch file, ASCET generates it encapsulated in ", or converts it to short Windows format.

2.1 The Hardware Options

ASCET-RP adds the "Hardware" node and its subnodes to the ASCET options window. These nodes contain all tool options available for hardware configuration.

2.1.1 "Hardware Configuration" Subnode

In this node, you can specify the hardware configurator you want to use.

- **Use separately installed INTECRIO for hardware configuration**
 Activate this option if you want to use a separately installed version of INTECRIO as Hardware Configurator, instead of the one installed with ASCET-RP V6.3.
 A compatible INTECRIO installation, i.e. V4.3 or higher, is required for this purpose.
- **HW configurator path**
 Only available if **Use separately installed INTECRIO ...** is activated.
 Path to the INTECRIO installation you want to use as Hardware Configurator.
- **HWC working directory**
 Working directory for internal usage by the Hardware Configurator. Here, temporary files (e.g., *.hwx) are stored.

2.1.2 "Signal Mapping" Subnode

In this node, you can specify settings for automatic mapping.

- **Message Creation Target**
 This option determines whether ASCET messages generated during automatic mapping are created in the project itself (`Project`, default) or in one of the included modules (`Module`).
- **Module Instance Name**
 This option is only effective when "Message Creation Target" is set to `Module`.
 Enter the name of the module for the generated messages in the input field. You can use any module included directly or indirectly (in another module) in the project. For modules in modules, the module names are separated by a dot. If, e.g., module A contains module B, you enter `A.B` if you want to create the messages in module B.
 If the selected module is not included in the project, the following error message appears:

```
Can't add Message '<msg name>', because module
'<module name>' doesn't exist in project!
```

2.1.3 "HWC-Paths" Subnode

This node shows various paths. It is read-only.

2.1.4 "Legacy HWC Editor (deprecated)" Subnode

This node has been removed because the legacy HWC editor is no longer available.

2.1.5 "Hardware Connection" Subnode

In this node, you can specify options for the connection between ASCET-RP and the hardware. It is suggested that you select the item you use most frequently.

Tip

The options in this node are described in the ASCET online help.

2.2 Hardware Connection with the ETAS Network Manager

Hardware configuration with ETAS Network Manager is described in the ASCET online help, section "INTECRIO Connectivity / ASCET-RP".

The hardware selection window, named "Experimental Target Hardware Selection", is described in the same section of the ASCET online help.

2.3 Interface Setup Without ETAS Network Manager (ES1000 Only)

Interface setup without ETAS Network Manager is described in the ASCET online help, section "INTECRIO Connectivity / ASCET-RP".

2.4 Selecting a Compiler

The GNU Cross Compiler (GNU-C V3.4.4 (Power PC) in the user interface) is integrated in ASCET-RP for the ES1130 and ES1135 and Prototyping targets. The QCC compiler (QCC V6.5.0 in the user interface) is integrated in ASCET-RP for the ES910 and RTPRO-PC targets.

Target	Compiler
ES1130, ES1135	GNU Cross Compiler
Prototyping	GNU Cross Compiler
ES910	QCC compiler
RTPRO-PC	QCC compiler

Tab. 2-1 Target/Compiler Overview

For each compiler, version-specific make files are provided in the `..\Target\ES113x\trgmake` or `..\Target\ES910\trgmake` or `..\Target\QNx86\trgmake` directory. Version-independent settings are stored in files named `settings_<compiler>_common.mk`. In these files, you can set separate compiler options for module code, project code and init code. The following section in the make files is provided for that purpose:

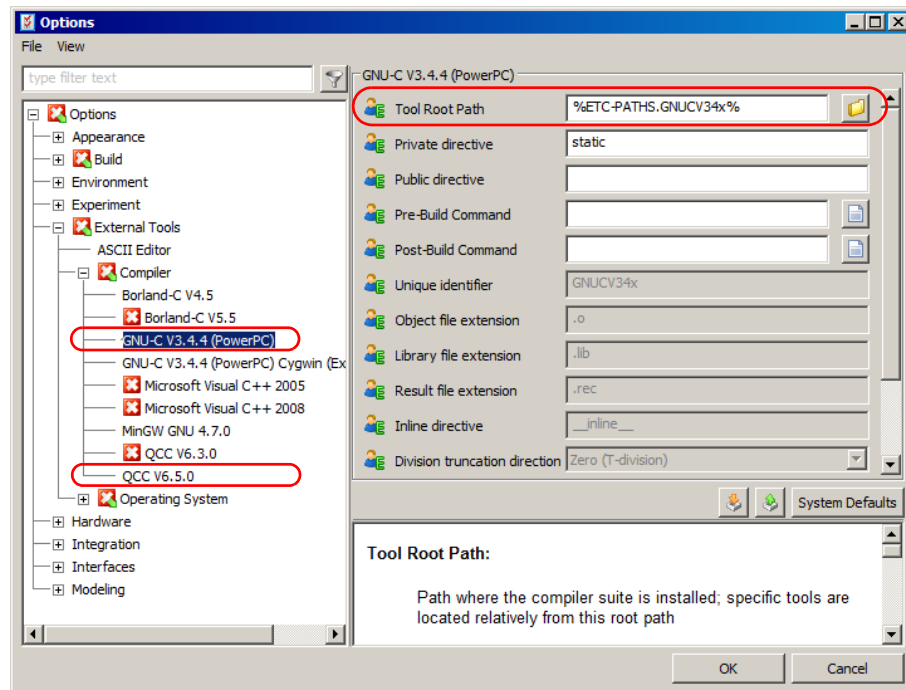
```
# Compilation settings for different lists of files
FILES_MODULES_INV = $(CC_INV)           ↵
                    #Add specific options here
FILES_PROJECT_INV = $(CC_INV)           ↵
                    #Add specific options here
FILES_INIT_INV    = $(CC_INV)           ↵
                    #Add specific options here
```

Precompiling C header files used by many C code files can significantly speed up the compilation process. The GNU Cross Compiler, V3.4 and higher, supports usage of precompiled headers. Only one precompiled header file can be used per compilation. Therefore, Project should be selected in the "Build" node of the project settings, to use precompiled headers as effectively as possible. Using pre-compiled headers is activated automatically (ASCET options window, "GNU-C V3.4.4 (PowerPC)" node, **Supports precompiled header** option).

2.4.1 Using Your Own Compiler

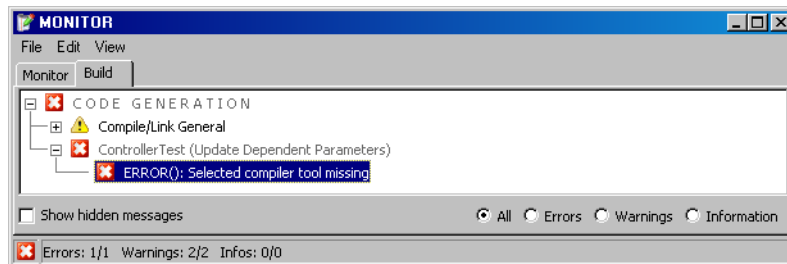
The MS-DOS version of the GNU Cross Compiler and the QCC compiler are supplied as a standard parts of the ASCET-RP V6.3 package.

If you want to use your own compiler, you have to reset the path of the GNU-C V3.4.4 (PowerPC) or QCC V6.5.0 compiler to the new compiler in the ASCET options dialog.

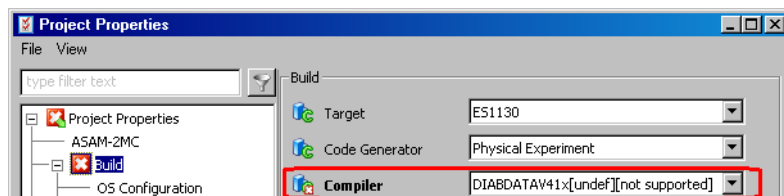


2.4.2 Changing to the GNU Cross Compiler or QCC Compiler

If you are working with an older project that uses the ES1130 target with the Diab Data compiler, *no* error message is shown when you open the project. Only the generation of executable code produces an error message in the ASCET monitor window.



When you double-click the error message, the "Project Properties" window opens in the "Build" node. The compiler is marked as undefined and not supported.



If you are working with an older project that uses the ES1112 target, the following error message is displayed when you open the project, and the missing target is replaced by the target PC.

The required Target >ES1112< is not available! Changed Target to >PC<!

In both cases, you have to select a suitable combination of target and compiler for the project. Proceed as follows.

To change to a suitable target/compiler combination:



- In the project editor, click on the **Project Properties** button.

The "Project Properties" window opens.

- In the "Build" node, select the target ES1130 or ES1135 or ES910 or RTPRO-PC.

A suitable compiler is selected automatically.

- Click **OK**.

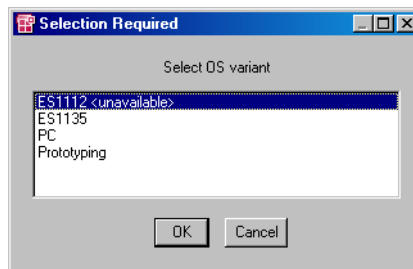
Depending on which target the project used, you now have to copy the C code, if your project uses C code blocks. This procedure is described in section "To copy operating system settings and C Code:" below.

- Finally, select **Build** → **Touch** → **Recursive** so that all components of the project are recompiled in the next run.

To copy operating system settings and C Code:

- Select the "OS" tab in the project editor.
- Select **Operating System** → **Copy From Target** from the project editor.

The "Selection Required" window opens.

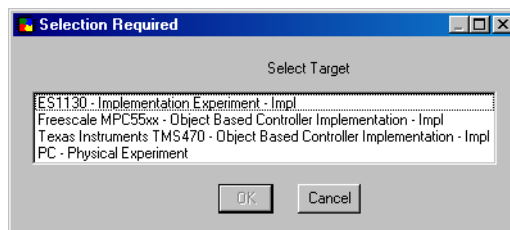


- From the "Selection Required" window, select the original target of the old project.
- Click **OK**.

The operating system code is copied from the old target to the ES1130 or ES1135 or ES910 or RTPRO-PC target.

- In the project editor, select **Extras** → **Copy C-Code From**.

The "Selection Required" window opens.



- In the "Selection Required" window, select the original target, experiment and implementation of the old project.
- Click **OK**.

The code is copied from the old target and experiment to the current settings.

2.5 Hints on Using ASCET-RP

2.5.1 Preprocessing Available Data Bases

ASCET databases which were created with ASCET versions prior to V4.x must be converted to ASCET V4.x before they can be opened and converted with ASCET V6.3.

Tip

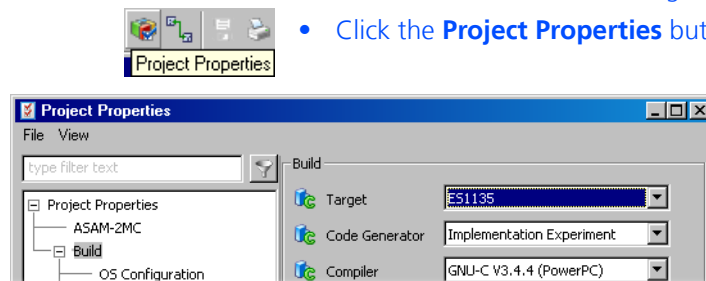
Detailed information on converting very old ASCET projects (with TIPExp V3.x and older, Target PPC) is given in the ASCET online help.

2.5.2 Converting Projects for ES1000.1 to a Supported Target

ASCET projects which were created for the ES1000.1 E-target have to be converted to a supported E-target, i.e. ES1000.2, ES1000.3, ES900 or RTPRO-PC. The following steps have to be executed for the conversion.

To convert an ASCET project for ES1000.1 to a supported target:

- Load the ASCET project.
A message is displayed that the ES1112 target is no longer available.
- Click **OK** to confirm the message.
The project opens; the target >PC< is selected instead of the unavailable target.
- Click the **Project Properties** button.



- In the "Project Properties" window, "Build" node, select the target ES1130 or ES1135 or ES910 or RTPRO-PC.
A suitable compiler is selected automatically.
- Click **OK** to close the "Project Properties" window.
- Copy the operating system settings and C code to the selected target, as described in "To copy operating system settings and C Code:" on page 17.

The project can now be edited for the ES1000.2/ ES1000.3, ES900 or RTPRO-PC system.

2.5.3 Using dT

The ERCOS^{EK} operating system is implemented for the ES113x target. An RTA-OSEK operating system is implemented for the ES910 and RTPRO-PC targets. Both operating systems enable access to the time dT which has elapsed between the last and second-last call of the running task. dT always refers to the task in which the variable is used (see Fig. 2-1).

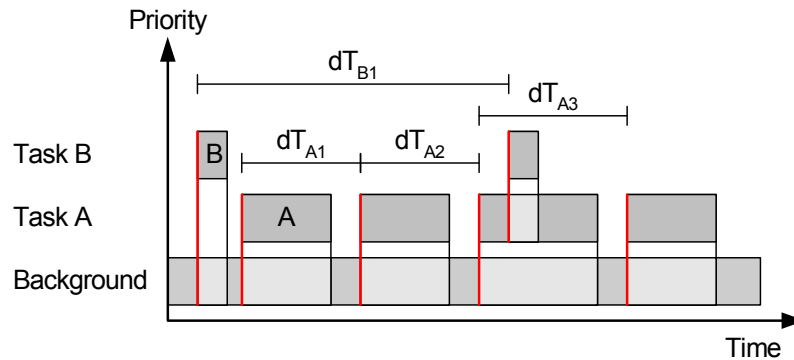


Fig. 2-1 dT Scheme

dT is a global `uint32` variable. It is declared in one of the ERCOS^{EK} or RTA-OSEK header files and contains the value for the current task in units of system ticks.



dT can be accessed from ASCET using the **dT** button in the editors. This enables you to create an element (`real64`) which contains the time in units of seconds.

If users do not generate this element in the C code editor, but still access dT , no error message appears because dT is declared in the ERCOS^{EK}/RTA-OSEK files. But as dT in ERCOS^{EK}/RTA-OSEK and dT in ASCET have different units (system ticks or seconds respectively), the calculations are incorrect. Users should therefore ensure that they generate the corresponding element with the **dT** button.

In the generated code for experimental targets, access to the global element dT is indirect: A C code macro is generated to access dT . The name of the macro is created as follows:

```
_PROJECTNAME_<IMPLEMENTATION>_dTAccess
```

The access macro for the project in chapter 9.2 "Tutorial – ES1222 (CAN-IO)", e.g., is named `_ES1222_PROJECT_IMPL_dTAccess`.

A default definition of the macro is generated by ASCET which can be replaced by a user-defined definition.

3 Rapid-Prototyping Experiments

This chapter describes the different possibilities of running a Rapid-Prototyping experiment.

Experiments that use an ES1000 target contain one memory page, experiments that use an ES910 or RTPRO-PC target contain two memory pages. To make use of both memory pages, you have to use INCA/INCA-EIP as experiment environment (see also section 3.2); the ASCET or INTECRIO experiment environment does not support multiple memory pages. See the INCA and INCA-EIP documentation for details on using memory pages.

3.1 Experimenting with ASCET

If you want to run the Rapid-Prototyping experiment in ASCET, you can choose between an online and an offline experiment in the project editor. For more details, please refer to the ASCET online help.

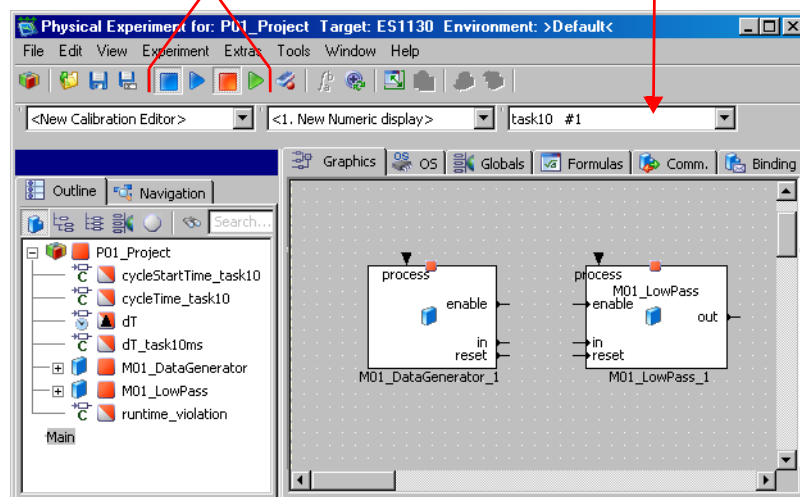
Only the special features of the online experiment are described here.

3.1.1 The User Interface

The user interface of the online experiment is very similar to that of the offline experiment. However, the buttons for controlling the experiment and the NVRAM cockpit (ES1135, ES910, and RTPRO-PC), the "Task" combo box and the functions in the **Experiment** and **Tools** menus are different from the offline experiment.

Control buttons (experiment)

"Task" combo box



Buttons



1. Exit to Component (ends the experiment and invokes the project editor)
2. Load Environment (loads an experiment environment, i.e. predefined measure and calibration windows with assigned variables)

3. Save Environment (saves the current experiment environment)
 4. Save Environment As (saves the current experiment environment under a freely definable name)
 5. Stop OS (stops the operating system and thus the experiment)
 6. Start OS (starts the operating system and thus the experiment)
 7. Stop Measurement (stops measurement, i.e. the data display)
 8. Start Measurement (starts measurement, i.e. the data display)
 9. Open Data Logger
 10. Open CT Solver (opens a window in which you can configure the integration method)
This button is *only* available if you are experimenting with a CT block or a hybrid project.
 11. Update Dependent Parameters (updates the values of dependent parameters)
 12. Expand / Collapse Window (shows/hides the component display)
 13. Always on top (keeps the experiment window on top)
This button is only available if the window is collapsed.
 14. Navigate down to selected component of element tree (shows the included component selected in the "Outline" tab.)
 15. Navigate up to parent component (shows the parent component)
- If you are experimenting with the ES1135, ES910 or RTPRO-PC target, the toolbar contains another button **Open NVRAM Cockpit**.

Experiment Menu

- *Stop OS (<F6>)*
Stops the operating system.
- *Start OS (<F7>)*
Starts the operating system.
- *Stop Measurement (<F8>)*
Stops the measurement.
- *Start Measurement (<F9>)*
Starts the measurement.

Tools Menu

- *Data Logger*
Opens the Data Logger.
- *Target Debugger*
Opens the debugger window for C code components.
- *NVRAM Cockpit*
(Only available when the target ES1135, ES910 or RTPRO-PC is selected in the code generation options.)
Opens the NVRAM Cockpit.

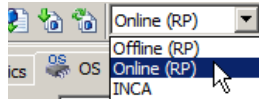
The other elements of the user interface correspond to those of the offline experiment; they are described in the "Experimentation" section of the ASCET online help.

3.1.2 Running Online Experiments

Start the online experiment environment for a project from the project editor.

To start the online experiment:

- Open the project or component you require.
- If you want to experiment with a component, open the relevant default project.
- In the project properties, "Build" node, select the target ES1130, ES1135, ES910 or RTPRO-PC and an appropriate compiler.
- In the "Experiment Target" combo box, select Online (RP).



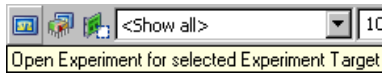
Offline (RP) is intended for offline experiments on the target, INCA and INTECRIO are intended for transfer to the respective tools.

The buttons **Open Experiment for selected Experiment Target** and **Reconnect to Experiment of selected Experiment Target** are now available.

- Select **Build** → **Experiment**

or

- click the **Open Experiment for selected Experiment Target** button.



To select hardware (with ETAS Network Manager):

Tip

If you are working without ETAS Network manager, skip this section and continue reading with section "What to do in case of an error:" on page 23.

When you activated the **Use ETAS Network Manager (enables 'Select Hardware')** option in the hardware options, the hardware selection window opens under certain conditions. The hardware options and the hardware selection window are described in the ASCET online help.

- In the "Select simulation board of type <type>" field, select the hardware you want to use.
The **OK** button becomes available.
- If required, perform other settings.
- Close the window with **OK**.

The system checks whether the selected hardware is available and agrees with the target you selected in the code generation options of the project. If this the case, the experiment environment opens.

What to do in case of an error:

If no agreement is found between selected and available hardware, you have to close the hardware selection window with **Cancel**. An error message opens:

Target connection check failed. No hardware selected for target <target>. Check target connection again before starting the build process?

- Click **Yes** to repeat the search for a hardware connection.

Or

- Click **No** to start the build process without hardware connection.

If you are using the ETAS Network Manager, the hardware selection window opens again after the Build process.

If you are not using the ETAS Network Manager, you are asked, after the Build process, whether you want to repeat the search for a suitable hardware or not.

Or

- Click **Cancel** to abort the experiment.

If no other environment is available, the default experiment environment for the component is opened immediately after starting the experiment (cf. page 22). If you are working without ETAS Network Manager, or after successful hardware selection (cf. page 22) if you are working with the ETAS Network Manager. Continue reading on page 24.

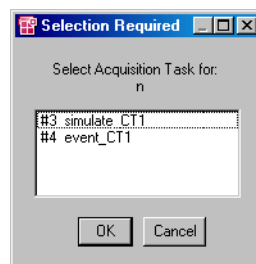
To open the experiment environment for the online experiment:

If several environments are available, the "Environment Browser" dialog window opens.

- In the "Environment Browser" dialog window, select the environment you want to use, and click **Load**.

For more details, refer to the section "Experimentation" in the ASCET online help.

If an existing experiment is opened, but a previous acquisition task is no longer available in the OS, you are prompted to select a replacement for each element that used the old acquisition task.



- In the "Selection Required" window, select one task and click **OK**.

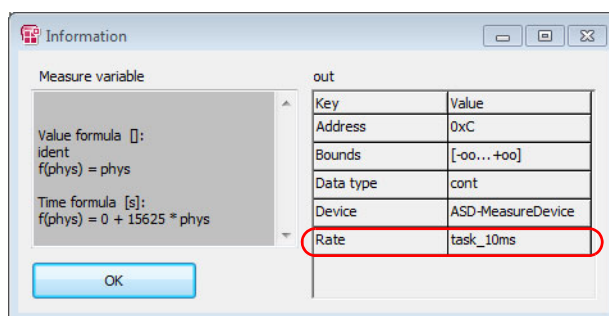
To assign a new element to a measure window:

- In the "Task" combo box, select the task you want to use as acquisition task for this element.
When the experiment environment is saved, the elements' acquisition tasks are referred to by name.
- In the "Outline" tab, select an element you want to measure.
You can select several elements at the same time. These are assigned the same acquisition task and measure window.
- Assign the selected element(s) to a measure window (see online help for details).
You can assign elements with different acquisition tasks to the same measure window.
If an element is assigned to several measure windows, all measure windows use the same acquisition task for this element.

To check an element's acquisition task:

You can check the acquisition task used by an element at a later time.

- Go to the measurement window that contains the element you want to check.
- Right-click the element and select **About Variable** from the context menu.
The "Information" window opens. The acquisition task name is shown in the "out" table on the right, row "Rate".

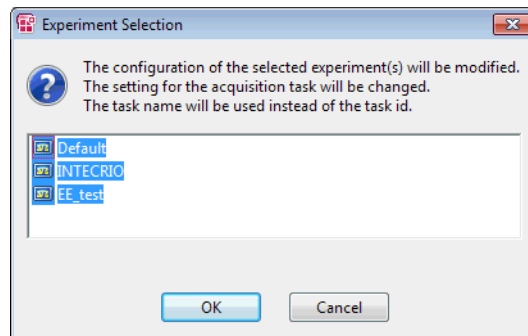


In ASCET-RP versions prior to V6.3, acquisition tasks were referenced by task number instead of task name. Experiments saved in ASCET-RP V6.2 or earlier are updated automatically when you open and re-save them in V6.3. Alternatively, you can use the **Extras** → **Bound acquisition task by name** menu option in the project editor to update all experiments saved for a given project.

To convert acquisition task references:

- In the project editor, select **Extras** → **Bound acquisition task by name**.

The "Experiment Selection" window opens. It lists all experiments saved for the project.



- Select the experiments you want to convert.
- Click **OK**.

The references to acquisition tasks are converted.

Setting up an online experiment only entails the setting up of the measure and calibration windows. The measure and calibration windows in the online experiment are the same as those in the offline experiment. The "Experimentation" section in the ASCET online help explains how to use them.

Once the experiment has been set up, you can start it. While the online experiment is running, you can modify the display options in all measure and calibration windows, open and close measure/calibration windows and modify data values with the calibration system.

To start an experiment and measurement:

- Open the experiment environment for the project you want to experiment with.
- Select **Experiment** → **Start OS**

or

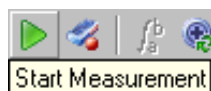


- click the **Start OS** button.

If you set the relevant options in the ASCET options window, "Experiment" node, variables and parameters are initialized each time the OS is started. The operating system and the experiment are started. Measure data is not displayed yet.

- Select **Experiment** → **Start Measurement**

or

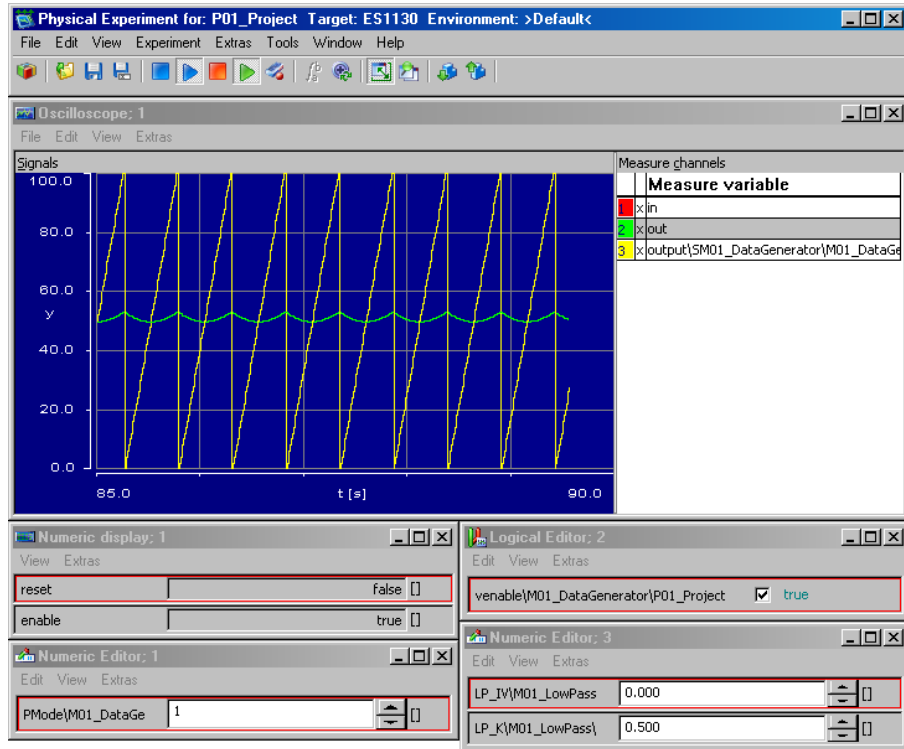


- click the **Start Measurement** button.

Tip

The measurement may affect the real-time behavior of the model.

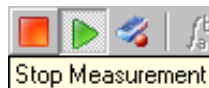
Measurement starts. All values set up in the measure system are shown in the respective windows.



To stop measurement:

- Select **Experiment** → **Stop Measurement**

or



- click the **Stop Measurement** button.

Measurement is stopped, but the experiment continues. When measurement is restarted, the time axis is set to the current value.

All settings remain active. The measure data is retained in the oscilloscope window; you can analyze the data.

To stop the experiment:

- Select **Experiment** → **Stop OS**

or



- click the **Stop OS** button.

Any measurement which is currently running is stopped. The operating system, and thus the experiment, is stopped and enters the inactive mode. The inactive mode of the operating system may contain one task with trigger mode `init` that is executed when the operating system is stopped. This can be used to reset external hardware, for instance.

When the operating system is restarted, it goes through the start mode and the corresponding `init` task again. There is no pause function for the operating system. If you set the relevant options in the ASCET options window, "Experiment" node, variables and parameters are initialized.



- Click the **Exit to Component** button to exit the experiment environment and return to the project editor.
- Select **File** → **Exit** to exit the experiment environment and close the project.

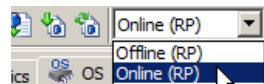
As with the offline experiment, you can take a look at the implementation of the project with which you are experimenting from the experiment environment at any time. It does not matter whether the experiment is running or whether it has been stopped.

Also as with the offline experiment, components specified in C Code offer additional opportunities for displaying debugger information or error messages during experimenting. You can embed debugger or error messages in your C code. Debugger information is displayed in the Debugger window which can be opened during experimenting. Error messages are displayed in the ASCET monitor window. The debugger in the online experiment works like the debugger in the offline experiment (see the section "Experimentation", subsection "Running Experiments" in the ASCET online help).

3.1.3 Standalone Mode

If you have a suitable experimental target, you can run ASCET experiments in standalone mode, without the experimentation environment. For this purpose the experimental target has to be equipped with flash memory and it must be possible to boot from it.

To load the experiment in the Flash memory:



- From the "Experiment Target" combo box, select **Online (RP)**.
- Select **Build** → **Flash Target**.

When you are working with the ES910 or RTPRO-PC target, you are asked if you want to continue.

- Click **Yes** to flash the target.

When you activated the **Use ETAS Network Manager (enables 'Select Hardware')** option in the hardware options, the hardware selection window may open.

Section "To select hardware (with ETAS Network Manager):" describes the required actions in this window.

The code generated by ASCET is written to the flash memory of the experimental target instead of to the RAM. A startup routine for booting from the flash memory is integrated into the code. The target

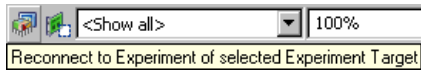
hardware will now execute the ASCET model after each reset.

To experiment in standalone mode:

- Select **Build** → **Reconnect**

or

- click on **Reconnect to Experiment of selected Experiment Target** to switch to a running online experiment.



When you activated the **Use ETAS Network Manager (enables 'Select Hardware')** option in the hardware options, the hardware selection window opens under certain conditions.

Section "To select hardware (with ETAS Network Manager):" describes the required actions in this window.

The online experimentation environment will start up as if the experiment had been started from scratch.

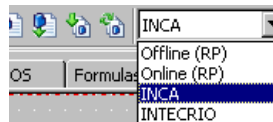
You can also reconnect to online experiments running in non-standalone mode that you disconnected from earlier. To disconnect from a running experiment simply exit the experimentation environment, without first stopping the online experiment running on the experimental target.

3.2 Experimenting with INCA

You can not only experiment in ASCET with your project but also in INCA (from Version 4.0.4), together with the add-on INCA-EIP. The project editor offers a function for this purpose which allows convenient transfer of the experiment.

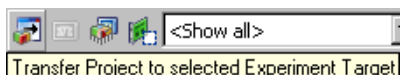
To initiate a transfer:

- Open the project with which you want to experiment.
- From the project properties in the project editor, "Build" node, select the ES1130 or ES1135 or ES910 or RTPRO-PC target and a suitable compiler.
- From the "Experiment Target" combo box, select INCA.



The **Transfer Project to selected Experiment Target** and **Reconnect to Experiment of selected Experiment Target** buttons are now available.

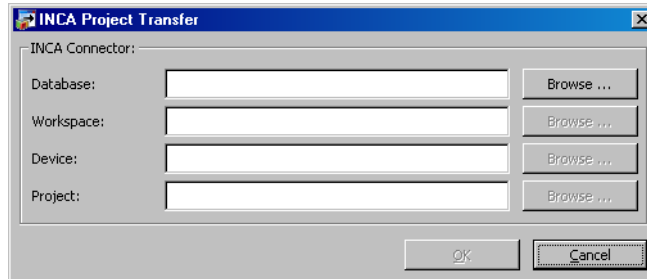
- Click on **Transfer Project to selected Experiment Target**



or

- select **Build** → **Transfer**.

The "INCA Project Transfer" window opens.



In this window, you define the INCA database, the workspace, and the project within the INCA database you want to use.

If you click one of the **Browse** buttons or the **OK** button, INCA will be launched, if it is not already used. If you have installed several INCA versions, the version that was installed last will be launched – even if this is not the version with the highest version number.

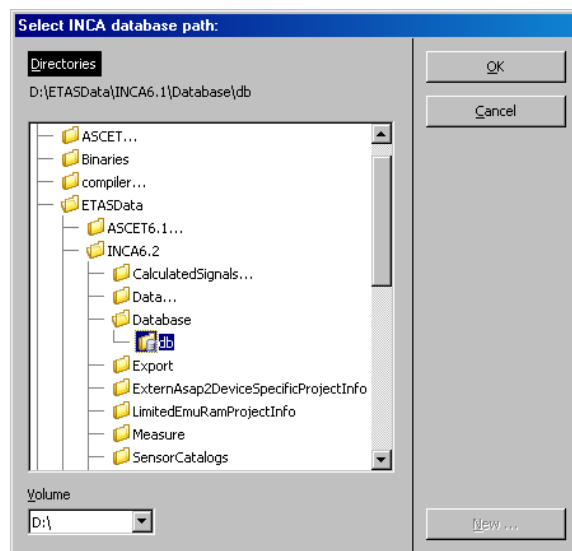
Tip

If your INCA version is too old (i.e. V3.x or older), or if INCA is not installed on your PC, an error message is displayed which cancels the transfer.

To set the INCA database path:

- In the "INCA Project Transfer" window, enter an existing database path in the "INCA Database" field
- or
- click on the **Browse** button to search for a database.

The "Select INCA database path" window opens. The databases are shown as folders with an overlay icon (🗄️).



- Select an INCA database path and click **OK**.

Tip

*In this window, ASCET databases are marked in exactly the same way as INCA databases. Make sure you really select an **INCA database**.*

Or

- click the **New** button to create a new directory, in which a new INCA database for the project transfer will be created. The new, empty directory is shown as a normal folder.

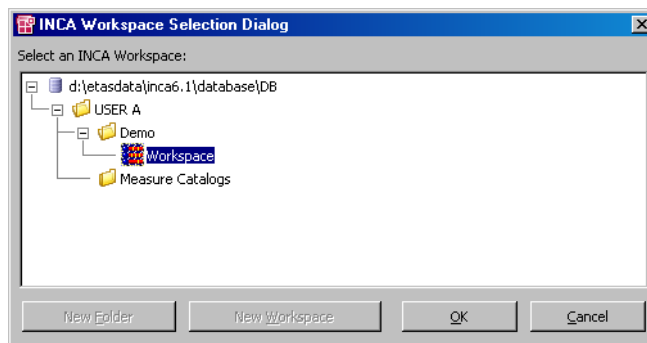
To select an INCA workspace:

- In the "INCA Project Transfer" window, enter the path and name of an existing workspace in the "INCA Workspace" field

or

- click on the **Browse** button to search for a workspace.

The "INCA Workspace Selection Dialog" window opens. It displays the workspaces in the selected INCA database.



- Select an existing workspace

or

- create a new workspace in a new or existing folder. The procedure is described on page 30.
- Once you have selected a workspace, click **OK**.

To create a folder/workspace in the "Workspace Selection Dialog" window:

1. Creating a folder

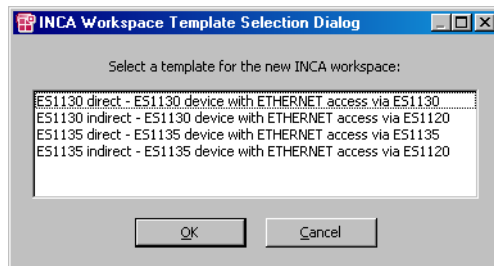
- In the "INCA Workspace Selection Dialog" window, select the database name or a folder. The **New Folder** button is activated.
- Click on **New Folder**.

- In the "Create New INCA Folder" window, enter a name and click on **OK**.

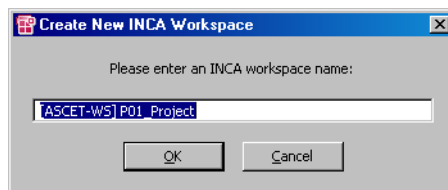
The new folder is created.

2. Creating a workspace

- Select a folder.
The **New Workspace** button is activated.
- Click on **New Workspace**.
The "INCA Workspace Template Selection Dialog" window opens.



- Select a template for the workspace and click **OK**.
The dialog window "Create New INCA Workspace" opens.



The name [ASCET-WS] <ascet project name> is assigned.

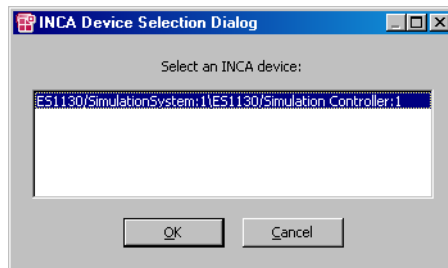
- Enter a name for the workspace and click on **OK**.
The new workspace is created in the selected folder and selected.
- Click **OK** to close the "INCA Workspace Selection Dialog".

To select an INCA device:

- In the "INCA Project Transfer" window, enter a device in the "INCA Device" field

or

- click on the **Browse** button to search for a device.
The "INCA Device Selection Dialog" window opens. It contains all suitable devices.



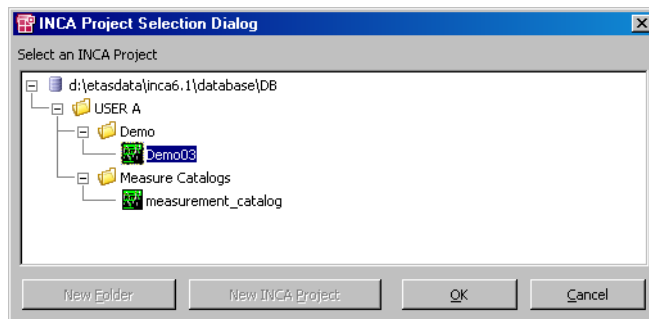
- Select a device and click **OK**.

To select an INCA project:

- In the "INCA Project Transfer" window, enter the path and name of an existing project in the "INCA Project" field

or

- click on the **Browse** button to search for a project.
The "INCA Project Selection Dialog" window opens. It displays the projects in the selected INCA database.



- Select an existing project
- or
- create a new project in a new or existing folder. The procedure is described on page 32.
 - Once you have selected a project, click on **OK**.

Tip

The selected project is replaced by the transferred project without any warning when the transfer starts.

To create a folder/project in the "INCA Project Selection Dialog" window:

1. Creating a folder

2. Creating a project

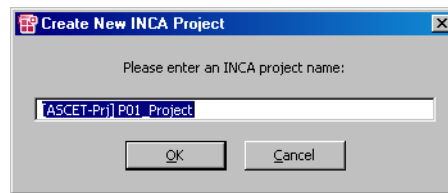
- To create a new folder, proceed as described under "Creating a folder" on page 30.

- Select a folder.

The **New INCA Project** button is activated.

- Click on **New INCA Project**.

The "Create New INCA Project" input window opens. The name [ASCET-Prj] <ascet project name> is assigned.



- Enter a name for the project and click on **OK**.
The new project is created in the selected folder and selected.

To start a transfer:

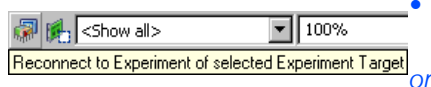
- Once you have made all settings in the "INCA Project Transfer" window, click on **OK**.
The transfer of the ASCET project to INCA is executed. First of all the ASAM-MCD-2MC code is generated and the project is refreshed if necessary (i.e. code generated, compiled and linked).
If you selected an existing project in the "INCA Project" field, it is overwritten.

Once transfer is complete, you can execute the experiment in INCA. For more details on how to do this, refer to the INCA and INCA-EIP documentation.

In addition, the Back-Animation in ASCET provides you with a special experiment environment in which you can calibrate values in the standard manner. The measure system of this experiment environment works in the standard way but is reduced in function in comparison to offline and online experiments in ASCET: oscilloscope, Recorder and Data Logger are not available. These need synchronous measuring which is not given for Back-Animation when experimenting with INCA. Instead, use the relevant instruments of INCA.

To use Back-Animation:

- Start the INCA experiment with your project.
- In the ASCET project editor, make sure that **INCA** is selected in the "Experiment Target" combo box.
- Click the **Reconnect to Experiment of Selected Experiment Target** button



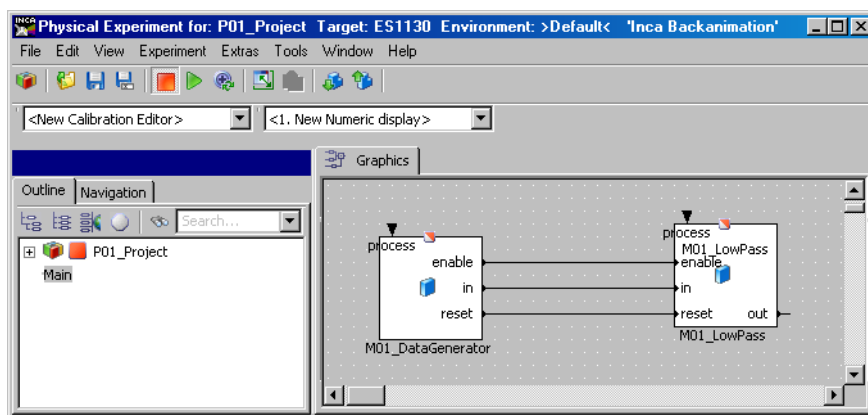
or

- select **Build** → **Reconnect**.

When you activated the **Use ETAS Network Manager (enables 'Select Hardware')** option in the hardware options, the hardware selection window opens under certain conditions.

Section "To select hardware (with ETAS Network Manager):" describes the required actions in this window.

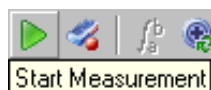
The connection to the running INCA experiment is established. The "Physical Experiment ..." window opens. "INCA Backanimation" indicates the special experiment environment.



Unlike with the online and offline experiment, this window only contains the "Graphics" tab.

- Create the necessary measure windows (see ASCET online help, "Experimentation" section) and set these up.
- Create the necessary calibration windows (see ASCET online help, "Experimentation" section) and set these up.
- Select **Experiment** → **Start Measurement**

or



- click the **Start Measurement** button to start measurement.

The displays of the measure and calibration windows are updated cyclically.

You can load, save and export environments as described in the section "Experimentation" of the ASCET online help. When you load an environment which contains unavailable elements (e.g. an oscilloscope), these are ignored.

The monitor function (see the ASCET online help) for monitoring numeric and logical variables is available. You can activate the function for individual or all variables of a component. The setting of the monitor function is saved in the environment.

You can navigate between the components of your project (see the ASCET online help).

If your project contains state machines, you can use the animation function for state machines (see the ASCET online help).

You can write data from the experiment into the ASCET model or onto the hard disk; you can also read in data from the hard disk. This is described in the ASCET online help, "Experimentation" section.

To end Back-Animation:

- Select **File** → **Exit**

or



- click the **Exit to Component** button.

The Back-Animation is ended and the experiment environment closed. The INCA experiment, however, continues running.

3.3 Experimenting with INTECRIO

If you have installed both ASCET-RP and INTECRIO, you can also experiment with your Rapid-Prototyping project in INTECRIO.

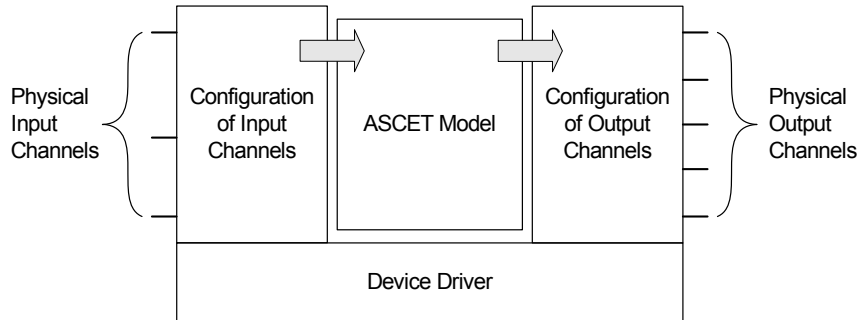
General instructions are given in the ASCET online help, section "INTECRIO Connectivity / ASCET-RP". A specific example can be found in the ASCET Getting Started manual¹, section 4.10 "Using INTECRIO Connectivity".

¹ ASCET V6.3 GettingStarted.pdf in the ETAS\ETAS Manuals folder or on the ASCET installation disk

4 Hardware Configuration

ASCET-RP allows easy prototype development for your ASCET model in a real-time environment on the ES1000, ES900 or RTPRO-PC hardware.

For that purpose, the model has to be embedded in your ES1000, ES900 or RTPRO-PC environment. On the logical level, the ASCET model has to be connected to the I/O channels.



This connection is performed in two steps. The Hardware Configurator (section 4.1 and online help) is used to describe and configure the I/O boards, i.e. to configure the physical input and output channels. Each of these channels is then mapped onto one ASCET message, which serves as an input or output to the ASCET model.

The configuration is saved to an XML description file (extension *.hwx) and stored with the ASCET project.

4.1 Hardware Configurator

The Hardware Configurator (available since ASCET-RP V6.1) is installed with ASCET-RP.

INTECRIO uses the same editor for hardware configuration; if a compatible INTECRIO version (i.e. V4.3 or higher) is installed on your PC, you can use that tool for hardware configuration.

To enable Hardware Configurator usage:

- In the Component Manager (or in the project editor), select **Tools** → **Options** to open the ASCET options window.
- In the options window, open the "Hardware\Hardware Configuration" node.
- Deactivate the **Use legacy HWC Editor (deprecated)** option.

With that, using the Hardware Configurator installed with ASCET-RP is enabled. The **Tools** → **RTIO** menu in the project editor is disabled in newly created projects.

- To use a separately installed INTECRIO, proceed as follows.

- Activate the **Use separately installed INTECRIO for hardware configuration** option.
- In the "HW configurator path" field, enter or select the installation directory of the INTECRIO version you want to use.
- In the "HWC working directory" field, enter or select a working directory for the Hardware Configurator.
- If you want to use ES910 or RTPRO-PC as target, you have to use the ETAS Network Manager to establish hardware connection (see section 2.2).
 - Open the "Hardware\Hardware Connection" node.
 - Activate the **Use ETAS Network Manager (enables 'Select Hardware')** option.

To create a hardware configuration for the Hardware Configurator:

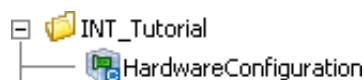
You have to create an empty hardware configuration in the Component Manager.

- In the Component Manager, select **Insert → Hardware Configuration**.

A new hardware configuration appears in the "1 Database" or "1 Workspace" list. Its name is highlighted for in-place editing.

- If desired, edit the name.
- Press <ENTER> to confirm the name.

The hardware configuration is displayed as follows.



To create a hardware description file component:

If you want to perform an ETK bypass or use a daisy chain, you *have to* create, at first, a hardware description file component in the Component Manager.

CAN databases, LIN description files, and FlexRay configuration files can optionally be added as hardware description file components, too. However this is not mandatory to use them for hardware configuration purposes.

Tip

*A2L files imported with **File → Import** or the **Import** button **cannot** be used with the Hardware Configurator.*

- In the Component Manager, select **Insert** → **Hardware Description File**.

The "Hardware Description File" dialog window opens. The following selections are available:

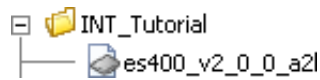
ASAM-2MC (*.a2l)	for ETK bypass
LDF (*.ldf)	LIN configuration
Daisy Chain,	daisy chain configuration
FIBEX (*.xml)	FlexRay configuration
CANdb (*.dbc)	CAN database

- Select the file you want to import.
- Click on **OK**.

A new hardware description file appears in the "1 Database" or "1 Workspace" list. The name of the imported file is used as default name; it is highlighted for in-place editing.

- If desired, edit the name.
- Press <ENTER> to confirm the name.

The hardware configuration is displayed as follows.



To add the hardware description to the hardware configuration:

After you created the hardware description file component, you have to add it to your hardware configuration.

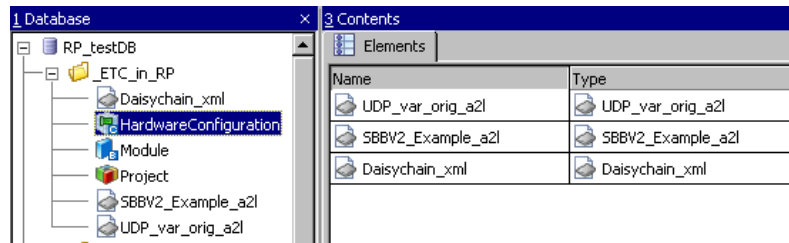
Tip

Only hardware description files added to a hardware configuration are available in the Hardware Configurator for this hardware configuration.

- In the Component Manager, select the hardware configuration you want to use.
The "3 Contents" field shows the "Container Components" tab (see ASCET online help on Containers).
- Right-click in the "Container Components" tab and select **Add** from the context menu.
The "Select Item" window opens.
- In the "1 Database" or "1 Workspace" list, select the hardware description files you want to add.

- Click on **OK**.

The hardware description file is added to the container and displayed in the "Container Components" tab.



For other editing possibilities, see the ASCET online help, section "Container".

To set up the project and open the Hardware Configurator:

The Hardware Configurator can be opened only via the project editor.

- Create and set up the project as usual.
- In the project editor, use **Insert** → **Component** to add the hardware configuration to the project.

Tip

A project can contain only **one** hardware configuration.

- Double-click the hardware configuration.

Or

- Right-click the hardware configuration and select **Open Component** from the context menu.

Or

- Select **Edit** → **Open Component**.

The Hardware Configurator opens.

Tip

Working with the Hardware Configurator is described in the Hardware Configurator online help, accessible via <F1> or the **Help** menu or the  button.

If OS or compiler settings of the project do not match the hardware configuration (if, e.g., the target PC or an ASCET-SE target is selected), a message of the following kind opens:

```
Current target <target name> does not support specification of hardware configuration. Do you want to edit the project properties?
```

OK opens the "Project Properties" dialog window. Select a suitable target and open the Hardware Configurator.

4.2 Migration of Existing Projects

Existing projects that contain configurations created with the legacy HWC editor must be migrated to the new Hardware Configurator.

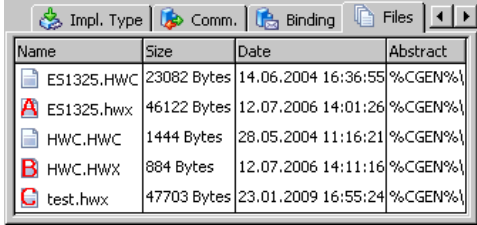
Tip

*Only legacy configurations available in *.hwx file format can be converted. The *.hwc files from older versions will be ignored. Projects containing only *.hwc files cannot be converted.*

To migrate a single project:

There are several possibilities to migrate projects from the legacy HWC editor to the new Hardware Configurator.

- Open the project you want to migrate.
The "Files" tab shows the existing configurations, e.g. the ones shown below.



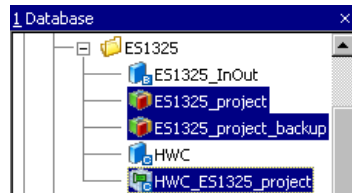
Name	Size	Date	Abstract
ES1325.HWC	23082 Bytes	14.06.2004 16:36:55	%CGEN%\
ES1325.hwx	46122 Bytes	12.07.2006 14:01:26	%CGEN%\
HWC.HWC	1444 Bytes	28.05.2004 11:16:21	%CGEN%\
HWC.HWX	884 Bytes	12.07.2006 14:11:16	%CGEN%\
test.hwx	47703 Bytes	23.01.2009 16:55:24	%CGEN%\

- In the project editor, select **Tools** → **RTIO** → **Convert Configuration**.
The "ASCET-RP: Convert Hardware Configuration" window opens.
You are informed that *.hwc files will not be converted, and that, during conversion, a backup of the original project will be created.
Cancel aborts the procedure.
- Click **Convert** to start the conversion.
Once the conversion procedure has started, existing *.hwx configurations are converted; existing *.hwc configurations are ignored.
A new hardware configuration (cf. page 37) is created that contains the hardware systems created from the *.hwx files. This hardware configuration is added to the project; the old *.hwx/*.hwc files and the HWC module are removed.
The ASCET monitor window summarizes the actions that have been performed.
When migration is complete, a message window opens.
- Confirm with **OK**.

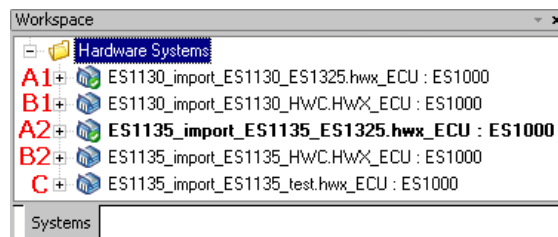
- Open the newly created hardware configuration in the Hardware Configurator (cf. page 39).
- Save the hardware configuration.

With that, the migration process is finished.


The "1 Database"/"1 Workspace" list now contains the converted project <project name>, the backup project <project name>_backup, and the newly created hardware configuration HWC_<project name>.



When you open <project name>, you can see HWC_<project name> in the "Outline" tab. The "Files" tab is empty. In the project properties, ERCOSEK 4.3 is selected as operating system. When you open the Hardware Configurator (page 39), you can see the hardware systems created from the *.hwx files.



If a *.hwx file contains a configuration that can be used with ES1130 and ES1135 (such as A and B in the figure on page 40), two hardware systems are created from the *.hwx file, one for each target (e.g., A1/A2 and B1/B2 in the figure above).

If more than one hardware system is created for the same target (e.g., A1 and B1 in the figure above), one is marked as active system by a green circle: 

To migrate all projects with legacy configurations:

You can convert all projects in the entire database.

- In the Component Manager, select **Tools** → **Database** → **Convert** → **Convert All Projects with Legacy Hardware Configuration**.
The "Hardware Configuration Migration" wizard opens.
- Read the text carefully and follow the instructions in the wizard.
The third page of the wizard lists all projects that can be migrated. Projects that contain no legacy configuration, or only *.hwc configurations, are not listed.
- Select the projects you want to migrate.

- If you want to create a backup copy of the database, activate the option below the project list.
- Click **Next** to continue.
The conversion starts. Page 4 of the wizard shows the conversion progress, and page 5 summarizes the results.
- Close the wizard.
To finish the migration process, the newly created hardware configurations must be opened in the Hardware Configurator and saved.

4.3 HWC Editor (Legacy)

Tip

Beginning with ASCET-RP V6.3, the legacy HWC editor is no longer available.

5 Hardware Systems

The operation of ASCET-RP requires an ETAS experimental system, i.e. one of the following systems:

- ES1000.2
- ES1000.3
- ES900 system (with ES910.2 or ES910.3)
- RTPRO-PC

These hardware systems can also be used with the INTECRIO Connectivity feature of ASCET. Please refer to the ASCET online help, section "INTECRIO Connectivity / ASCET-RP", for descriptions of the hardware systems.

6 Legacy HWC Editor

Tip

Beginning with ASCET-RP V6.3, the legacy HWC editor is no longer available.

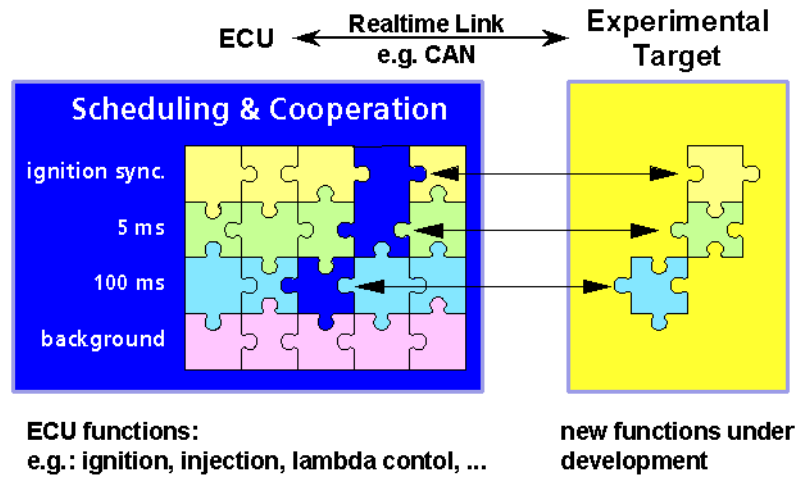
7 The ETK Bypass

This chapter describes the RTIO Package for ETK bypass. It assumes that the user is familiar with the necessary ASCET techniques; the use and operation of ASCET is not explained.

7.1 ETK Bypass: Definition

In an ETK bypass, certain functions of the control unit (ECU) are outsourced to a simulation computer, i.e. the PowerPC processing node of the ETAS experimental system. In doing so, data is transferred from the control unit to the experimental hardware. Based on these data, individual ECU functions are computed on the experimental hardware. Thus, outsourced data can be easily modified and tested. The results are transferred back to the ECU.

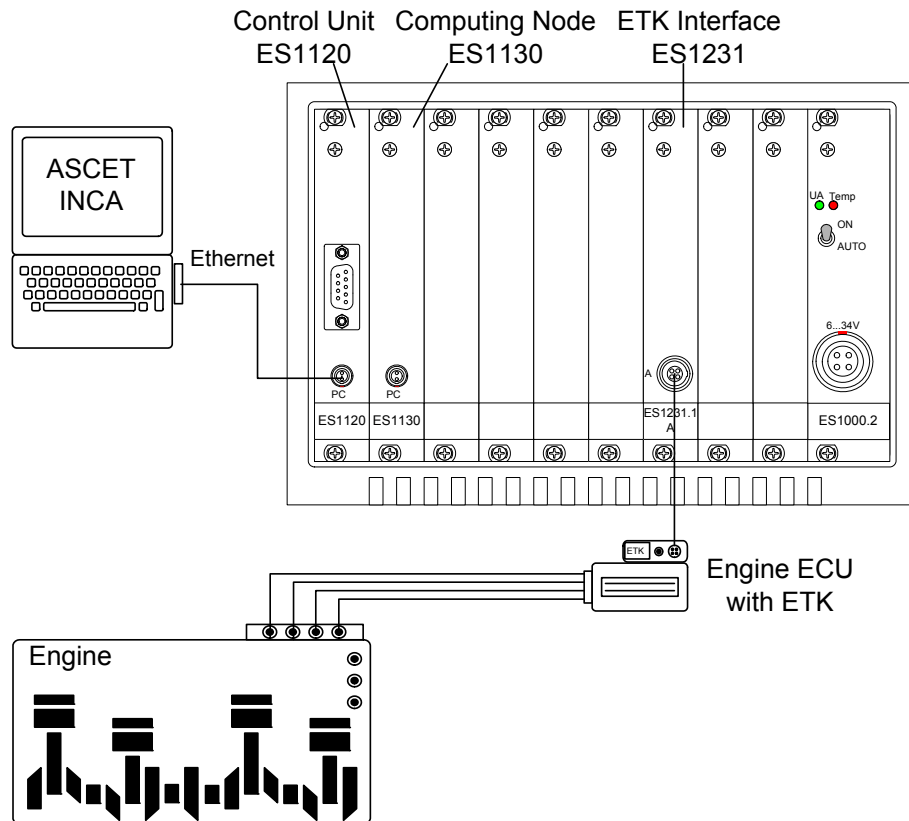
Modifications in the control unit software, referred to as "bypass hooks", determine the functions to be outsourced.



The connection to the ECU is implemented via an ETK (emulator test probe). A dual ported RAM is used for the communication between ECU and ETK.

7.2 Hardware Configuration of an ETK Bypass

The figure below shows a sample configuration for an ETK bypass application with the ES1000.2 system:



In an ETK bypass application, the outsourced functions are specified in ASCET; code is then generated from the specification that can be executed on the PPC module of the ETAS experimental system. The generated code is downloaded by the host PC to the ETAS experimental system. The ETAS experimental system is connected to the ETK of the control unit via an ETK interface (ES1000 system: ES1232; ES900 system: ETK Bypass device).

Additionally, interface parameters can be set in the control unit program that are required for the configuration of the software interface between the simulation computer and the control unit. Furthermore, new software versions can be downloaded to the control unit. Measurement and calibration tasks in the control unit can be performed while running the bypass.

7.3 ASCET Project for the ETK Bypass

Each ASCET RTIO project for ETK bypass needs to consider the following special features:

- A hardware configuration (cf. section 4.1) must be added to the ASCET project.

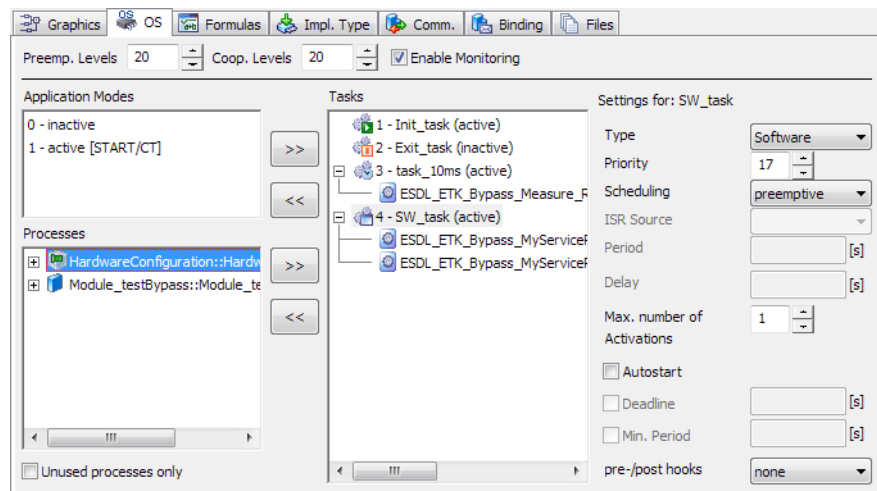
- The task list in the OS Editor must include an init task (Type: Init / Application Mode: active) and an exit tasks (Type: Init / Application Mode: inactive).

Tip

Init and Exit are reserved keywords; they must not be used as task names.

- The task list in the OS Editor must contain the necessary tasks that can be assigned to the signal groups which provide the bypass data received from the driver or the bypass data to be sent to the driver. These tasks must have the type *alarm* or *software*.
The task assignment must be specified manually in the Hardware Configurator.
- Global messages are available for the RTIO communication.

The following illustration shows an example configuration in the "OS" tab:



The following table shows the tasks settings:

Task Name	App. Mode	Trigger Mode	Prio.	Group	Max. No of Act.	Period
Init_Task	active	Init	-	-	-	-
Exit_Task	inactive	Init	-	-	-	-
task_10ms	active	Alarm	16	cooperative	1	-
SW_task	active	Software	17	preemptive	1	-

7.4 How the ETK Bypass Works

To conduct a bypass project, a control unit with bypass hooks, i.e. software modified for the bypass, is required. The bypass hooks enable you to switch between the functions running in the control unit and those running on the simulation computer. The bypass hooks include all information required to transmit the bypass input data to the simulation computer and to process the bypass output data in the control unit.

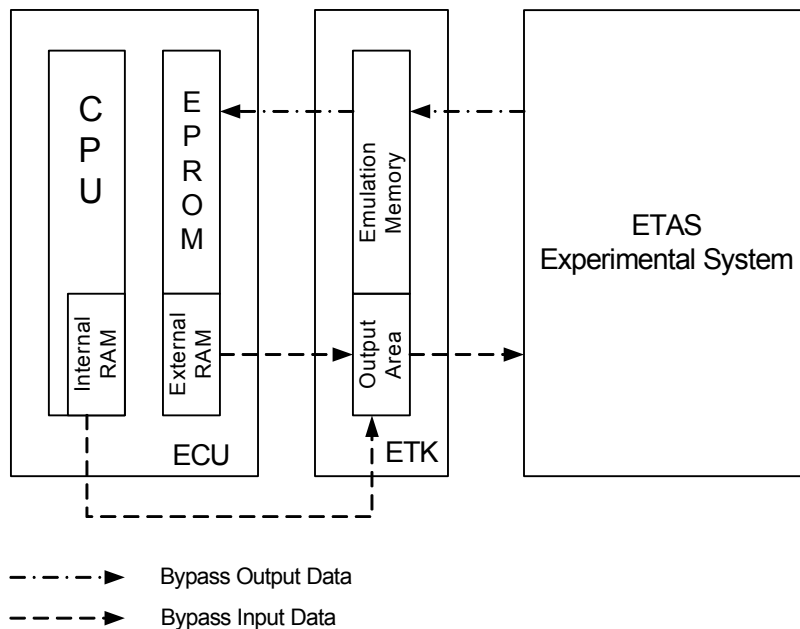
The bypassed functions in the control unit are usually also calculated if the bypass is enabled. But instead of the results of the bypassed functions, the results obtained by the simulation computer, i.e., the bypass output data, are used.

The bypass input data is the data that the experimental target reads from the control unit. This data is used to calculate the outsourced functions. The calculation results are returned to the control unit program as bypass output data.

The ETK memory is divided into an emulation memory area and an output area. The control unit program resides in the emulation area; this area replaces the control unit ROM. The bypass output data is also stored in this area because the control unit cannot read directly from the output area.

The output area is used only for transferring the bypass input data from the control unit to the simulation computer. This area contains a copy of the control unit RAM. Depending on the ETK model and control unit, the CPU-internal RAM of the control unit can also be located in this area.

The following figure illustrates the data flow within an ETK bypass project:



The data flows cyclically within the bypass project, i.e. the bypass input data is first read and then processed by the hardware system. Then the bypass output data calculated in the hardware system is written back to the control unit where it is processed further. Reading the bypass input data is synchronized by a time- or angle-synchronous grid in the control unit. Writing the data back to the control unit is not synchronized.

7.5 Data Exchange Between Control Unit and ETAS Experimental System

The DISTAB data exchange method is used for data transfer between the ETAS experimental system and the control unit.

For several years, the DISTAB 12 method has been used for the interaction between the control unit and the ETK. DISTAB 12 supports measured data of up to two bytes in length.

Four-byte integer variables or four-byte or eight-byte real or float variables require the use of DISTAB 13 or DISTAB16. These supports capturing 1, 2, 4 and 8 byte long measured data from the control unit regardless of its format (signed / unsigned integer or float).

For handling DISTAB, the ETK bypass project requires various parameters that can be passed to ASCET in a complete ASAM-MCD-2MC project description file matching the current software version of the control unit. DISTAB is defined in the DISTAB_CFG section of the TP_BLOB:

```

/begin DISTAB_CFG
0xC          /* type of display table:          */
             /* 0xC =DISTAB12, 0xD =DISTAB13      */
0x1          /* Data type of display table:          */
             /* 1=byte 2=word (ECU Data Mode)        */
             /* additional code table for          */
             /* distab13 depending on bus      */
             /* width/bus access (see distab13 */
             /* spec. for more information)  */
MSB_LAST    /* Byte Order: MSB_FIRST/MSB_LAST      */
0x383000    /* Trigger Segment Address              */
0x0         /* Trigger Configuration                */
TRG_MOD 0xB7 /* Dyn. length for TRG_MOD              */
             /* (special code)                       */
/end DISTAB_CFG

```

Tip

The comments are usually not included in the ASAM-MCD-2MC file. They are added here only for clarity.

ASAM-MCD-2MC is an established standard in the automotive industry for describing a control unit project (calibration parameters, measured variables, conversion rules, addresses, etc.). The ASAM-MCD-2MC file needs to be created for each new program version and should therefore be the result of the software development process. Detailed knowledge of the DISTAB method is not required for the ETK process because all necessary settings are part of the bypass hooks. Nevertheless, we will briefly look at the working principle to facilitate examining the ETK memory for diagnostic purposes.

The communication mechanisms for the data transfer between the control unit and the simulation computer are referred to as "bypass channels".

For the number of channels, see the online help Hardware Configurator.

ASCET-RP supports several AML versions (see the following sections for more details). ASAM-MCD-2MC files of all supported versions may contain XETK_AML V1.0 blocks; the content of such a block is of no consequence for working with ASCET-RP.

Bypass Communication (AML V1.1)

The bypass communication is explained here using bypass channel A of the ES1231. The process is the same for both channels. Tab. 7-1 contains the names and description of the parameters used in the ASAM-MCD-2MC file (*.a21, AML V1.1) to define both channels.

Tip

*The parameter names are not part of the ASAM-MCD-2MC standard; they are not necessarily included in the *.a21 file. They are added manually to the examples of this manual to provide clarity.*

Parameter	Description
BYPASS_S	Start address of pointer list for bypass input data (bypass channel A)
BYPASS_X	Start address of pointer list for bypass input data (bypass channel B)
CHNL_S	Start address of data buffer for bypass input data (bypass channel A)
CHNL_X	Start address of data buffer for bypass input data (bypass channel B)
CHNL_T	Start address of data buffer for bypass output data (bypass channel A)
CHNL_Y	Start address of data buffer for bypass output data (bypass channel B)
TRGID_S	Trigger ID address for bypass input data (bypass channel A)
TRGID_X	Trigger ID address for bypass input data (bypass channel B)
BPMAX_S	Size of data buffer for bypass input data (bypass channel A)
BPMAX_X	Size of data buffer for bypass input data (bypass channel B)
BPMAX_T	Size of data buffer for bypass output data (bypass channel A)
BPMAX_Y	Size of data buffer for bypass output data (bypass channel B)
TRGSEG_A	Trigger address for bypass channel A
TRGSEG_B	Trigger address for bypass channel B

Tab. 7-1 Bypass communication parameters (AML V1.1)

The parameters for channel A are provided in a QP_BLOB in the IF_DATA ETK section of the ASAM-MCD-2MC file.

```

/begin IF_DATA ETK
  /begin SOURCE "BYPASS A"
    0
    0
  /begin
    QP_BLOB

```

```
4      /* Acquisition raster;                */
      /* 1=A (typ. angle synchronous)        */
      /* 2=B (typ. time synch. 10ms)        */
      /* 3=C (typ. time synch. 100ms)       */
      /* 4=S/T angle synch. (bypass only)   */
      /* 5=X/Y time synch. (bypass only)    */
100     /* BPMAX_S                            */
0x81025E /* BYPASS_S                          */
0x3801E0 /* CHNL_S                            */
0x38302E /* TRGID_S                          */
2       /* trigger repetition rate          */
        /* (worst case)                    */
100     /* BPMAX_T                            */
0x8103F2 /* CHNL_T                          */
/end QP_BLOB
/end SOURCE
...
/end IF_DATA
```

Fig. 7-1 schematically shows the process of the bypass cycle with the DISTAB data exchange method using the example of bypass channel A. The numbers represent the individual steps of the bypass cycle:

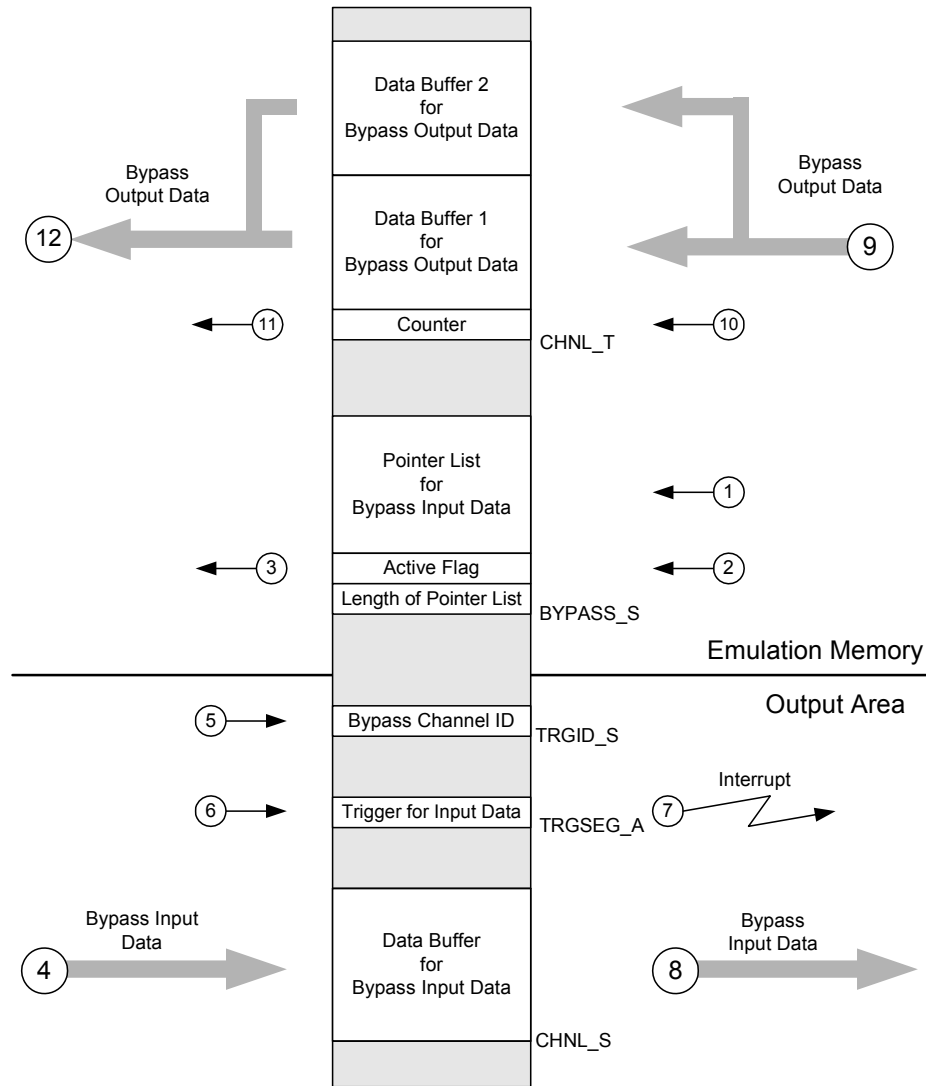


Fig. 7-1 Schematic process of a bypass cycle (DISTAB method)

1. When starting the bypass experiment on the simulation computer, the pointer list for the bypass input data is filled.

DISTAB12	DISTAB13 and DISTAB16
For each byte to be sent to the simulation computer, there is a pointer pointing to the address of the byte in the control unit memory. With the DISTAB 12 method, data is always transmitted in bytes; therefore, two pointers are required to send a word. After filling the pointer list, its length is written into the first byte of the list.	Bytes 4 to 7 of the pointer list contain the number of 8-byte, 4-byte, 2-byte, and 1-byte signals. The following bytes contain the signal addresses, beginning with the first 8-byte signal, and ending with the last 1-byte signal. Each address covers 4 bytes; the byte order of the addresses is determined by the <code>Byte Order</code> parameter (cf. page 49).

The bypass offsets (cf. step 11, page 54) are transferred in this step, too.

2. After the pointer list has been filled, the active-flag is set to 1. This initiates communication between the control unit and the simulation computer. Steps 1 and 2 are executed only once, at the beginning of a bypass experiment.

DISTAB12	DISTAB13 and DISTAB16
active-flag: the last bit of the second byte of the pointer list	active-flag: bit 0 of the first byte (byte 0) of the pointer list

3. The control unit continuously checks the active-flag byte of the pointer list. Once the last bit is set to 1, the bypass is activated and the data transmission begins.
4. The bypass input data is written by the control unit into the appropriate data buffer, `CHNL_S`.

DISTAB12	DISTAB13 and DISTAB16
This is done byte by byte; for each entry in the pointer list, the byte is written to the corresponding address of the data buffer. The transmission finishes when the number of bytes has been sent that is specified by the contents of the first byte in the list.	The data is written signal-wise, in the order provided by the pointer list. First, the bytes of the first 8-byte signal are written; the <code>Byte Order</code> parameter again defines which bit is written first. The other signals follow.

5. The control unit writes the ID of the current bypass channel (i.e., 4 for bypass channel A) to the address `TRGID_S`. This is necessary because only two addresses are available for the channel IDs, while there are a total of five channels. Two channels are used as bypass channels; the other three are available for calibration purposes. The ES1232 has up to 32 channels, 16 bypass channels and 16 measurement channels.
6. The trigger address `TRGSEG_A` is loaded by the control unit with a random value.

7. Writing to the trigger address triggers an interrupt in the simulation computer.
8. By reading the channel ID from the address `TRGID_S`, the simulation computer determines which channel was initialized and then reads the corresponding data buffer. From the information in the ASAM-MCD-2MC file, such as the conversion formula, the simulation computer is able to convert the raw data from the control unit into physical model variables that can then be processed by the ASCET model.
9. The simulation computer computes the functions outsourced to the bypass model on the basis of the bypass input data. The results are then written back to the appropriate data buffer as bypass output data. For each bypass channel there are two data buffers that are alternately written to. The simulation computer first internally (i.e. in the simulation computer) increments the counter that is later written to the `CHNL_T` address. The counter is incremented when the interrupt in step 7 occurs. Depending on the contents of the counter, one of the data buffers is used for writing. If the count is odd, the first data buffer is used; if it is even, the second buffer is used.
10. After writing to the data buffer, the internal count is written to the `CHNL_T` address.
11. The value of the counter is read by the control unit. The control unit decides on the basis of this value which data buffer is used for reading. If the value is odd, the first data buffer is read from; if it is even, the second buffer is read from. This ensures that the data consistency of the bypass output data is maintained.

The layout of these two buffers depends on the selected bypass output signal. Thus, the ECU must know where in the buffer the signal value actually is located. For that purpose, the signal position relative to the buffer start address—the *bypass offset*—is determined. For safety reasons, no pointer list exists for the bypass signals; instead, the ECU software contains a special bypass offset parameter for each signal. The bypass offsets are filled into the bypass offset parameters in step 1 (cf. page 53). These parameters can be directly accessed because they are located in an ECU application data area allocated in the ETK RAM.

12. The bypass output data is read by the control unit. In general, a safety mechanism is implemented in the control unit that determines the behavior of the control unit in case of a failure in the bypass communication. For details, please consult your control unit programmer.

Two data buffers are required for writing the bypass output data back, because this process is not synchronized. The results of the calculations in the simulation computer are written back as soon as the calculations are finished. The control unit may, therefore, request the results before or while the data is being written back. For this reason, a copy of the data is always held in a data buffer until the new data has been fully written back to ensure consistency of the bypass output data.

Other AML versions

A `QP_BLOB` of version AML V1.1.1 does not differ from the above AML V1.1 example (see page 50).

In AML V1.2, several changes have been made. Some parameters have been removed, and new parameters have been added. The (optional) names have changed, too. The QP_BLOB reads as follows:

```

/begin IF_DATA ETK
  /begin SOURCE  "BYPASS A4"
    0
    0
  /begin
    QP_BLOB
      0x0100      /* version 1.0          */
      15         /* hardware trigger          */
      INDIRECT   /* indirect/direct triggering */
      5          /* raster number/priority    */
                /* (32..1)                   */
      BYPASS     /* raster type                */
                /* (BYPASS/MEASUREMENT)     */
      0x38302E   /* trigger id/flag address for */
                /* indirect triggering        */
      100        /* Max. length of display table*/
                /* in bytes                   */
      0x81025E   /* address of the display table*/
                /* in the memory              */
      0x3801E0   /* address, where the ECU      */
                /* writes the display values   */
      100        /* Max. size of bypass receive */
                /* table                       */
      0x0        /* StartAddress of the Address */
                /* table for bypass output    */
      0x8103F2   /* Output address of the      */
                /* bypass table               */
      2         /* worst case raster timing   */
    /end QP_BLOB
  /end SOURCE
  ...
/end IF_DATA

```

AML V1.3, AML V1.4, AML V1.6 and AML V1.7 can be used, too. The differences between those versions and AML V1.2 are of no consequence for working with ASCET-RP.

Tip

AML V1.1.x describes ETK data for 3+2 measurement rasters. **AML V1.2 and higher** describes ETK data for multirasters.

7.6 Initially Required Information and Data

Before you can begin creating an ETK bypass project, you need some information and data about your control unit. Normally you obtain this from the control unit programmer who implemented the bypass hooks. The items listed in this section may be used as a checklist to ensure a smooth information transfer.

Control Unit Program

The bypass hooks control unit program must be loaded in the control unit.

ASAM-MCD-2MC File

In addition to other tasks, the ASAM-MCD-2MC file describes the data structures of the control unit. The ETK bypass package uses the ASAM-MCD-2MC file to access the input and output variables of the bypass project. Therefore, it is crucial to have the proper ASAM-MCD-2MC file matching the current software version of the control unit. The supported AML version depends on the hardware. The descriptions of the various boards include the necessary information.

Data Format of the Control Unit Processor

Different processor families use different word data storage formats that are known in general as *Big Endian* or *Little Endian*. When setting up the ETK bypass, it must be known whether the processor of the control unit uses the *Little Endian* or *Big Endian* storage format, or whether the word data storage format is specified in the ASAM-MCD-2MC file and can be read by the Hardware Configurator. In the *Big Endian* format (e.g., used by Motorola processors), the first byte represents the *most significant byte*. In the *Little Endian* format (e.g., used by Intel processors), the first byte represents the *least significant byte*.

Base Addresses

The start addresses of the transfer channels and the trigger addresses must be specified in the ETK bypass package, or the start addresses are specified in the ASAM-MCD-2MC file and can be read by the HWC Editor. The significance of the base addresses is explained in chapter 7.5 "Data Exchange Between Control Unit and ETAS Experimental System" on page 48.

Parameters for Activating the Bypass

The bypass must be enabled in the control unit program. If the bypass is enabled, the appropriate results in the simulation computer are used instead of the function results in the control unit program. Each bypass function is activated by setting a parameter with INCA.

Bypass Output Variables

The ETK bypass project can only modify certain variables in the control unit. These so-called bypass output variables are specified by the control unit programmer in the bypass hooks. The bypass output variables have to be known when creating the project, or the relevant information is stored in the ASAM-MCD-2MC file from where it can be read by the HWC Editor.

Bit Masks for the Transformation in the NEAR Region

The DISTAB 12 method supports only 16-bit memory addresses for transfer between the control unit and the bypass computer. For control units using memory addresses larger than 16 bits, these have to be transformed to 16 bits by means of a data page pointer. The `LongAdrANDMask` and `LongAdrORMask` bit masks are used for this purpose. The bit masks need to be specified in the Hardware Configurator.

What to do in case of an Error / Safety Mechanism

In general, a safety mechanism is implemented in the control unit that defines the behavior of the control unit in case of a failure in the bypass communication. Sometimes the results of the control unit functions are used; at other times the control unit is switched into *Reset* or *Emergency* mode (depending on the control unit). It is crucial to know the behavior of the control unit if an error occurs, particularly for bypass experiments in the vehicle.

8 HWC Items

Tip

Beginning with ASCET-RP V6.3, the legacy HWC editor is no longer available.

9 ES1000 Tutorial

The ES1000 tutorial describes the following tasks that use supplied examples:

- with the Hardware Configurator:
 - setup of ES1222 (section 9.2)
 - setup of ES1303 (section 9.3)
 - setup of ES1325 without (section 9.4) and with trigger (section 9.5)

An example for using RTPRO-PC is given in chapter 10 "RTPRO-PC Tutorial".

ASCET-RP V6.3 includes the sample files `Tutorial INTECRIO.*` and `Tutorial RTIO.*` (* = `exp` or `axl`). During the installation, the sample files are stored in the subdirectory `ASCET6.3\export` of your ASCET installation.

The sample files can be imported into a new or an existing database/workspace.

Tip

This tutorial is written under the assumptions that

- a) an ES1000 system is used,
- b) a database is used,
- c) the ETAS Network Manager is **not** used.

If you are using the ETAS Network Manager, slightly different behavior may result; see section 2.2 and chapter 3.

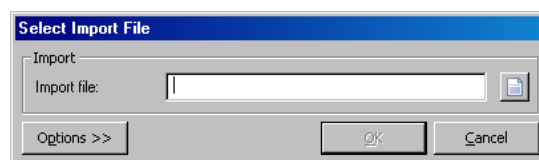
To create a new database:


- In the Component Manager, select the **File** → **New Database** menu option.
The "New Database" window opens.
- Enter the name for the new database.
- Click **OK**.
The database is created and opened.

Creating a new workspace is described in the ASCET online help.

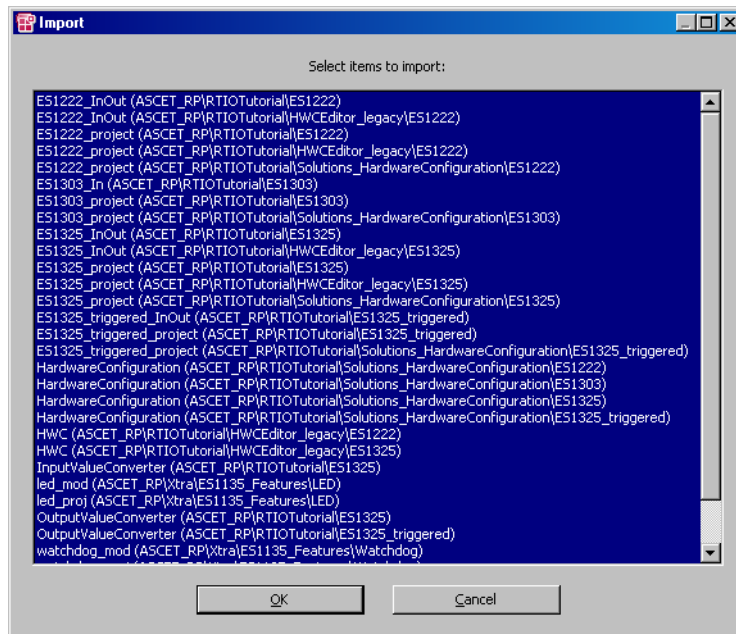
To import the exercise example:

- In the Component Manager, select **File** → **Import**.
The "Select Import File" window opens.



- Use the  button to select the `Tutorial RTIO.*` file from the `ASCET6.3\export` subdirectory of your ASCET installation.
The **OK** button is now available.
You can use the default import options, so that no further action is required.

- Click **OK** to start the import.
- Make sure all components have been selected for the import in the "Import" window..



- Click **OK**.
The files are imported. A list of the imported database items is shown in the "Imported Items" window.
- Close the "Imported Items" window.
- Import the Tutorial INTECRIO.* file the same way.
Make sure all components are selected for the import.

To activate the Hardware Configurator:

- In the Component Manager, select **Tools** → **Options** to open the ASCET options window.
- In the options window, open the "Hardware\Hardware Configuration" node.
- Deactivate the **Use legacy HWC Editor (deprecated)** option.
- Click **OK** to close the options window.

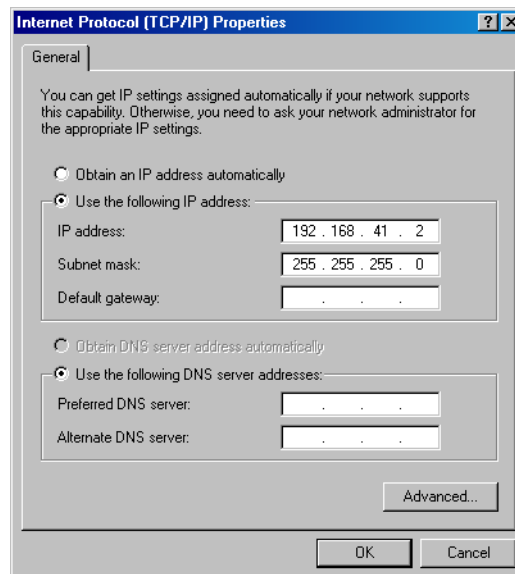
To configure the TCP/IP protocol options:

Tip

If you are using the ETAS Network Manager, the instruction is obsolete.

To avoid conflicts with a second network card that might be used for the LAN, the following TCP/IP settings should be selected:

- Disable the DHCP service.
- Enter the IP address 192 . 168 . 40 . 240.
- Enter the subnet mask 255 . 255 . 255 . 0.



- For the DNS service, use the local settings of your internal network.
- Disable the WINS service.
- Make sure that the "IP Forwarding" option is **not** activated.

9.1 Tutorial – Experimenting with INTECRIO

This part of the ASCET-RP tutorial has been moved to the ASCET tutorial. It is now found in the ASCET Getting Started manual¹, section 4.10 "Using INTECRIO Connectivity".

¹ ASCET V6.3 GettingStarted.pdf in the ETAS\ETAS Manuals folder or on the ASCET installation disk

9.2 Tutorial – ES1222 (CAN-IO)

The example for ES1222 contains three variables which are sent by CAN channel 1 and received by CAN channel 3. If both channels are connected (together with a terminator, if the cables are longer than 2 m; see also Fig. 9-3 on page 78), the experiment can be performed.

The model is present and the preparations required for integration of the ES1222 (tasks, messages) have been made; you have to set up the hardware configuration and generate the code.

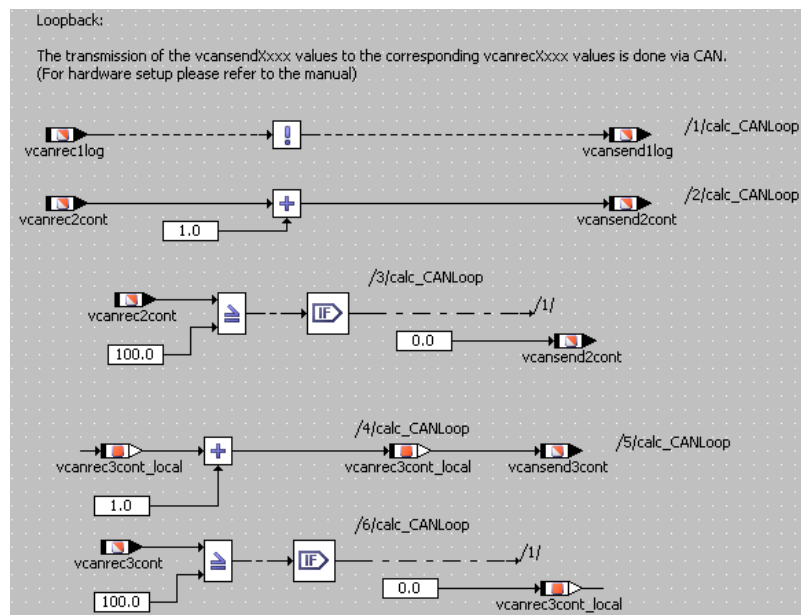


Fig. 9-1 ES1222 - ES1222_InOut module

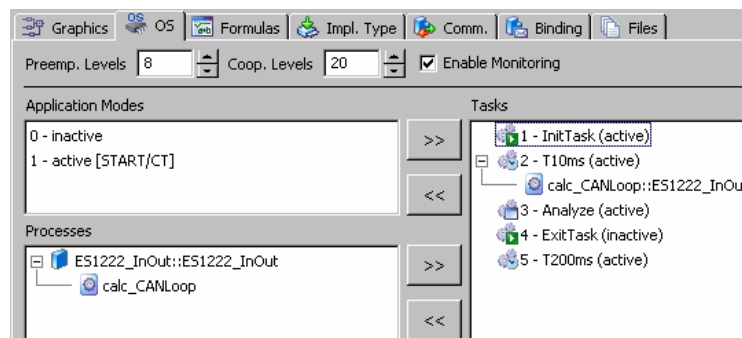
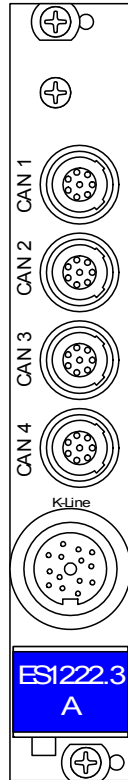


Fig. 9-2 ES1222 – Presets in the OS editor

9.2.1 The ES1222 Board

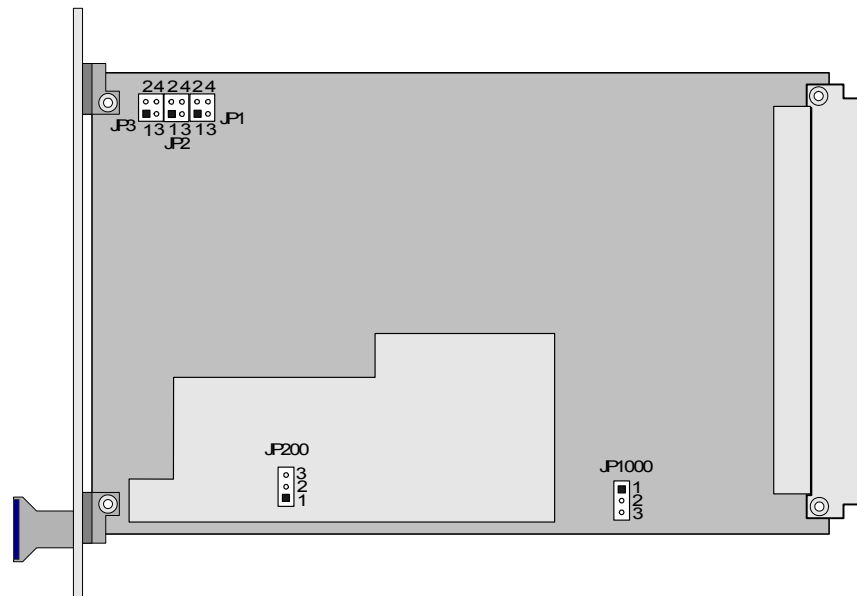
The figure below shows the front panel of the ES1222 CAN board.



The inputs CAN1 to CAN4 are independent and galvanically isolated CAN interfaces; the K wire input is a serial K and L line interface used for connecting VME-bus systems with external devices.

Jumpers of the ES1222

For the experiment to work, you must verify the correct configuration of the jumpers.



Jumper	PIN	Meaning
JP1	open	CAN1 and CAN2 are independent
JP2	open	
JP3	open	
JP1000	1-2 closed	
JP200 (ES1222.3 only)	any	K Line is not sed in the example

Operating Several ES1222 Boards

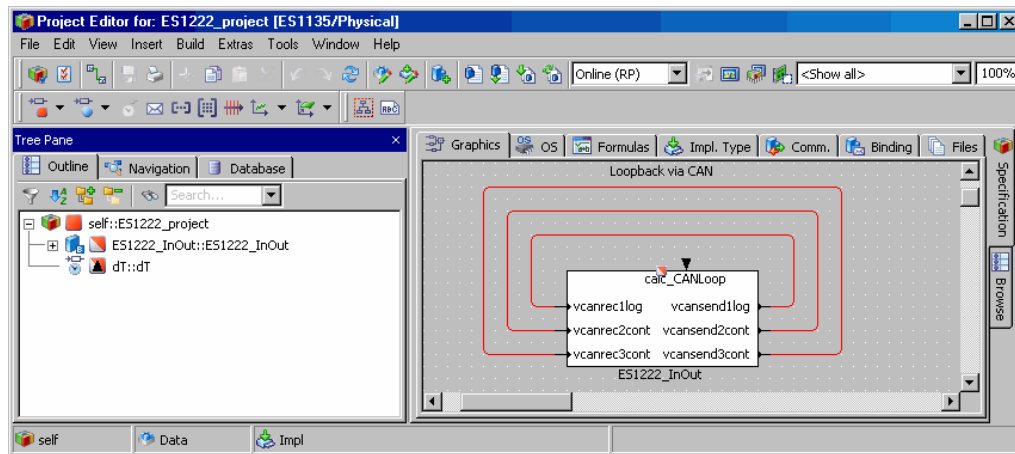
Up to four ES1222 boards can be used in one ES1000 system.

9.2.2 Sample Project

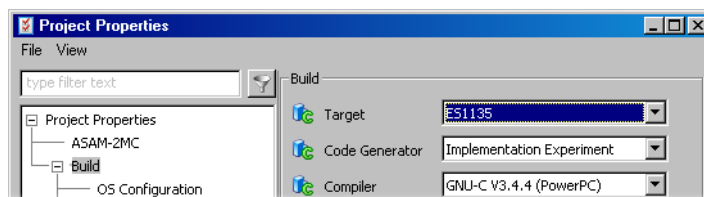
To open the exercise example:

- In the Component Manager, select the `... \RTIOTutorial\ES1222` folder.

- Open the ES1222_project project.



- Click the **Project Properties** button.
The "Project Properties" window opens.



- In the "Build" node, select the options
Target: ES1135 or ES1130,
Compiler: GNU-C V3.4.4 (PowerPC),
Operating System: ERCOSEK 4.3,

Tip

Only messages declared as "Exported" are available for communication.

9.2.3 Creating the Hardware Configuration

To prepare the hardware configuration:

You have to create an empty hardware configuration in the Component Manager and add it to the project.

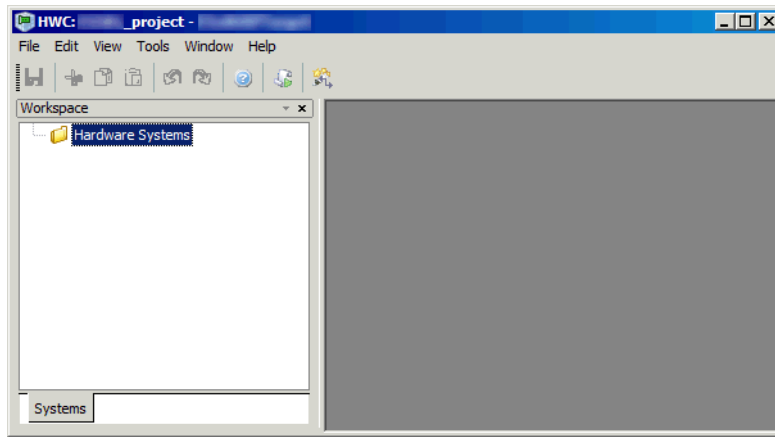
- In the Component Manager, select **Insert** → **Hardware Configuration**.
A new hardware configuration appears in the "1 Database" list. Its name is highlighted for in-place editing.
- If desired, edit the name (default: HardwareConfiguration).
- Press <ENTER> to confirm the name.

- Add the hardware configuration to the project, e.g., via **Insert Component** in the project editor.

To open the Hardware Configurator:

- In the "Outline" tab of the project editor, double-click the `HardwareConfiguration`.

The Hardware Configurator opens. The title bar contains the project name and the selected experimental target. The `Hardware Systems` folder always exists, it forms the root of the hardware configuration tree.

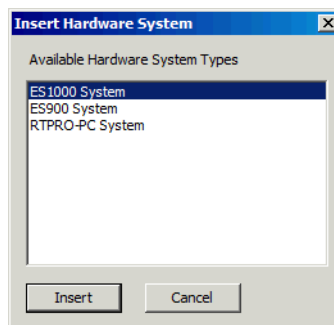


To specify the hardware configuration:

The hardware is described as a tree structure in the Workspace Browser.

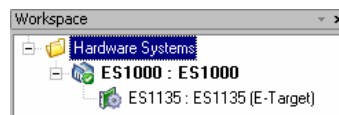
- In the Hardware Configurator, right-click the `Hardware Systems` folder and select **Add Hardware System** from the context menu.

The "Insert Hardware System" window opens. This window lists the available hardware systems.



- Select the `ES1000` system entry.
- Click **Insert**.

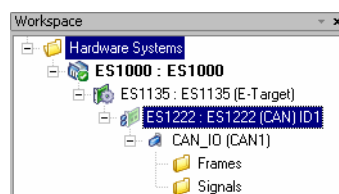
- In the "Rename" dialog window, enter a name (default: ES1000) and click **OK**.
The hardware system is added below the Hardware Systems folder.
- Right-click the hardware system and select **Insert Target** from the context menu.
The "Insert New Item" window opens. This window lists the items that can be inserted at the given level.
- In the "Available Items" list, select ES1135 or ES1130.
Make sure that you select the target that is selected in the "Build" node of the Project Properties (cf. page 64).
- Click **Insert**.
- In the "Rename" dialog window, enter a name and click **OK**.
The target is added below the hardware system.



To integrate and configure the ES1222:

1. Adding the ES1222

- Right-click the target and select **Insert** from the context menu.
The "Insert New Item" window opens.
- Select the ES1222 (CAN) entry.
This entry is used to describe the ES1222 CAN interfaces.
- Click **Insert**.
- In the "Rename" dialog window, enter a name (default: ES1222) and click **OK**.
The ES1222 board is added. One CAN_IO device and, below that, folders for Frames and Signals are added automatically.



2. Adding a second CAN_IO device

- Right-click the ES1222 board and select **Insert** from the context menu.

Since only CAN_IO devices can be added, the "Insert New Item" window is omitted, and the "Rename" window opens immediately.

- Enter a name and click **OK**.

3. Configuring the ES1222

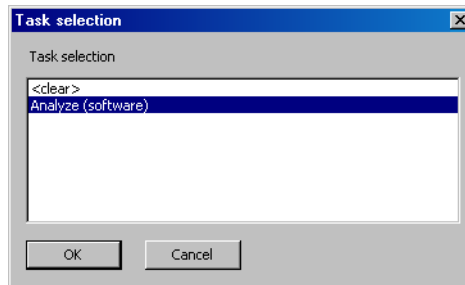
- Double-click the ES1222 board.

The editor for the board opens. It lists all board parameters. For the tutorial, you will only assign an analyze task.

Parameter	Value
Name	
Analyze Task	
Board-ID	ID1

- In the board editor, click the button in the "Analyze Task" row, "Value" column.

The "Task selection" window opens. It lists all tasks of the project that can be selected.



- Select the Analyze software task and click **OK**.

The board number in the Board-ID option is set automatically. Since the ES1222 board is the first of its type, the value is set to ID1.

To configure the CAN devices:

- In the Hardware Configurator, double-click the CAN_IO device.

The editor for the device opens. The "Parameters" tab lists all parameters for the device.

- In the "Parameters" tab of the device editor, click on the value 1000000 next to the "Baud Rate [Baud]" option.

The table field changes to a combo box containing the available baud rates.

	Parameter	Value
1	Name	CAN_IO
2	Port	CAN1
3	CANdb File	
4	Baud Rate [Baud]	500000
5	Special Timing BRP	1000000
6	Special Timing SJW	500000
7	Special Timing TSEG1	250000
8	Special Timing TSEG2	125000
9	Special Baud Rate [Baud]	1000000
10	Sample Mode	50000
	Parameters	20000
		10000
		special settings

- From the combo box, select a rate of 500000 Baud.

- Select the same baud rate and the port CAN3 for the second CAN_IO device (CAN_IO1).

9.2.4 Configuring the CAN Channels

Next, the two CAN channels must be configured, i.e. the necessary CAN signals and frames must be created.

The first channel (CAN_IO in the Hardware Configurator) will send the signals which are received by the second channel (CAN_IO1).

Channel 1

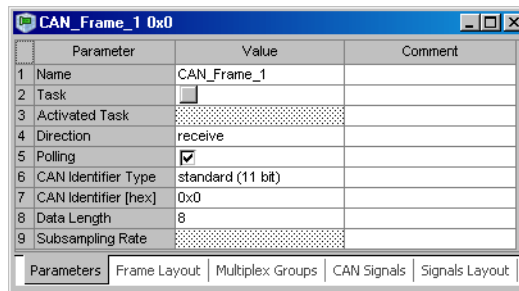
CAN Frames: The CAN messages are transmitted in form of CAN frames (signal groups).

To create CAN frames:

- In the Hardware Configurator, right-click the **F**rames folder below the first CAN_IO device (CAN_IO) and select **I**nsert from the context menu. The "Rename" dialog window opens.
- Enter a name and click **O**K. The CAN frame is added to the folder. Newly created frames always have the direction **R**ecieve.
- Add a second CAN frame.

To configure the first CAN frame:

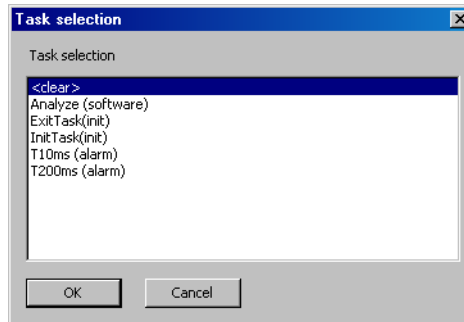
- In the Hardware Configurator, double-click the first CAN frame. The editor for the frame opens. The "Parameters" tab lists all parameters for the frame.



- In the "Parameters" tab of the frame editor, double-click in the "Name" row, "Value" column. The table field changes to an input field.
- Name the frame **F**rame_64.

- Click the button in the "Task" row, "Value" column.

The "Task selection" window opens. It lists all tasks of the project that can be selected.



- Select the T10ms alarm task and click **OK**.
- Click in the "Direction" row, "Value" column. The table field changes to a combo box.
- Select the direction Send.
- Double-click in the "CAN Identifier [hex]" row, "Value" column.
- Enter the value 0x64 for the identifier.
- Close the editor.

To configure the second CAN frame:

- Open the editor for the second CAN frame.
- Name the second frame `Frame_65`.
- Select the T200ms alarm task.
- Select the direction Send.
- Enter the identifier value 0x65.
- Close the editor.

CAN signals: The CAN signals are first created in the `signals` folder, and then assigned to a CAN frame. The example uses two numeric and one logical signal.

To create signals:

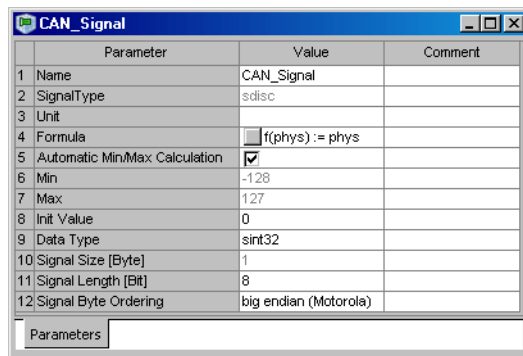
- In the Hardware Configurator, right-click the `signals` folder below the first CAN_IO device (CAN_IO) and select **Insert** from the context menu. The "Rename" dialog window opens.
- Enter a name and click **OK**. The CAN signal is added to the folder.
- Add two more CAN signals.

The logical and one numeric signal are transferred in the first CAN frame, the other numeric signal is transferred in the second CAN frame.

To configure a logical signal:

- In the Hardware Configurator, double-click the first CAN signal.

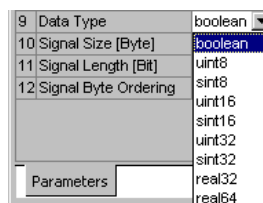
The editor for the signal opens. The "Parameters" tab lists all signal parameters.



	Parameter	Value	Comment
1	Name	CAN_Signal	
2	SignalType	sdisc	
3	Unit		
4	Formula	f(phys) := phys	
5	Automatic Min/Max Calculation	<input checked="" type="checkbox"/>	
6	Min	-128	
7	Max	127	
8	Init Value	0	
9	Data Type	sint32	
10	Signal Size [Byte]	1	
11	Signal Length [Bit]	8	
12	Signal Byte Ordering	big endian (Motorola)	

Parameters

- In the signal editor, name the signal `vcansend1log`.
- Click in the "Data Type" row, "Value" column.
The table field changes to a combo box which lists the available types.



9	Data Type	boolean	
10	Signal Size [Byte]	boolean	
11	Signal Length [Bit]	uint8	
12	Signal Byte Ordering	sint8	
		uint16	
		sint16	
		uint32	
		sint32	
		real32	
		real64	

Parameters

- From the combo box, select `boolean`.

To configure the numeric signals:

- Open the editor for the second CAN signal.
- Name the signal `vcansend2cont`.
- Select the "Data Type" `uint8`.
- Open the editor for the third CAN signal.
- Name the signal `vcansend3IRQ`.
- Select the "Data Type" `uint8`.
- Close the signal editors.

The configuration of the signals is not yet complete, but before the final configuration steps can be done, the signals must be assigned to CAN frames.

To assign CAN signals to CAN frames:

- Right-click the `Frame_64` frame and select **Insert** from the context menu.
The "Insert Item" window opens. It lists all available CAN messages.
- Select the `vcansend1log` signal and click **Insert**.
- Assign the `vcansend2cont` signal to the same CAN frame.
- Assign the `vcansend3IRQ` signal to the `Frame_65` CAN frame.

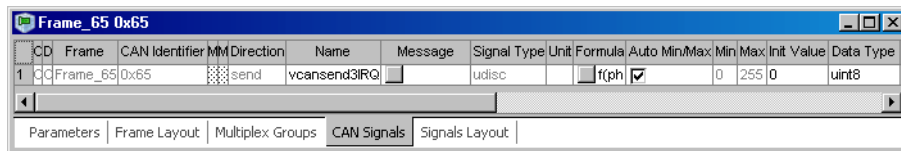
Now you can complete the signal configuration, i.e. assign ASCET messages to the CAN signals and edit the signals layout. However, these configuration options are only available when you edit the signals from the hierarchy branch beginning with the `Frames` folder.

During signal mapping, CAN signals and ASCET messages from the project are assigned to each other. A selection dialog can be opened via the "Message" parameter of each CAN signal assigned to a CAN frame. This dialog contains only messages featuring the *Exported* attribute. Make sure that you assign only ASCET messages that correspond to the frame direction, i.e.,

- frame direction = receive → receive-messages,
- frame direction = send → send-messages

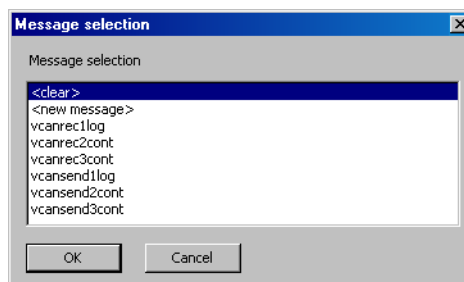
To assign ASCET messages manually:

- Double-click the `Frame_65` frame.
The editor for the frame opens.
- Go to the "CAN Signals" tab.



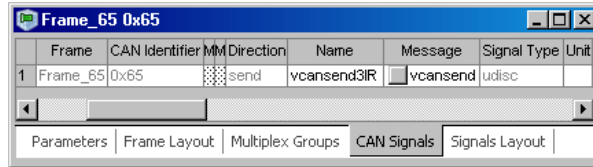
This tab contains the signal parameters of all signals assigned to the frame.

- Click the button in the "Message" column.
The "Message selection" window opens. It lists all messages in the project that can be selected.



- Select the ASCET message `vcansend3cont` and click **OK**.

The ASCET message is assigned to the `vcansend3IRQ` CAN signal.



- Close the frame editor.

To assign ASCET messages automatically:

The signals in `Frame_64` have the same names as two send-messages in the project. You can assign the messages automatically.

- Right-click the `Frame_64` frame and select **Automatic Signal Mapping** from the context menu.
The "Item Signal Mapping" window opens.



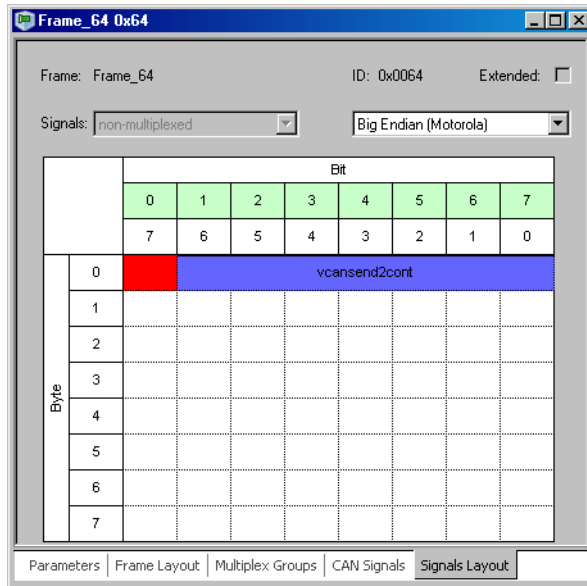
<F1> opens a detailed description.

- Activate the **Map unmapped signals** and **Create ASCET messages** options.
- Click **OK** to start the mapping.

The `vcansend1log` and `vcansend2cont` signals are assigned to the ASCET messages of the same name.

To edit the signals layout:

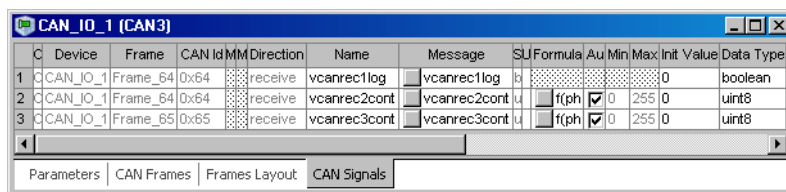
- In the frame editor for `Frame_64`, go to the "Signals Layout" tab.



<F1> opens a detailed description.

- Check the signal map for signal overlaps (i.e. two or more signals using the same bit); these are highlighted in red.
- Check for unwanted gaps (unoccupied bits) between signals.
- Remove any overlaps and/or unwanted gaps by dragging the associated signals.
- Close the frame editor.

This completes the configuration of the first CAN channel.



Channel 3

CAN frames: The second CAN_IO device needs two CAN frames with direction *Receive*.

To create and configure the CAN frames:

- In the Hardware Configurator, create two CAN frames in the `Frames` folder below the second CAN_IO device (`CAN_IO1`).

- Assign the following parameters to the first CAN frame in CAN_IO1:

Name	Frame_64
Task	T10MS
CAN Identifier [hex]	0x64

- Assign the following parameters to the second CAN frame in CAN_IO1:

Name	Frame_65
Polling	(deactivated)
CAN Identifier [hex]	0x65

The read action for this CAN frame is performed in the Analyze task (see page 68).

CAN signals: The second CAN_IO device needs three CAN signals.

To add and configure the CAN signals:

- In the Hardware Configurator, create three CAN signals (see page 70) in the Signals folder below the second CAN_IO device (CAN_IO1).

- Assign the following parameters to the first CAN signal in CAN_IO1:

Name	vcanrec1log
Data Type	Boolean

- Assign the following parameters to the second CAN signal in CAN_IO1:

Name	vcanrec2cont
Data Type	uint8

- Assign the following parameters to the second CAN signal in CAN_IO1:

Name	vcanrec3cont
Data Type	uint8

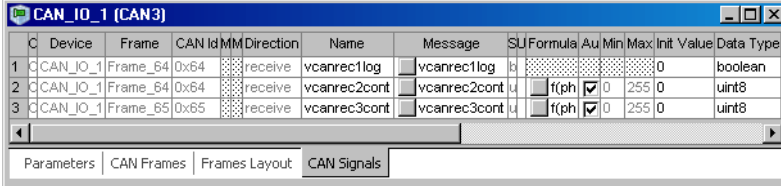
- Assign the first two signals to the Frame_64 frame.
- Assign the third signal to the Frame_65 frame (see page 72).
- Adjust the signals layout in Frame_64 (see page 74).

- Assign messages to the signals.

For the three signals, receive-messages with the same names exist in the project so that all messages can be assigned automatically (see page 73).

Of course, the messages can also be assigned manually as described on page 72.

This completes the configuration of the second CAN channel.



C	Device	Frame	CAN Id	MM	Direction	Name	Message	SU	Formula	Au	Min	Max	Init Value	Data Type
1	CAN_IO_1	Frame_64	0x64		receive	vcanrec1log	vcanrec1log	b					0	boolean
2	CAN_IO_1	Frame_64	0x64		receive	vcanrec2cont	vcanrec2cont	u	r(ph)		0	255	0	uint8
3	CAN_IO_1	Frame_65	0x65		receive	vcanrec3cont	vcanrec3cont	u	r(ph)		0	255	0	uint8

9.2.5 Saving the Hardware Configuration and Generating Code

The created hardware configuration must be saved, and code must be generated. This is done with a single command.

To save the hardware configuration and generate code:



- In the Hardware Configurator, click on the **Save** button

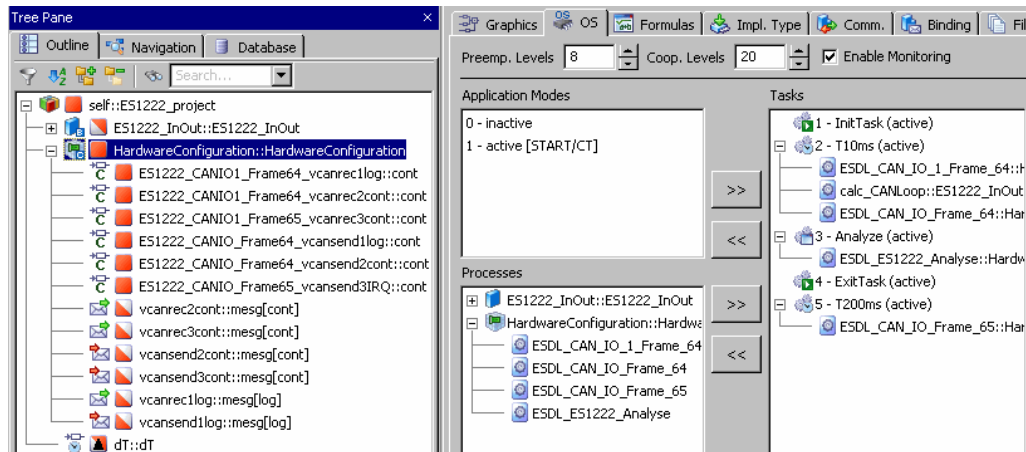
or

- select **File** → **Save Workspace** to save the hardware configuration.

A hardware configuration file (* .hwx) is written to the working directory specified in the Hardware options (see section 2.1.1 on page 13).

Code is generated for the hardware configuration; information, warning and error messages are shown in the ASCET monitor window.

If no error occurred, the HardwareConfiguration component in the project editor is updated, and the hardware processes are assigned to the respective tasks.



You can export the hardware configuration to a file.

To export the hardware configuration as a file:

- In the Hardware Configurator, right-click the ES1000 hardware system and select **Export** from the context menu.

The "Hardware Configuration Export" dialog window opens.

- Enter or select (via the button) path and file name for the export file.
- In the "Export Options" field, select the export option(s) (**Export referenced files** and/or **Store relative pathnames**) you want to use.
- Click **Finish** to perform the export.

More details are given in the Hardware Configurator online help.

9.2.6 Experimenting with the Sample Project

The hardware configuration is complete, and you can close the Hardware Configurator. The next step is to start the regular code generation for the experimental target from within the project editor.

Tip

It is not recommended to include the hardware configuration in the graphical display of the Project Editor. Since this module can change with each hardware configuration code generation process, it could result in an unfavorable representation.

To generate code for the experimental target:

- In the project editor, select **Build** → **Build All** to generate code for the entire project.

- Select **Build** → **View Generated Code** to view the generated code.

To experiment online:

- Connect the CAN 1 and CAN 3 inputs of the ES1222 with the appropriate cables and—if the cables are longer than 2 m—a terminator.

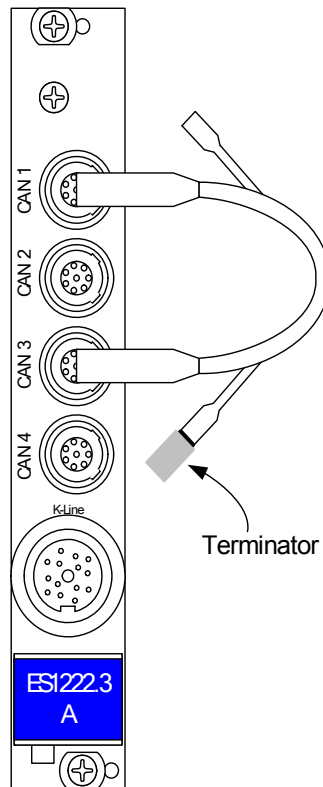
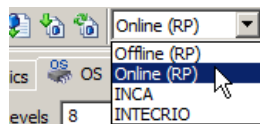


Fig. 9-3 ES1222 with cable and terminator

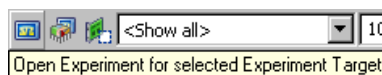
Tip

If you omit the terminator for long cables, no communication will take place.

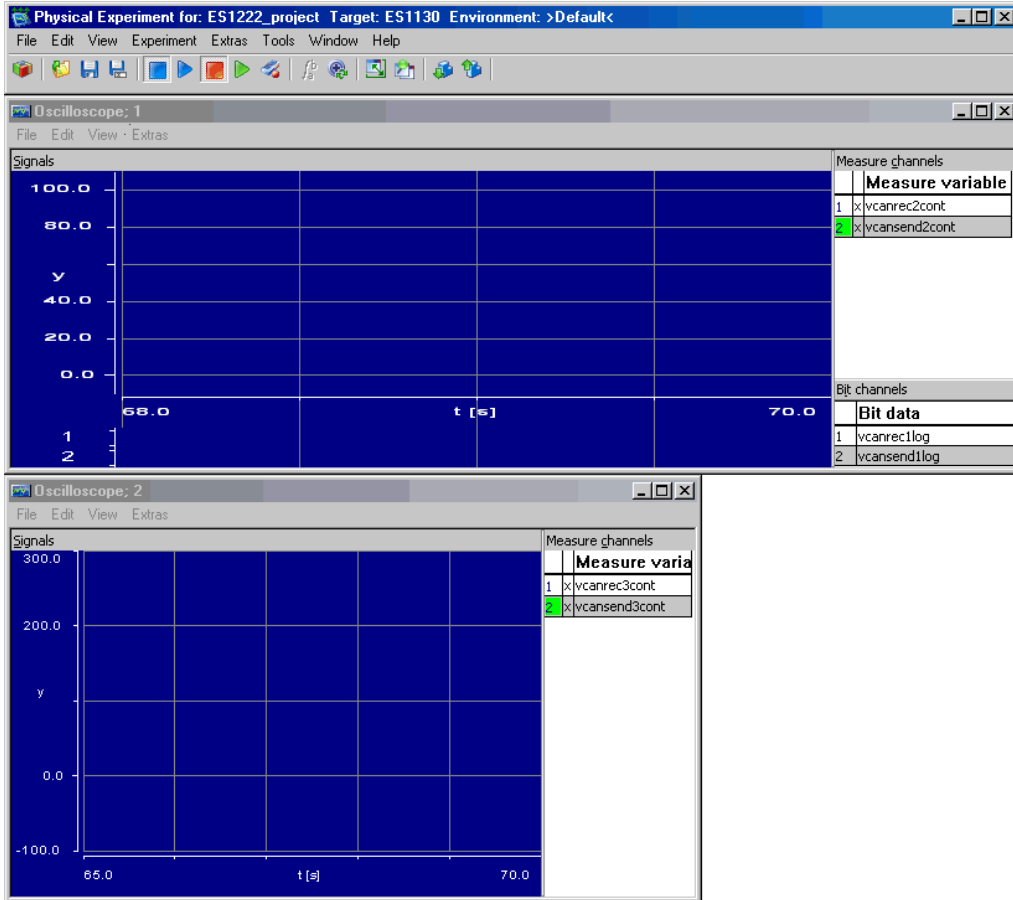


- Select **Online (RP)** from the "Experiment Target" combo box. The other entries are irrelevant for this tutorial.
- Select **Build** → **Experiment**

or



- click the **Open Experiment for selected Experiment Target** button. The "Physical Experiment" window and the pre-defined experiment environment consisting of two oscilloscopes open.



- In the "Physical Experiment" window, select **Experiment** → **Start OS**

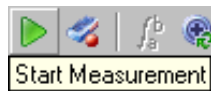
or



- click the **Start OS** button.
The operating system starts and the model is executed.

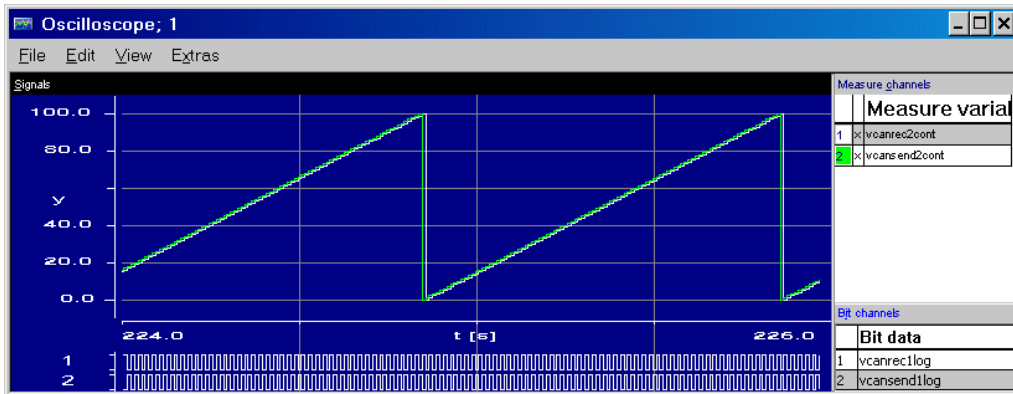
- In the "Physical Experiment" window, select **Experiment** → **Start Measurement**

or



- click the **Start Measurement** button.
The values of the ASCET messages are displayed on both oscilloscopes.

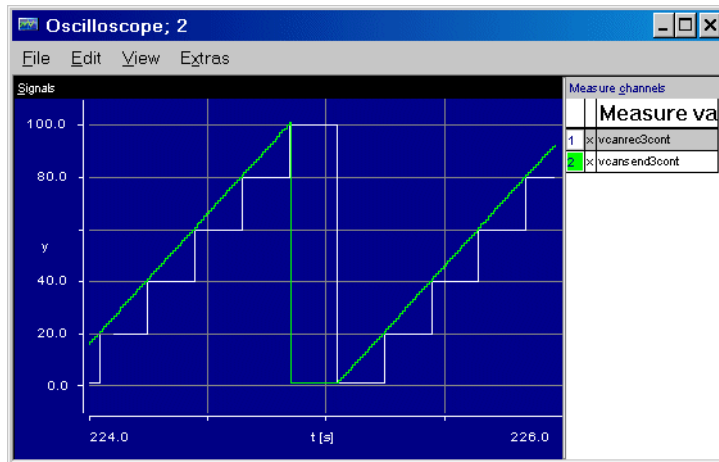
The top oscilloscope shows the `vcansend2cont` send-message (green, left curve) and the `vcanrec2cont` receive-message (white, right curve) in the "Signals" area. The bit display shows the `vcanrec1log` receive-message (top curve) and the `vcansend1log` send-message (bottom curve).



Characteristic and offset of the curves are generated as follows:

- The values for the send-messages are processed (`calcCANLoop` process in the `T10ms` task) and sent (`ESDL_CAN_IO_Frame64` process in the `T10ms` task) in the 10-ms raster.
- The processed value is read in again via receive-message (`ESDL_CAN_IO_1_Frame64` process in the `T10ms` task), but delayed by one time unit.

The bottom oscilloscope shows the `vcansend3cont` send-message (green curve with small steps) and the `vcanrec3cont` receive-message (white curve with large steps).



The `vcansend3cont` send-message is processed in the 10-ms raster (`calcCANLoop` process in the `T10ms` task), which causes the green curve to respond as smoothly as the curve of `vcansend2cont`. The value is sent in the 200-ms raster (`ESDL_CAN_IO_1_Frame65` process in the `T200ms` task) and received again in the interrupt mode via receive-message after being triggered by the `Analyze` task. The graduated curve for `vcanrec3cont` responds accordingly.

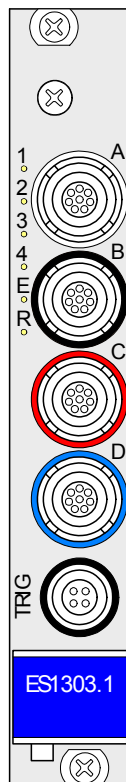
9.3 Tutorial – ES1303

This section describes the integration of an ES1303 in an ASCET project using the new Hardware Configurator. Prerequisites for the integration are the steps that are described in 5 "Hardware Systems". Remember that the mapping between the hardware channel and the model can only be specified by messages that are declared as *Exported* in the remaining model.

Since the ES1303 board can only receive, but not send, signals, this example does not include a final experimental demonstration.

9.3.1 The ES1303 Hardware

The figure below shows the front panel of the ES1303 analog/digital converter card:



The inputs A to D each bundle four input channels of the ES1303. The input TRIG is used for the trigger signals.

Input Voltage Ranges of the ES1303

The input voltage range of each channel the ES1303 board can be set either to ± 10 V or ± 60 V.

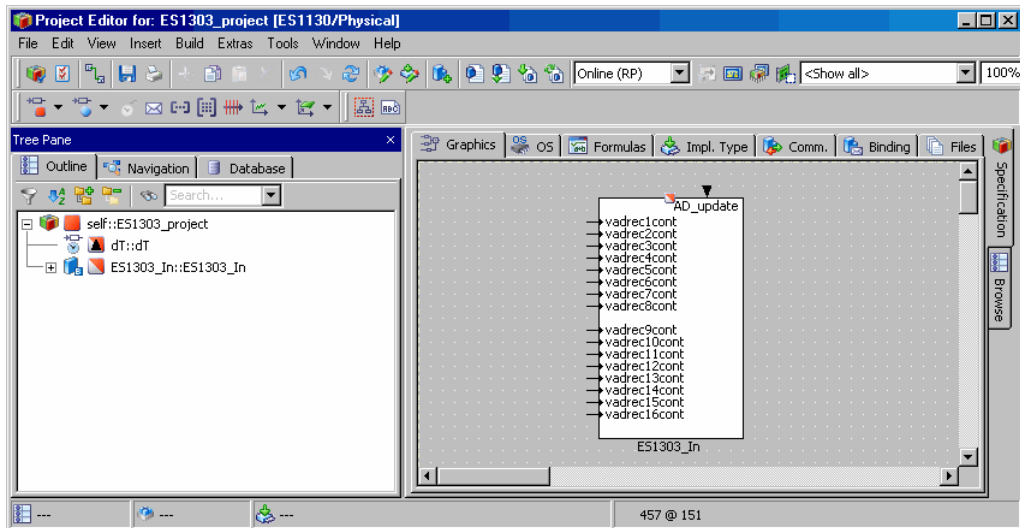
Use of Several ES1303 Boards

Up to 4 ES1303 boards can be used in one ES1000 system.

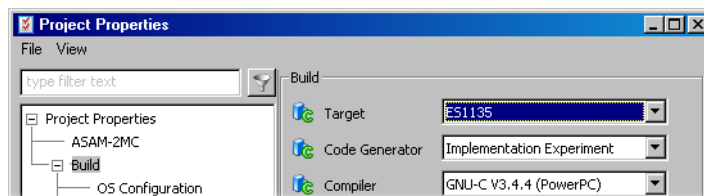
9.3.2 Sample Project

To open the exercise example:

- In the Component Manager, select the ... \RTIOTutorial\ES1303 folder.
- Open the ES1303_project project.



- Click the **Project Properties** button. The "Project Properties" window opens.



- In the "Build" node, select the options
Target: ES1135 or ES1130,
Compiler: GNU-C V3.4.4 (PowerPC),
Operating System: ERCOSEK 4.3.

Tip

Only messages declared as "Exported" are available for communication.

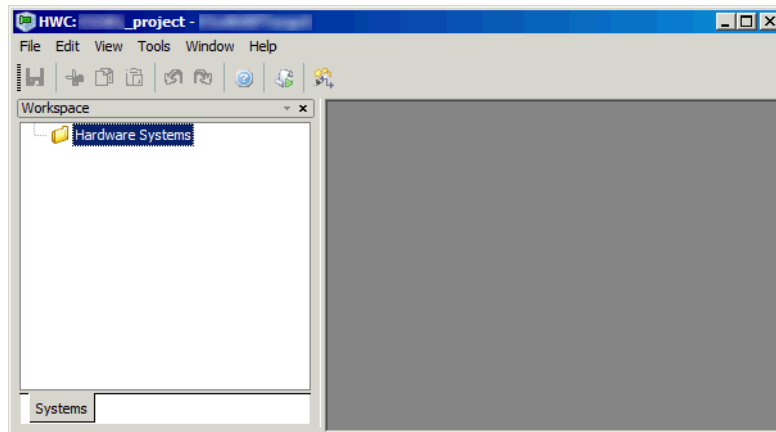
9.3.3 Creating the Hardware Configuration

To start the hardware configuration:

- Create an empty hardware configuration and add it to ES1303_project (see page 65).

- In the "Outline" tab of the project editor, double-click the `HardwareConfiguration`.

The Hardware Configurator opens. The title bar contains the project name and the selected experimental target. The `Hardware Systems` folder always exists, it forms the root of the hardware configuration tree.

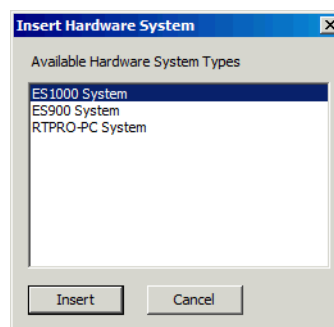


To specify the hardware configuration:

The hardware is described as a tree structure in the Workspace Browser.

- In the Hardware Configurator, right-click the `Hardware Systems` folder and select **Add Hardware System** from the context menu.

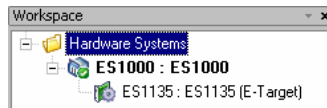
The "Insert Hardware System" window opens. This window lists the available hardware systems.



- Select the `ES1000` system entry.
- Click **Insert**.
- In the "Rename" dialog window, enter a name (default: `ES1000`) and click **OK**.

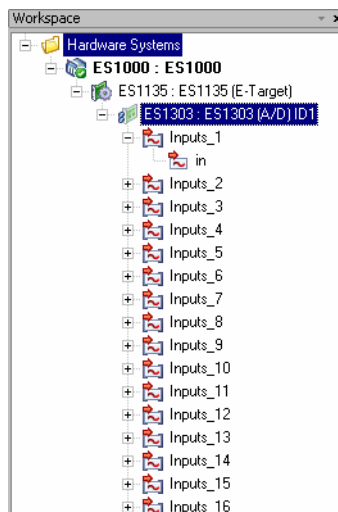
The hardware system is added below the `Hardware Systems` folder.

- Right-click the **Hardware Systems** folder and select **Insert Target** from the context menu.
The "Insert New Item" window opens. This window lists the items that can be inserted at the given level.
- In the "Available Items" list, select **ES1135** or **ES1130**.
Make sure that you select the target that is selected in the "Build" node of the Project Properties (cf. page 82).
- Click **Insert**.
- In the "Rename" dialog window, enter a name and click **OK**.
The target is added below the hardware system.



To integrate the ES1303:

- Right-click the target and select **Insert** from the context menu.
The "Insert New Item" window opens.
- Select the **ES1303 (A/D)** entry.
This entry is used to describe the ES1303 A/D interfaces.
- Click **Insert**.
- In the "Rename" dialog window, enter a name (default: **ES1303**) and click **OK**.
The **ES1303** board is added. For each channel, a signal group and a signal are added.



The item tree for the description of the sample system is now fully specified.

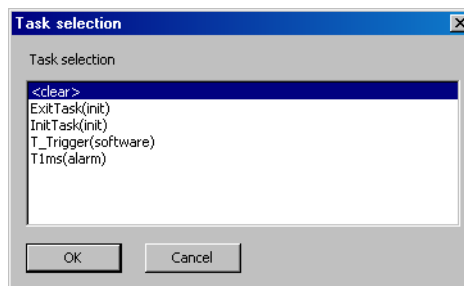
To configure the ES1303:

- Double-click the ES1303 board.
The editor for the board opens. Its "Parameters" tab lists all board parameters.

Parameter	Value
Name	ES1303
Analyze Task	T_Trigger
Board-ID	ID1

- In the "Parameters" tab of the board editor, click the button in the "Analyze Task" row, "Value" column.

The "Task selection" window opens. It lists all tasks of the project that can be selected.



- Select the T_Trigger software task and click **OK**.
The board number in the Board-ID option is set automatically. Since the ES1303 board is the first of its type, the value is set to ID1.
- If you want to limit the bandwidth of the input signal, activate the Anti-Aliasing option.
- In the Trigger Mode option, select the edge of the first trigger signal to start the measurement (e.g., Falling Edge).
- Use the Trigger Gate option to determine whether the second trigger signal is used (On) or not (Off).

Tip
For signal groups in polling mode, the options Trigger Mode and Trigger Gate are irrelevant.

The "Parameters" tab should now look as follows.

	Parameter	Value	Comment
1	Name	ES1303	
2	Analyze Task	T_Trigger	
3	Board-ID	ID1	
4	Anti-Aliasing	<input checked="" type="checkbox"/>	
5	Trigger Mode	Triggered at Falling Edge	
6	Trigger Gate	On	

Parameters | Signal Groups | Signals

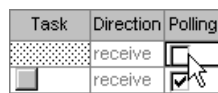
9.3.4 Configuring the ES1303 Signal Groups and Signals

Signal Group Configuration: The ES1303 provides a fixed signal group (input) for each signal. You select the task assignment, input voltage range and receive mode (interrupt-driven or polling).

To configure the signal groups:

- In the Workspace Browser, double-click the ES1303.
The editor for the board opens.
- In the board editor, go to the "Signal Groups" tab.
- Perform the following steps for each signal to be used.

1. Select the receive type



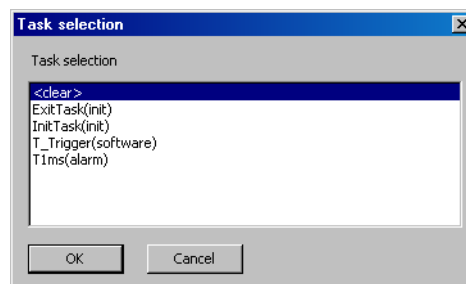
- To receive a signal group in interrupt mode, deactivate the `POLLING` option in the respective row.
At the same time, the "Task" field is reset and blocked.
- To receive a signal group in polling mode, use the default setting of the "Polling" column.

2. Assign a task (only for polling mode)

In the "Task" column, you can assign, to signal groups in polling mode, a task in which signal transfer is to take place.



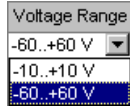
- In the row of the signal group you want to edit, click the button in the "Task" column.
The "Task selection" window opens. It lists all tasks of the project that can be selected.



- In the task selection list, select the `T1ms` task. This task should be used for data transfer.
You can also select more than one task in the task selection list. For the ES1303, these are generally the alarm tasks.
- Click **OK**.
The selected tasks are shown in the "Task" column.

3. Select the input voltage range:

- Click in the "Voltage Range" column in the relevant row.



The table cell becomes a combo box.

- Select a value for the voltage range from the combo box.

The selection is shown in the "Voltage Range" column.

The "Signal Groups" tab could now look as follows:

Device	Name	Task	Direction	Polling	Voltage Range	Hardware Channel
1 ES1303	Inputs_1		receive	<input type="checkbox"/>	-60..+60 V	1 (Port A, Pin 1+2)
2 ES1303	Inputs_2	T1ms	receive	<input checked="" type="checkbox"/>	-60..+60 V	2 (Port A, Pin 3+4)
3 ES1303	Inputs_3	T1ms	receive	<input checked="" type="checkbox"/>	-60..+60 V	3 (Port A, Pin 5+6)
4 ES1303	Inputs_4	T1ms	receive	<input checked="" type="checkbox"/>	-60..+60 V	4 (Port A, Pin 7+8)
5 ES1303	Inputs_5	T1ms	receive	<input checked="" type="checkbox"/>	-60..+60 V	5 (Port B, Pin 1+2)
6 ES1303	Inputs_6	T1ms	receive	<input checked="" type="checkbox"/>	-60..+60 V	6 (Port B, Pin 3+4)

Signal Configuration: The signals that were created when you inserted the ES1303 are contained in the "Signals" tab of the board editor.

To rename signals:

By default, all signals are named `in`. Rename all signals you want to use, so that each signal has a unique name.

- In the respective row, double-click in the "Name" column.
The table cell becomes an edit field.
- Change the signal name, e.g., to `in_1`.

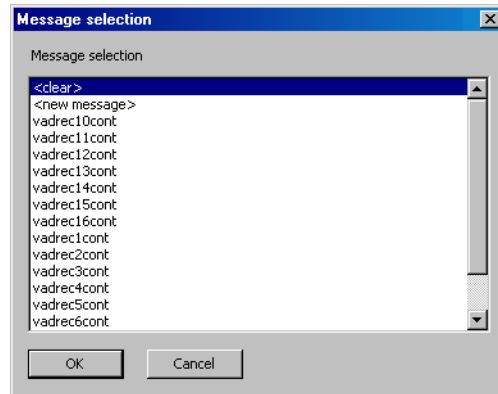
During signal mapping, ASCET messages from the project are assigned to the signals of the ES1303 signal groups. A selection dialog can be opened via the "Message" parameter of each signal. This dialog contains only messages which have the attribute *Exported* and correspond to the transfer direction of the signal group (here: *receive*); i.e.:

Direction = receive → Receive-Messages

To assign ASCET messages manually:



- Click the button in the "Message" column.
The "Message selection" window opens. It lists all messages in the project that can be selected.



- Use the "Message selection" window to map the sixteen receive messages one by one to the corresponding signals.
- Click **OK**.

The "Signals" tab could now look as follows:

	Device	Signal Group	Direction	Name	Message	Signal Type	Unit	Formula
1	ES1303	Inputs_1	receive	in_1	vadrec1cont	cont		f(phys) := phys
2	ES1303	Inputs_2	receive	in_2	vadrec2cont	cont		f(phys) := phys
3	ES1303	Inputs_3	receive	in_3	vadrec3cont	cont		f(phys) := phys
4	ES1303	Inputs_4	receive	in_4	vadrec4cont	cont		f(phys) := phys
5	ES1303	Inputs_5	receive	in_5	vadrec5cont	cont		f(phys) := phys
6	ES1303	Inputs_6	receive	in_6	vadrec6cont	cont		f(phys) := phys

9.3.5 Saving the Hardware Configuration and Generating Code

The created hardware configuration must be saved, and code must be generated. This is done with a single command.

To save the hardware configuration and generate code:



- In the Hardware Configurator, click on the **Save** button
- or
- select **File** → **Save Workspace** to save the hardware configuration.
A hardware configuration file (* .hwz) is written to the working directory specified in the Hardware options (see section 2.1.1 on page 13).
- Code is generated for the hardware configuration; information, warning and error messages are shown in the ASCET monitor window.

If no error occurred, the `HardwareConfiguration` component in the project editor is updated, and the hardware processes are assigned to the respective tasks.

You can export the hardware configuration to a file, see page 77.

9.3.6 Final Actions

The hardware configuration is complete, and you can close the Hardware Configurator. The next step is to start the regular code generation for the experimental target from within the project editor.

Tip

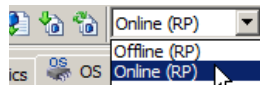
It is not recommended to include the hardware configuration in the graphical display of the Project Editor. Since this module can change with each hardware configuration code generation process, it could result in an unfavorable representation.

To generate code for the experimental target:

- In the project editor, select **Build** → **Build** to generate code for the entire project.
- Select **View** → **View Generated Code** to look at the generated code.

To experiment online:

For a meaningful experiment, you must connect sensors to the ES1303 inputs.



- Switch on the power supply of the ES1000.x.
- In the project editor, select `Online (RP)` from the "Experiment Target" combo box.

`Offline (RP)` is intended for offline experiments on the Target.

- Select **Build** → **Open Experiment**.
- In the Environment Browser, select the `Tutorial` environment.
- In the "Physical Experiment" window, select **Experiment** → **Start OS**.
- In the "Physical Experiment" window, select **Experiment** → **Start Measurement**.

The analog measured values of the ES1303 are displayed on an ASCET oscilloscope.

9.4 Tutorial – ES1325 (without Trigger)

The ES1325 sample project without using the trigger contains models of all important applications of the board. The models can be found in the `ES1325_InOut` module; they are shown in graphic hierarchies to keep the block diagram clear.

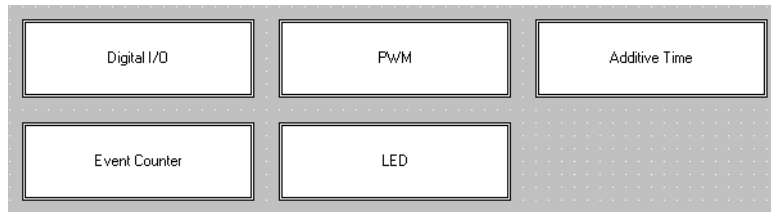


Fig. 9-4 ES1325 – Model Overview

Digital I/O: Input channel 3 is read in and transferred to output channel 3. The signal is inverted by assigning the "active" state to the low level on reading, whereas on sending, the high level is assigned the "active" state (see "Selecting levels" on page 104).

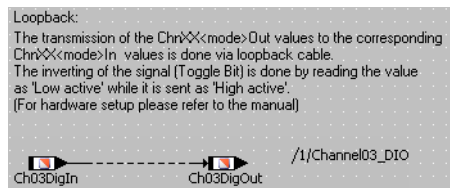


Fig. 9-5 ES1325 – Digital I/O

PWM: Channel 4 has a static PWM signal with fixed active and inactive times. The `InputValueConverter` class uses the values received from the `ActiveTime` and `InactiveTime` hardware input signals to calculate the duty cycle of channel 4.

Channel 5 has a dynamic PWM signal whose duty cycle is determined by a sawtooth signal (in the `DutyCycle` hierarchy). The `OutputValueConverter` class calculates the active time and inactive time of channel 5 and provides the values to be sent to the `ActiveTime` and `InactiveTime` hardware output signals..

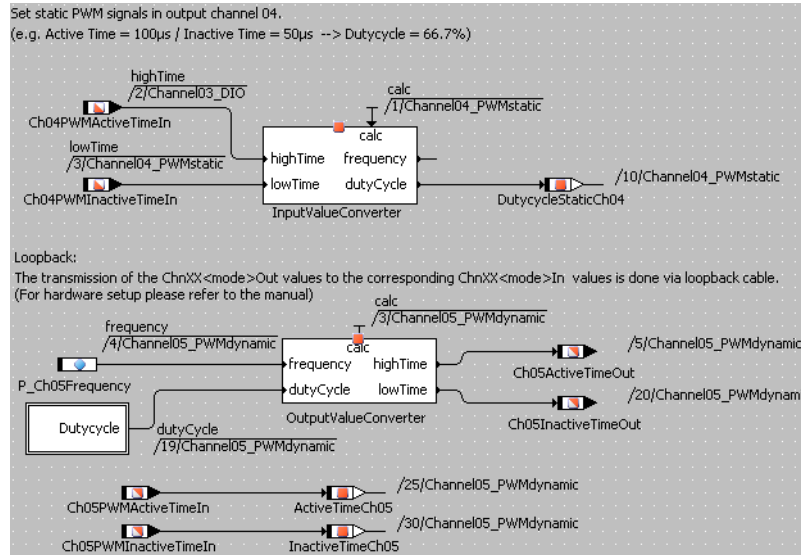


Fig. 9-6 ES1325 – PWM

Additive Time: The output signals of channel 5 are also transmitted by output channel 6. In this case it is useful for additive time measuring, i.e. the totaling of the active phases of the signal during task runtime.

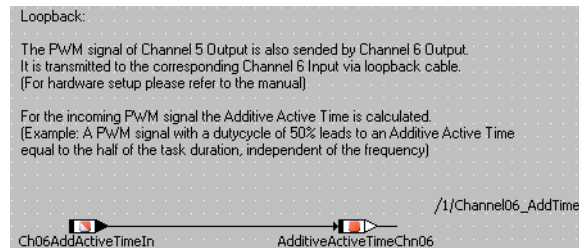


Fig. 9-7 ES1325 – Additive Time

Event Counter: Channel 7 counts every rising edge of an event; channel 8 counts every edge of an event. Both channels are controlled by the P_Ch0708DigOut parameter.

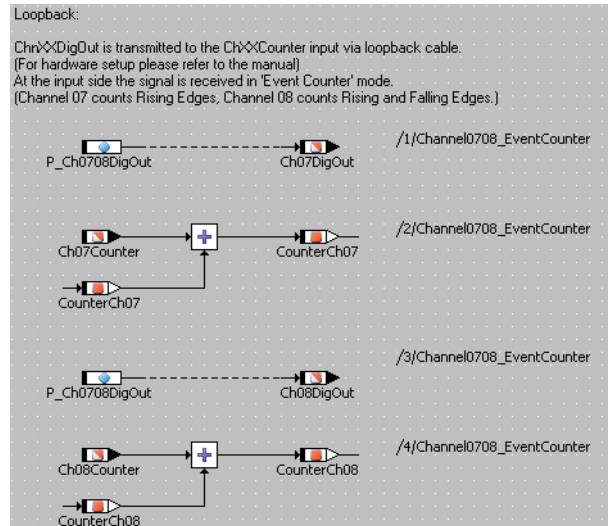


Fig. 9-8 ES1325 – Event Counter

LED: This is where the input signals for LEDs 1 to 3 are set. LED1 is controlled by a parameter of type `log`, LED2 via a parameter of type `sdisc`, and LED3 via a parameter of type `cont`.

Tip

*cont and sdisc values can be used to control the LEDs, but we **do not recommend** this at all. In the interests of good modeling/programming style, mapping to a logical variable, e.g. with case differentiation, is to be preferred.*

LED4 is controlled via the Ch03DigOut message.

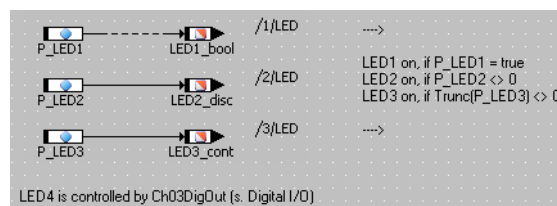
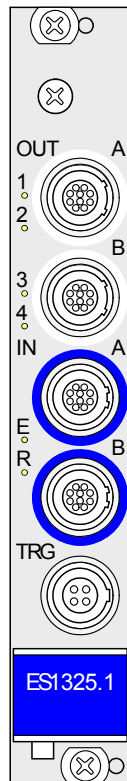


Fig. 9-9 ES1325 – LED

9.4.1 The ES1325 Board

The following figure shows the front panel of the ES1325 Board:



Ports Out-A and OUT-B each contain 8 output channels; ports IN-A and IN-B each contain 8 input channels. TRG contains 2 trigger inputs. LEDs 1 – 4 are available for display purposes.

Connections

To ensure that the experiment works, make sure you have the right wiring. In the tutorial you will be using ports Out-A, IN-A and (in section 9.5) TRG; the wiring specified here applies to both tasks on the ES1325.

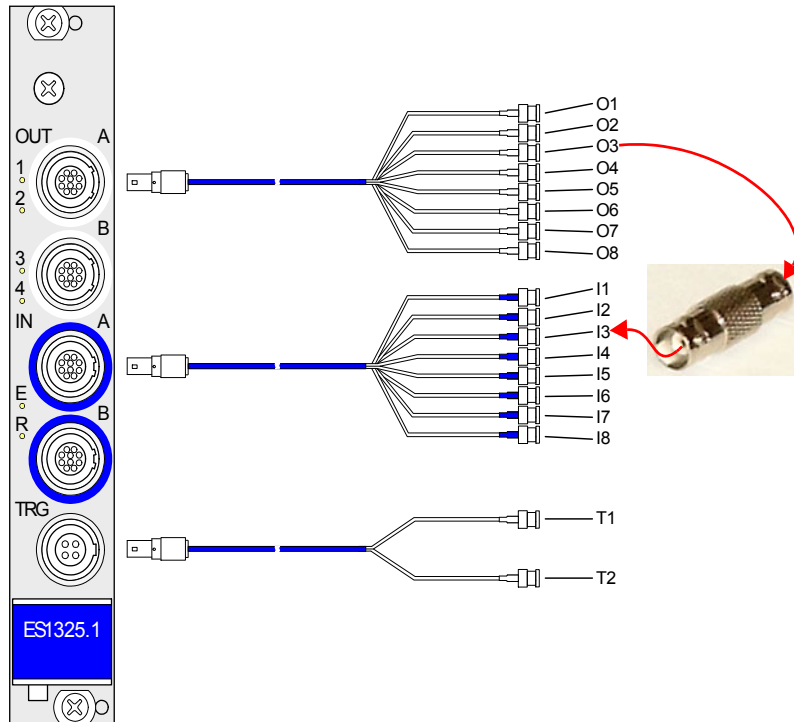


Fig. 9-10 Wiring of the ES1325

The following table specifies which outputs are connected with which inputs. Inputs I1 and I2 remain empty.

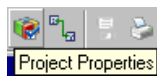
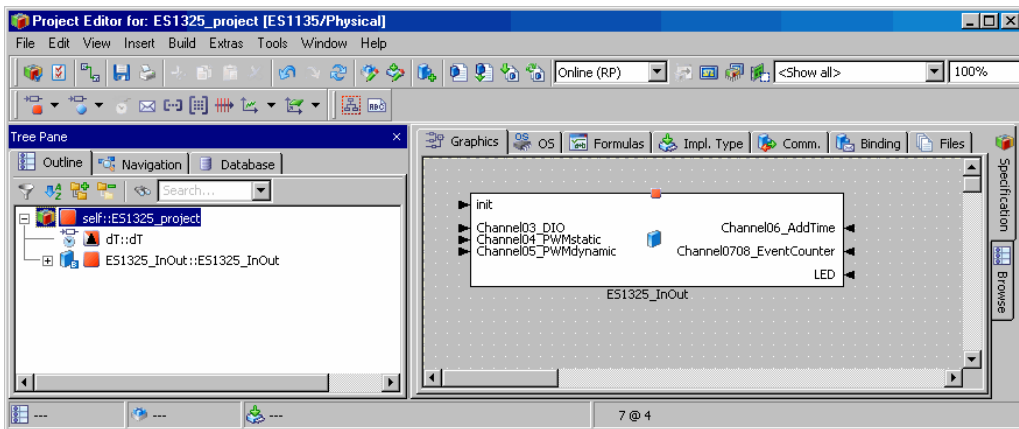
Output	O1	O2	O3	O4	O5	O6	O7	O8
Input			I3	I4	I5	I6	I7	I8
Trigger	T1	T2						

9.4.2 Sample Project

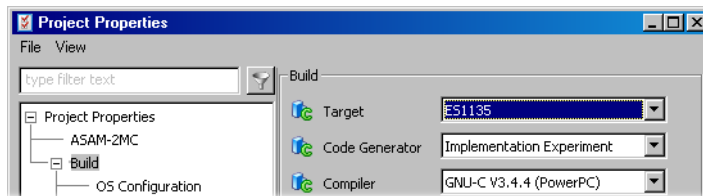
To open the exercise example:

- In the Component Manager, select the ... \RTIOTutorial\ES1325 folder.

- Open the ES1325_project project.



- Click the **Project Properties** button.
The "Project Properties" window opens.



- In the "Build" node, select the options
Target: ES1135 or ES1130,
Compiler: GNU-C V3.4.4 (PowerPC),
Operating System: ERCOSEK 4.3.

Tip

Only messages declared as "Exported" are available for communication.

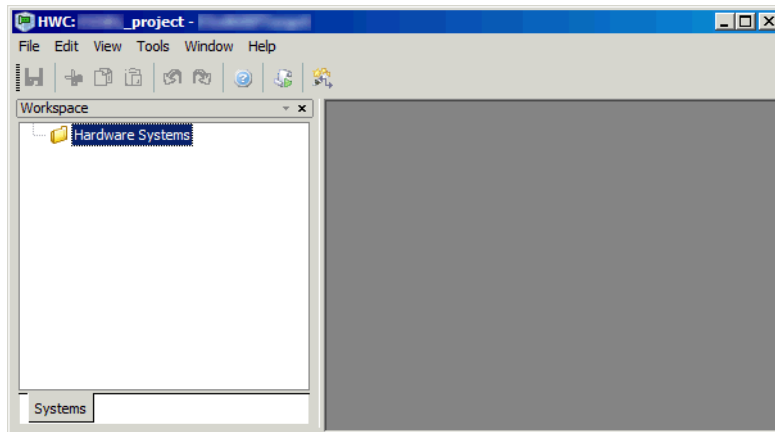
9.4.3 Creating the Hardware Configuration

To start the hardware configuration:

- Create an empty hardware configuration and add it to ES1325_project (see also page 65).

- In the "Outline" tab of the project editor, double-click the `HardwareConfiguration`.

The Hardware Configurator opens. The title bar contains the project name and the selected experimental target. The `Hardware Systems` folder always exists, it forms the root of the hardware configuration tree.

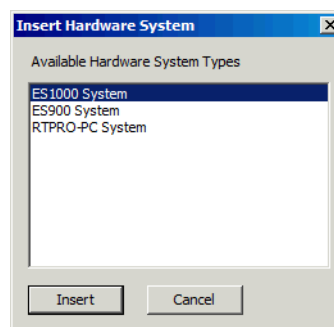


To specify the hardware configuration:

The hardware is described as a tree structure in the Workspace Browser.

- In the Hardware Configurator, right-click the `Hardware Systems` folder and select **Add Hardware System** from the context menu.

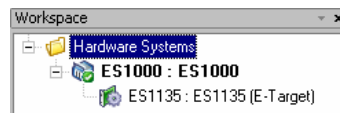
The "Insert Hardware System" window opens. This window lists the available hardware systems.



- Select the `ES1000` system entry.
- Click **Insert**.
- In the "Rename" dialog window, enter a name (default: `ES1000`) and click **OK**.

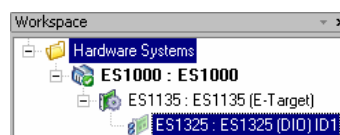
The hardware system is added below the `Hardware Systems` folder.

- Right-click the `Hardware Systems` folder and select **Insert Target** from the context menu.
The "Insert New Item" window opens. This window lists the items that can be inserted at the given level.
- In the "Available Items" list, select `ES1135` or `ES1130`.
Make sure that you select the target that is selected in the "Build" node of the Project Properties (cf. page 94).
- Click **Insert**.
- In the "Rename" dialog window, enter a name and click **OK**.
The target is added below the hardware system.



To integrate the ES1325:

- Right-click the target and select **Insert** from the context menu.
The "Insert New Item" window opens.
- Select the `ES1325 (DIO)` entry.
This entry describes the ES1325 interfaces.
- Click **Insert**.
- In the "Rename" dialog window, enter a name (default: `ES1325`) and click **OK**.
The `ES1325` board is added.

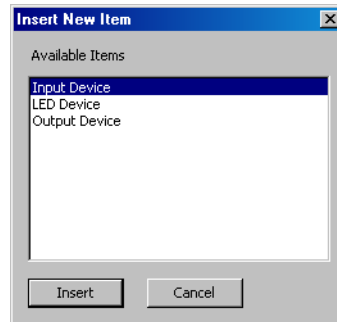


- Double-click the `ES1325` board.
The editor for the board opens. It lists all board parameters.
The board number is automatically set in the `Board ID` option. As the `ES1325` item is the first of this type, `ID1` is set.
For this task, you will use the default settings.

To create devices:

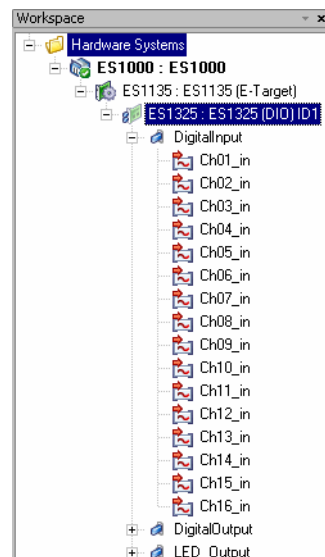
- Right-click the ES1325 board and select **Insert** from the context menu.

The "Insert New Item" window opens. It lists all ES1325 devices that can be added to the board.



- Select **Input Device** and click **Insert**.
This is used to describe the input channels of the ES1325.
- In the "Rename" dialog window, enter a name (default: DigitalInput) and click **OK**.
The DigitalInput device is added. Below the device, the signal groups are added automatically.
- Next, add the LED device and the Output device.

The item tree for the description of the sample system is now completely specified.



9.4.4 Configuring the Devices of the ES1325

The inputs and outputs as well as the LEDs now have to be configured. The outputs provide the signals which the inputs receive. The LEDs are used for display purposes.

Outputs – DigitalOutput Device

Below the DigitalOutput device, one signal group is created automatically for each output channel (Ch<n>_out, <n> = 01 – 16). Ch01_out – Ch08_out belong to the Out_A port; you will use only the signal groups Ch03_out – Ch08_out.

Signal Group Configuration: For each signal group, you select the channel mode (and with that, the available signals), assign a task, and make other necessary settings.

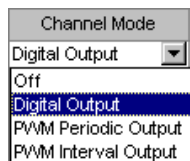
To configure the output signal groups:

- Double-click the DigitalOutput device.
The editor for the device opens.
- In the device editor, go to the "Signal Groups" tab.
This tab lists all signal groups and their parameters.

	Device	Name	Task	Direction	HW Channel	Channel Mode	Use HW Trigger	Active State	Permit Adjustment
1	DigitalOutput	Ch01_out		send	Ch01	Off			
2	DigitalOutput	Ch02_out		send	Ch02	Off			
3	DigitalOutput	Ch03_out		send	Ch03	Off			
4	DigitalOutput	Ch04_out		send	Ch04	Off			
5	DigitalOutput	Ch05_out		send	Ch05	Off			
6	DigitalOutput	Ch06_out		send	Ch06	Off			
7	DigitalOutput	Ch07_out		send	Ch07	Off			
8	DigitalOutput	Ch08_out		send	Ch08	Off			

1. Selecting the channel mode

- Click the "Channel Mode" column in the relevant row.
The table cell changes to a combo box.



- Select the following modes for the different groups:

Group	Channel Mode
Ch03_out, Ch07_out, Ch08_out	Digital Output
Ch04_out, Ch05_out, Ch06_out	PWM Periodic Output

In the "Active State" column, the default values are entered automatically. As all groups use the default setting High, you do not have to make any changes here.

The necessary signals for the channel modes are created automatically.

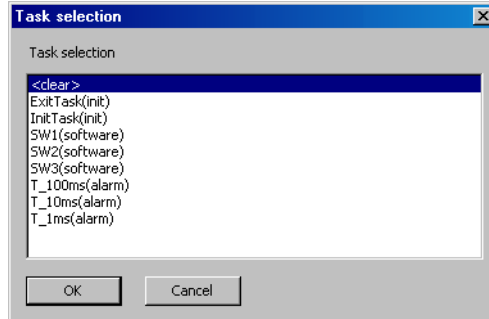
2. Assigning tasks

In the "Task" column, you can now assign to each group a task in which signal transfer is to take place.

	Device	Name	Task
3	DigitalOutput	Ch04_out	
4	DigitalOutput	Ch05_out	

- In the row of the signal group you want to edit, click the button in the "Task" column.

The "Task selection" window opens. It lists all tasks of the project that can be selected.



- Select the following tasks for data transfer in the task selection list.

Group	Task
Ch03_out, Ch04_out	T_100ms
Ch05_out, Ch06_out, Ch07_out, Ch08_out	T_10ms

- Click **OK**.

The selected tasks are displayed in the "Task" column.

The "Signal Groups" tab should now look as follows:

	Device	Name	Task	Direction	HW Channel	Channel Mode	Use HW Trigger	Active State	Permit	Adjustment
1	DigitalOutput	Ch01_out		send	Ch01	Off				
2	DigitalOutput	Ch02_out		send	Ch02	Off				
3	DigitalOutput	Ch03_out	T_100ms	send	Ch03	Digital Output		High		
4	DigitalOutput	Ch04_out	T_100ms	send	Ch04	PWM Periodic Output		High		
5	DigitalOutput	Ch05_out	T_10ms	send	Ch05	PWM Periodic Output		High		
6	DigitalOutput	Ch06_out	T_10ms	send	Ch06	PWM Periodic Output		High		
7	DigitalOutput	Ch07_out	T_10ms	send	Ch07	Digital Output		High		
8	DigitalOutput	Ch08_out	T_10ms	send	Ch08	Digital Output		High		

Signal Configuration: The signals that were created during signal group configuration are contained in the "Signals" tab.

	Device	Group	Direction	HW Ch	Name	Message	Signal Type	Unit	Formula	Min	Max	Init Val	Signal Mode	Pin
1	DigitalO	Ch03_out	send	Ch03	State_Ch03		boolean	..	f(ph)			0	State	Out A / 4
2	DigitalO	Ch04_out	send	Ch04	ActiveTime		cont	µs	f(ph)	0	107374182375	0	Active Time	Out A / 5
3	DigitalO	Ch04_out	send	Ch04	InactiveTime		cont	µs	f(ph)	0	107374182375	0	Inactive Time	Out A / 5
4	DigitalO	Ch05_out	send	Ch05	ActiveTime		cont	µs	f(ph)	0	107374182375	0	Active Time	Out A / 6
5	DigitalO	Ch05_out	send	Ch05	InactiveTime		cont	µs	f(ph)	0	107374182375	0	Inactive Time	Out A / 6
6	DigitalO	Ch06_out	send	Ch06	ActiveTime		cont	µs	f(ph)	0	107374182375	0	Active Time	Out A / 7
7	DigitalO	Ch06_out	send	Ch06	InactiveTime		cont	µs	f(ph)	0	107374182375	0	Inactive Time	Out A / 7
8	DigitalO	Ch07_out	send	Ch07	State_Ch07		boolean	..	f(ph)			0	State	Out A / 9
9	DigitalO	Ch08_out	send	Ch08	State_Ch08		boolean	..	f(ph)			0	State	Out A / 10

To rename signals:

The signals for CH04 – Ch06 have identical names. For clarity, you will rename them.

- In the second row (Ch04, ActiveTime), double-click in the "Name" column.
The table cell becomes an edit field.
- Change the signal name to Ch04out_ActiveTime.
- Rename the other signals as follows.

Group	New signal name
Ch04_out	Ch04out_ActiveTime
Ch04_out	Ch04out_InactiveTime
Ch05_out	Ch05out_ActiveTime
Ch05_out	Ch05out_InactiveTime
Ch06_out	Ch06out_ActiveTime
Ch06_out	Ch06out_InactiveTime

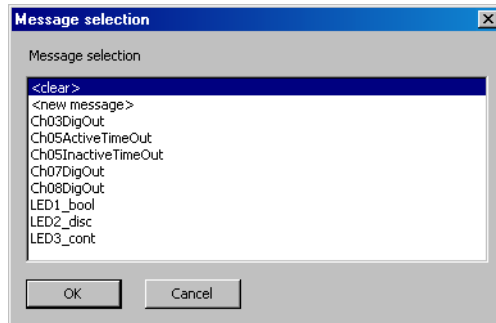
During signal mapping, ASCET messages from the project are assigned to the signals of the DigitalOutput signal group. A selection dialog can be opened via the "Message" parameter of each signal. This dialog contains only messages which have the attribute *Exported* and correspond to the transfer direction of the signal group (here: *send*); i.e.:

- Direction = send → send messages

To assign ASCET messages manually:

Name	Message
State_Ch03	
Ch04out_ActiveTime	

- Click the button in the "Message" column.
The "Message selection" window opens. It lists all messages in the project that can be selected.



- Select the following ASCET messages for the signals.

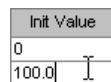
Signal	ASCET message
State_Ch03	Ch03DigOut
Ch05out_ActiveTime, Ch06out_ActiveTime	Ch05ActiveTimeOut
Ch05out_InactiveTime, Ch06out_InactiveTime	Ch05InactiveTimeOut
State_Ch07	Ch07DigOut
State_Ch08	Ch08DigOut

- Click **OK**.

To enter a value manually:

No message is assigned to the signals Ch04out_ActiveTime and Ch04out_InactiveTime. Instead, fixed values are assigned manually in the "Init Value" column.

- Double-click the "Init Value" column in the relevant row.



- The table cell becomes an edit field.
- Enter the following values.

Signal	Init Value
Ch04out_ActiveTime	100.0
Ch04out_InactiveTime	50.0

The "Signals" tab should now look as follows. Some columns have been minimized for better readability.

D	C	s	HW C	Name	Message	Signal T	Unit	For	Min	Max	Init Value	Signal Mode	P
1	D	C	s	Ch03	State_Ch03	Ch03DigOut	boolean	..			0	State	
2	D	C	s	Ch04	Ch04out_ActiveTime		cont	µs	0	107374182375	100	Active Time	
3	D	C	s	Ch04	Ch04out_InactiveTime		cont	µs	0	107374182375	50	Inactive Time	
4	D	C	s	Ch05	Ch5out_ActiveTime	Ch05ActiveTimeOut	cont	µs	0	107374182375	0	Active Time	
5	D	C	s	Ch05	Ch5out_InactiveTime	Ch05InactiveTimeOut	cont	µs	0	107374182375	0	Inactive Time	
6	D	C	s	Ch06	Ch6out_ActiveTime	Ch05ActiveTimeOut	cont	µs	0	107374182375	0	Active Time	
7	D	C	s	Ch06	Ch6out_InactiveTime	Ch05InactiveTimeOut	cont	µs	0	107374182375	0	Inactive Time	
8	D	C	s	Ch07	State_Ch07	Ch07DigOut	boolean	..			0	State	
9	D	C	s	Ch08	State_Ch08	Ch08DigOut	boolean	..			0	State	

Inputs – DigitalInput Device

Below the DigitalInput device, one signal group is created automatically for each input channel (Ch<n>_in, <n> = 01 – 16). Ch01_in – Ch08_in belong to the In_A port; only the signal groups Ch03_in – Ch08_in are used in this tutorial.

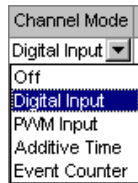
Signal Group Configuration: For each signal group, you select the channel mode (and with that, the available signals), assign a task, and make other necessary settings.

To configure the input signal groups:

- Double-click the DigitalInput device.
The editor for the device opens.
- In the device editor, go to the "Signal Groups" tab.
This tab lists all signal groups and their parameters.

Device	Name	Task	Direction	HW Ch	Channel Mode	Use HW Trigger	Polling	Active State	Significant Edge	Hysteresis	Low Thr
DigitalInput	Ch01_in		receive	Ch01	Off						
DigitalInput	Ch02_in		receive	Ch02	Off						
DigitalInput	Ch03_in		receive	Ch03	Off						
DigitalInput	Ch04_in		receive	Ch04	Off						
DigitalInput	Ch05_in		receive	Ch05	Off						
DigitalInput	Ch06_in		receive	Ch06	Off						
DigitalInput	Ch07_in		receive	Ch07	Off						
DigitalInput	Ch08_in		receive	Ch08	Off						

1. Selecting the channel mode



- In the "Channel Mode" column, select the following modes for the different groups:

Group	Channel Mode
Ch03_in	Digital Input
Ch04_in, Ch05_in	PWM Input
Ch06_in	Additive Time
Ch07_in, Ch08_in	Event Counter

The default values for the relevant modes are entered in the columns "Active State", "Significant Edge", "Hysteresis", "Low Threshold [V]" and "High Threshold [V]".

The necessary signals for the channel modes are created automatically.

2. Assigning tasks

In the "Task" column, you can now assign tasks in which signal transfer will take place.



- In the row of the signal group you want to edit, click the button in the "Task" column. The "Task selection" window opens. It lists all tasks of the project that can be selected.
- Select the following tasks in which data transfer is to take place from the task selection list.

Group	Task
Ch03_in, Ch04_in	T_100ms
Ch05_in, Ch06_in, Ch07_in, Ch08_in	T_10ms

- Click **OK**. The selected tasks are displayed in the "Task" column.

3. Selecting levels



- Click the "Active State" column in the relevant row to assign the active state to a level of the input signal. The table cell becomes a combo box.
- Select the following states.

Group	Active State
Ch03_in	Low
Ch04_in - Ch08_in	High

The selection is displayed in the "Active State" field.

4. Selecting an edge (only PWM Input and Event Counter modes)

- Click the "Significant Edge" column in the relevant row to assign an event to an edge.

The table cell becomes a combo box.



- Select the following options.

Group	Significant Edge
Ch04_in, Ch05_in, Ch07_in	Inactive-Active
Ch08_in	Both

The selection is displayed in the "Significant Edge" field.

The "Signal Groups" tab should now look as follows:

Device	Name	Task	Direction	HW Ch	Channel Mode	Us	Polling	Active State	Significant Edge	Hysteresis
DigitalInput	Ch01_in		receive	Ch01	Off					
DigitalInput	Ch02_in		receive	Ch02	Off					
DigitalInput	Ch03_in	T_100ms	receive	Ch03	Digital Input	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Low		TTL
DigitalInput	Ch04_in	T_100ms	receive	Ch04	PWM Input	<input type="checkbox"/>	<input checked="" type="checkbox"/>	High	Inactive-Active	TTL
DigitalInput	Ch05_in	T_10ms	receive	Ch05	PWM Input	<input type="checkbox"/>	<input checked="" type="checkbox"/>	High	Inactive-Active	TTL
DigitalInput	Ch06_in	T_10ms	receive	Ch06	Additive Time	<input type="checkbox"/>	<input checked="" type="checkbox"/>	High		TTL
DigitalInput	Ch07_in	T_10ms	receive	Ch07	Event Counter	<input type="checkbox"/>	<input checked="" type="checkbox"/>	High	Inactive-Active	TTL
DigitalInput	Ch08_in	T_10ms	receive	Ch08	Event Counter	<input type="checkbox"/>	<input checked="" type="checkbox"/>	High	Both	TTL

Signal Configuration: The signals that were created during signal group configuration are contained in the "Signals" tab.

Device	Group	Direction	HW Ch	Name	Message	Signal Type	Unit	Formula	Min	Max	Init Val	Signal Mode	Pin
DigitalIn	Ch03_in	receive	Ch03	State_Ch03		boolean	..	f(ph)				State	In A / 4
DigitalIn	Ch04_in	receive	Ch04	ActiveTime		cont	µs	f(ph)				Active Time	In A / 5
DigitalIn	Ch04_in	receive	Ch04	InactiveTime		cont	µs	f(ph)				Inactive Time	In A / 5
DigitalIn	Ch05_in	receive	Ch05	ActiveTime		cont	µs	f(ph)				Active Time	In A / 6
DigitalIn	Ch05_in	receive	Ch05	InactiveTime		cont	µs	f(ph)				Inactive Time	In A / 6
DigitalIn	Ch06_in	receive	Ch06	AdditiveActiveTime		cont	µs	f(ph)				Additive Active Time	In A / 7
DigitalIn	Ch07_in	receive	Ch07	CounterValue		udisc	..	f(ph)				Counter Value	In A / 9
DigitalIn	Ch08_in	receive	Ch08	CounterValue		udisc	..	f(ph)				Counter Value	In A / 10

To rename signals:

Some signals have identical names. For clarity, you will rename them.

- In the relevant row, double-click in the "Name" column. The table cell becomes an edit field.
- Change the signal names as follows.

Group	new signal name
Ch04_in	Ch04in_ActiveTime
Ch04_in	Ch04in_InactiveTime
Ch05_in	Ch05in_ActiveTime

Group	new signal name
Ch05_in	Ch05in_InactiveTime
Ch07_in	Ch07in_CounterValue
Ch08_in	Ch08in_CounterValue

During signal mapping, ASCET messages from the project are assigned to the signals of the DigitalInput signal group. A selection dialog can be opened via the "Message" parameter of each signal. This dialog contains only messages which have the attribute *Exported* and correspond to the transfer direction of the signal group (here: *receive*); i.e.:

- Direction = receive → receive messages

To assign ASCET messages manually:

- Click the button in the "Message" column. The "Message selection" window opens. It lists all messages in the project that can be selected.
- Select the following ASCET messages for the signals.

Signal

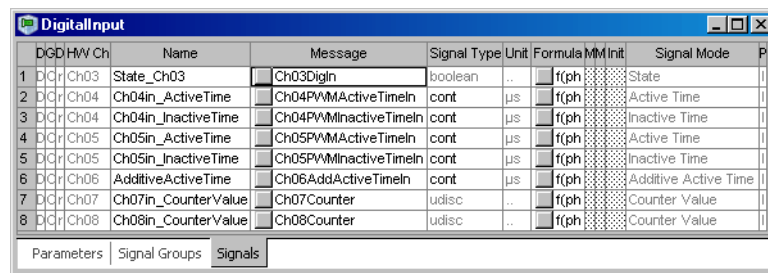
State_Ch03
 Ch04in_ActiveTime
 Ch04in_InactiveTime
 Ch05in_ActiveTime
 Ch05in_InactiveTime
 AdditiveActiveTime
 Ch07in_CounterValue
 Ch08in_CounterValue

ASCET Message

Ch03DigIn
 Ch04PWMActiveTimeIn
 Ch04PWMInctiveTimeIn
 Ch05PWMActiveTimeIn
 Ch05PWMInactiveTimeIn
 Ch06AddActiveTimeIn
 Ch07Counter
 Ch08Counter

- Click **OK**.

The "Signals" tab should now look as follows. Some columns have been minimized for better readability.



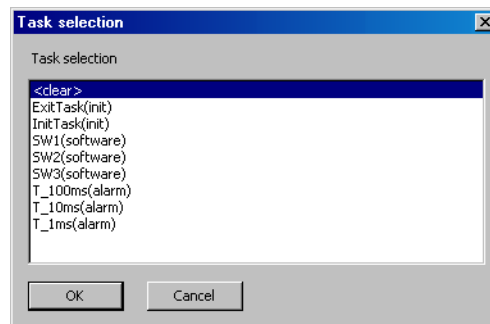
LEDs – LED_output Device

Below the LED_output device, the signal group LED_out and the signals LEDState1 – LEDState4 for the four LEDs are created automatically.

Signal Group Configuration: The only required setting for the `LED_out` signal group is a task assignment.

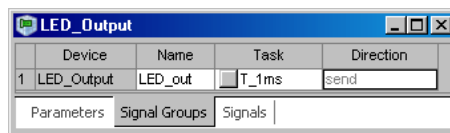
To configure the LED signal group:

- Double-click the `LED_Output` device.
The editor for the device opens.
- In the device editor, go to the "Signal Groups" tab.
- Click the button in the "Task" column.
The "Task selection" window opens. It lists all tasks of the project that can be selected.



- Select the `T_1ms (Alarm)` task from the task selection list and click **OK**.
The selected task is displayed in the "Task" column.

The "Signal Groups" tab should now look as follows:



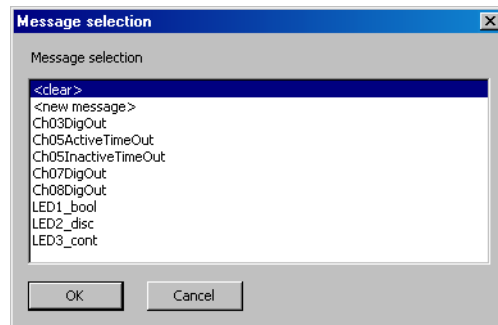
Signal Configuration: This tab contains the signals which belong to the `LED_out` signal group. There is one signal for each LED on the board.

During signal mapping, ASCET messages from the project are assigned to the signals of the `LED_out` signal group. A selection dialog can be opened via the "Message" parameter of each signal. This dialog only contains messages which have the attribute *Exported* and correspond to the transfer direction of the signal group (here: *send*); i.e.:

- Direction = send → send messages

To assign ASCET messages manually:

- Click the button in the "Message" column.
The "Message selection" window opens. It lists all messages in the project that can be selected.



- Select the following ASCET messages for the signals.

Signal	ASCET Message
LEDState1	LED1_bool
LEDState2	LED2_disc
LEDState3	LED3_cont
LEDState4	Ch03DigOut

- Click **OK**.

The "Signals" tab should now look as follows. Some columns have been minimized for better readability.

	Device	Group	Direction	Name	Message	Signal Type	Unit	Formula	MM	Init Value	Signal Mode	Pin
1	LED_Output	LED_out	send	LEDState1	LED1_bool	boolean	..	f(ph)		0	LED State	LED 1
2	LED_Output	LED_out	send	LEDState2	LED2_disc	boolean	..	f(ph)		0	LED State	LED 2
3	LED_Output	LED_out	send	LEDState3	LED3_cont	boolean	..	f(ph)		0	LED State	LED 3
4	LED_Output	LED_out	send	LEDState4	Ch03DigOut	boolean	..	f(ph)		0	LED State	LED 4

9.4.5 Saving the Hardware Configuration and Generating Code

The created hardware configuration must be saved, and code must be generated. This is done with a single command.

To save the hardware configuration and generate code:



- In the Hardware Configurator, click on the **Save** button

or

- select **File** → **Save Workspace** to save the hardware configuration.

A hardware configuration file (*.hwx) is written to the working directory specified in the Hardware options (see section 2.1.1 on page 13).

Code is generated for the hardware configuration; information, warning and error messages are shown in the ASCET monitor window.

If no error occurred, the HardwareConfiguration component in the project editor is updated, and the hardware processes are assigned to the respective tasks.

You can export the hardware configuration to a file, see page 77.

9.4.6 Experimenting with the Sample Project

The hardware configuration is complete, and you can close the Hardware Configurator. The next step is to start the regular code generation for the experimental target from within the project editor.

Tip

It is not recommended to include the hardware configuration in the graphical display of the Project Editor. Since this module can change with each hardware configuration code generation process, it could result in an unfavorable representation.

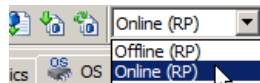
To generate code for the experimental target:

- In the Project Editor, select **Build** → **Build** to generate code for the entire project.
- Select **Build** → **View Generated Code** to view the generated code.

To experiment online:

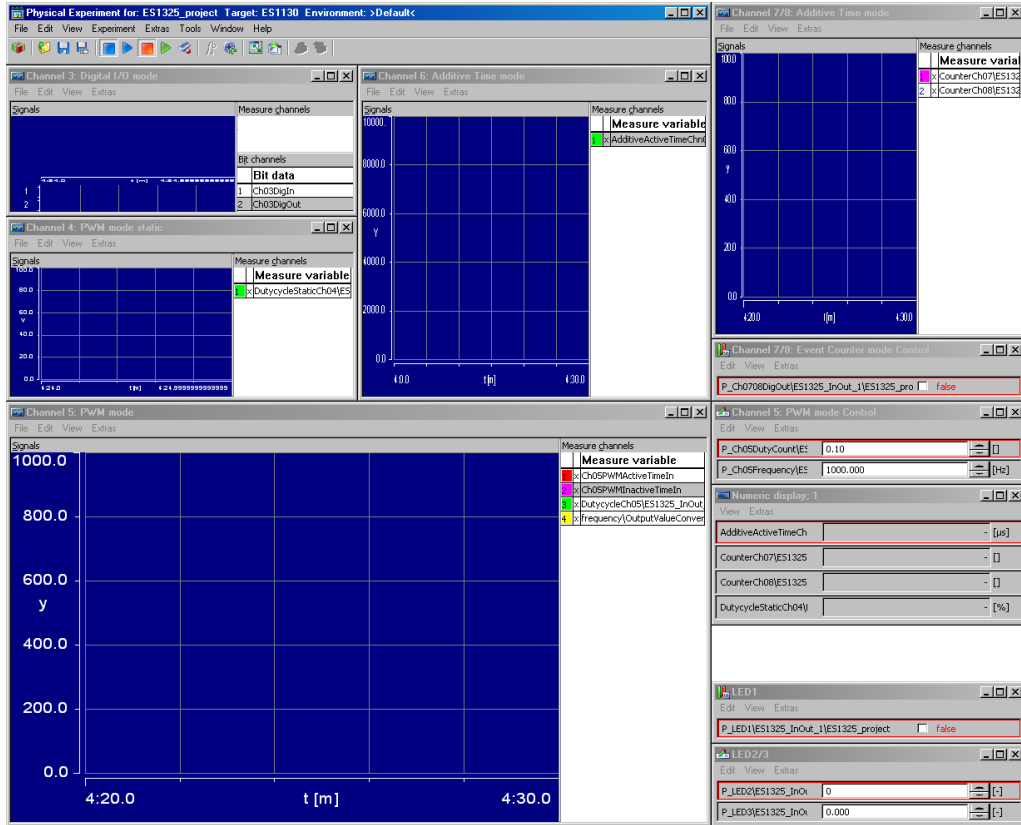
- Connect the inputs and outputs of the ES1325 according to section "Connections" on page 94.
- Switch on the power supply of the ES1000.x.
- In the project editor, select **Online (RP)** from the "Experiment Target" combo box.

Offline (RP) is intended for offline experiments on the Target.



- Select **Build** → **Experiment**.

The "Physical Experiment" window and the pre-defined experiment environment consisting of five oscilloscopes, a numeric display and four calibration windows open.



- Select **Experiment** → **Start OS** in the "Physical Experiment" window

or

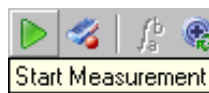


- click the **Start OS** button.

The operating system is launched; the model is run.

- Select **Experiment** → **Start Measurement** in the "Physical Experiment" window

or

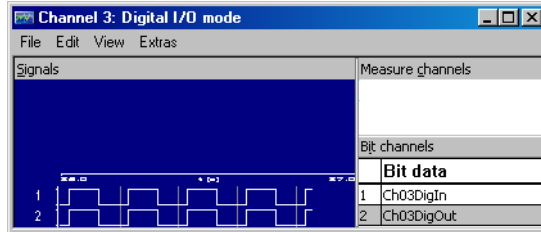


- click the **Start Measurement** button.

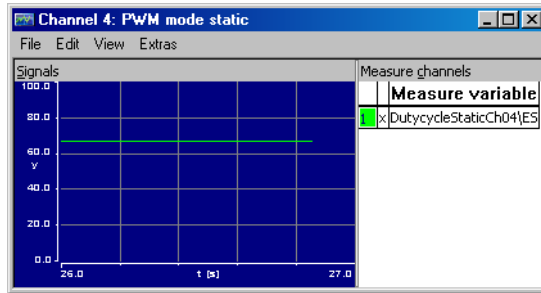
The values of the ASCET messages are displayed in the oscilloscopes and in the numerical display.

The displays and calibration possibilities of the individual model blocks are described below.

Digital I/O: The "Channel 3: Digital I/O mode" oscilloscope shows the messages Ch03DigIn and Ch03DigOut.

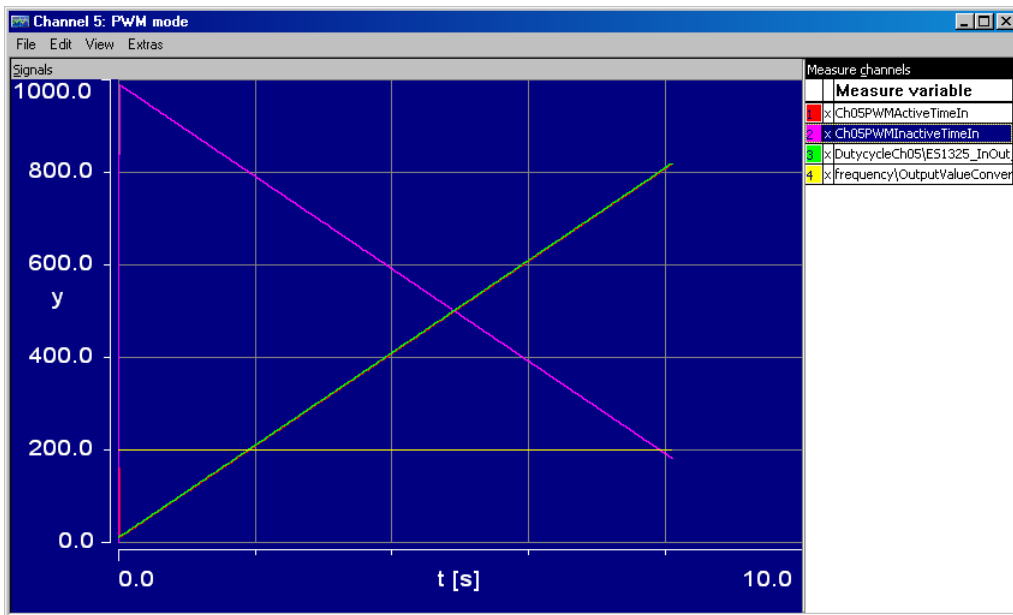


PWM: The "Channel 4: PWM mode static" oscilloscope shows the DutycycleStaticCh04 variable which contains the duty cycle of the PWM signal, calculated from the ActiveTime and InactiveTime signals (see Fig. 9-6 on page 91). As active and inactive time are both specified as fixed values, the value of DutycycleStaticCh04 is constant.



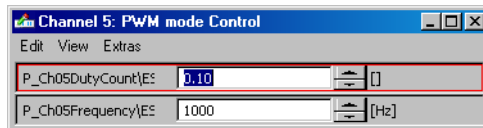
The "Channel 5: PWM mode" oscilloscope shows the input messages Ch05PWMActiveTimeIn (red curve) and Ch05PWMinactiveTimeIn (violet curve) which contain the active and inactive time of the PWM signal. The variables DutyCycleCh05 (green curve) and frequency (yellow curve) are displayed which contain the frequency (in Hz) and duty cycle (in %) of the PWM signal. These variables are used to compute the values of the output messages Ch05ActiveTimeOut and Ch05ActiveTimeOut (see Fig. 9-6 on page 91).

Each measure channel in the oscilloscope has its own y axis settings.

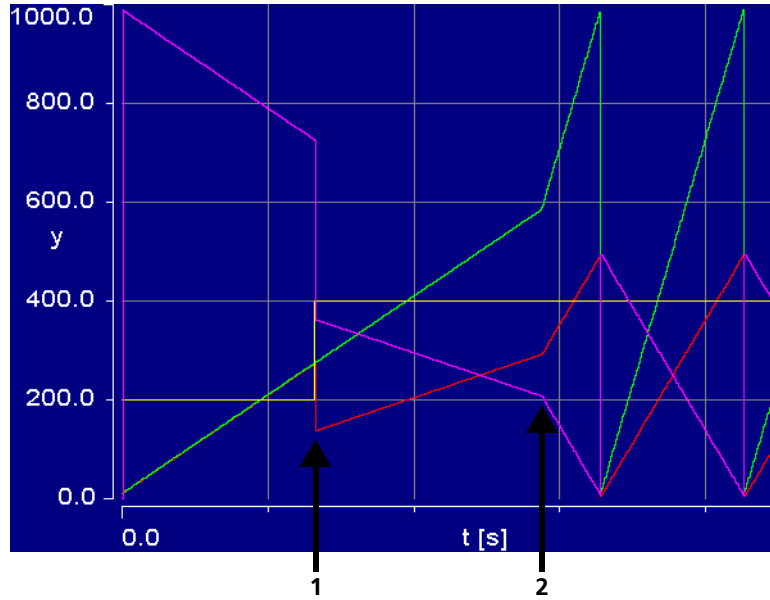


If you modify the `P_Ch05Frequency` parameter at the bottom of the "Channel 5: PWM mode Control" calibration window, the frequency changes and thus the active and inactive time. The duty cycle remains the same.

If you modify the parameter `P_Ch05DutyCount` at the top of the "Channel 5: PWM mode Control" calibration window, the course of the duty cycle changes and thus the active and inactive time. The frequency remains the same.

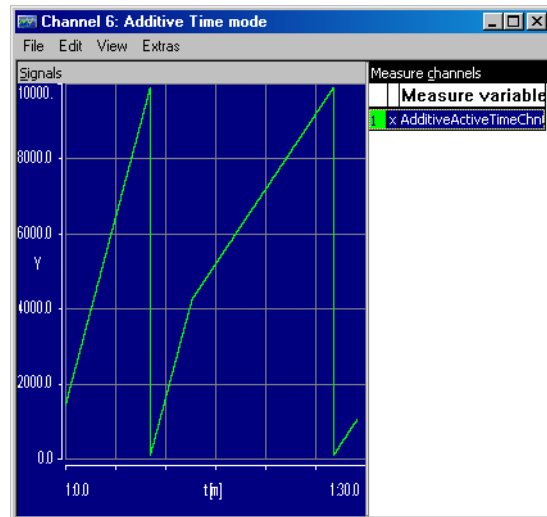


At point 1 in the following diagram, a frequency of 1000 Hz was set to 2000 Hz so that the curves for Ch05PWMActiveTimeIn, Ch05PWMinactiveTimeIn and frequency show discontinuity. P_Ch05DutyCount was not changed; the curve for DutyCycleCh05 retains its slope.

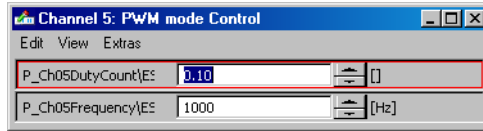


At point 2, the P_Ch05DutyCount parameter was set from 0.1 to 0.5 so that the curves for DutyCycleCh05, Ch05PWMActiveTimeIn and Ch05PWMinactiveTimeIn break off. The frequency remains unchanged at this point.

Additive Time: The "Channel 6: Additive Time mode" oscilloscope shows the AdditiveActiveTimeChn06 variable which accepts the additive active time of the signal.

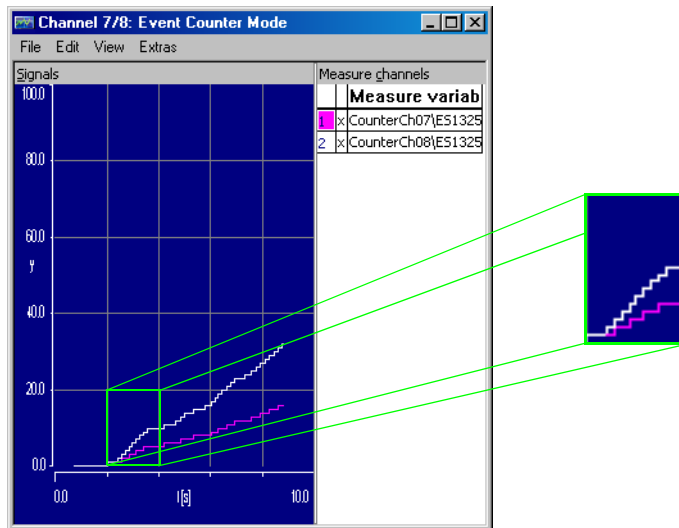


You can influence the slope of the sawtooth by modifying the `P_Ch05DutyCount` parameter at the top of the "Channel 5: PWM mode Control" calibration window. A value of 0.1 was set for the first part of the curve shown, and for the second, a value of 0.04.

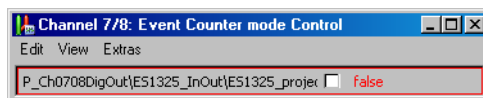


The `AdditiveActiveTimeChn06` signal is independent of the frequency set as the additive active time remains unaffected by it during task runtime.

Event Counter: The "Channel 7/8: Event Counter Mode" oscilloscope shows the variables `CounterCh07` (lower curve) and `CounterCh08` (upper curve) which receive event counts of the signal.

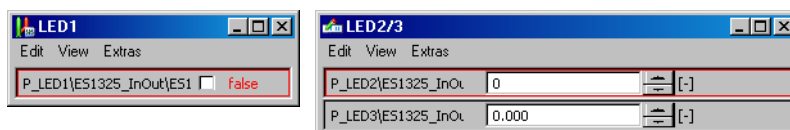


You can add an increment to both variables by setting the `P_Ch0708DigOut` parameter in the "Channel 7/8: Event Counter mode Control" calibration window alternately to `true` and `false`.



`CounterCh07` grows half as quickly as `CounterCh08` because channel 8 counts every change whereas channel 7 only counts changes from `false` to `true`.

LED: In the "LED1" and "LED2/3" windows, you control the LEDs on the front panel of the ES1325 via the parameters `P_LED1`, `P_LED2` and `P_LED3`.



LED1 lights up when `P_LED1` is `true`, LED2 lights up when `P_LED2` is not equal to 0, and LED3 lights up when the value of `P_LED3` is not equal to 0 after the decimal places are cut off.

9.5 Tutorial – ES1325 (with Trigger)

The ES1325 sample project using the trigger contains a model with a dynamic PWM signal (PWM block, see Fig. 9-12), whose duty cycle is determined by a saw-tooth signal.

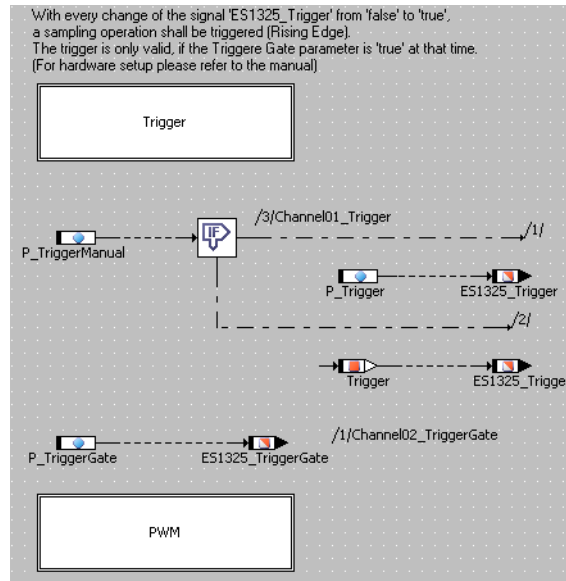


Fig. 9-11 ES1325 (with Trigger) – Model

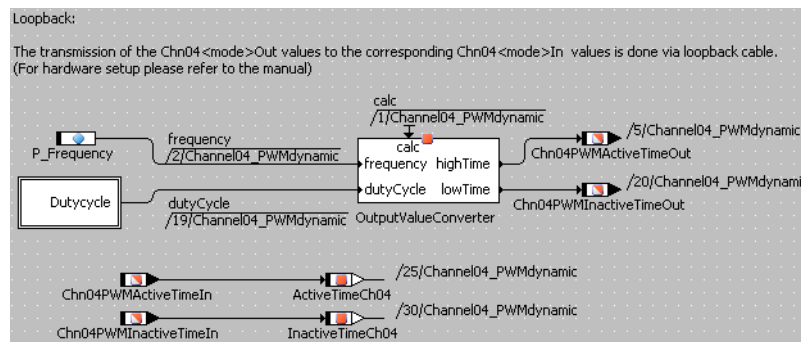


Fig. 9-12 ES1325 (with Trigger) – PWM

Frequency and duty cycle are used to compute active time and inactive time; these values are fed to the output signals. The output signals (messages `Chn04PWMActiveTimeOut` and `Chn04PWMInactiveTimeOut`) are sent continuously, but reading the input signals (`Chn04PWMActiveTimeIn` and `Chn04PWMInactiveTimeIn` messages) is controlled by a hardware trigger. This can either be triggered manually or automatically (`P_TriggerManual` parameter, see Fig. 9-11.).

With automatic triggering, the first trigger signal ES1325_Trigger in the Trigger block (Fig. 9-13) is calculated; with manual triggering, the trigger signal is determined via the P_Trigger parameter.

The second trigger signal ES1325_TriggerGate is always determined via the P_TriggerGate parameter (Fig. 9-11).

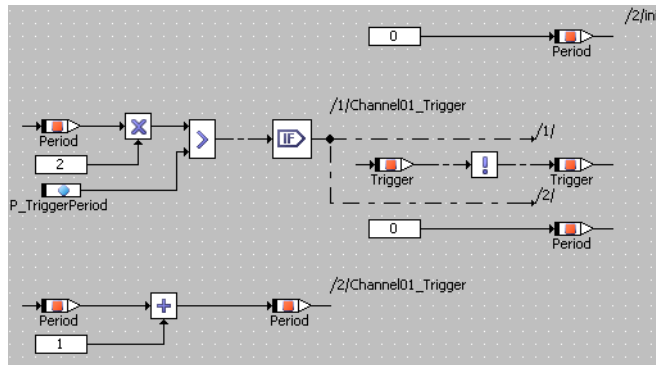


Fig. 9-13 ES1325 – Trigger

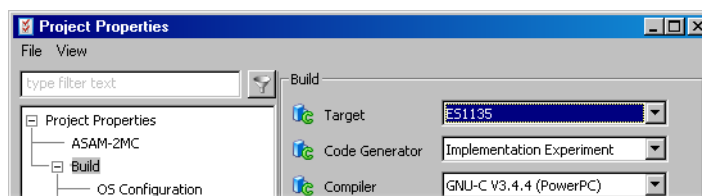
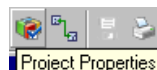
9.5.1 The ES1325 Board

For details on the board and the necessary wiring please refer to section 9.4.1 "The ES1325 Board" on page 93.

9.5.2 Sample Project

To open the exercise example:

- Select the `... \RTIOTutorial\ES1325_triggered` folder from the Component Manager.
 - Select the `ES1325_triggered_project` project.
 - Open the project.
 - Click the **Project Properties** button.
- The "Project Properties" window opens.



- In the "Build" node, select the options
 Target: ES1135 or ES1130,
 Compiler: GNU-C V3.4.4 (PowerPC),
 Operating System: ERCOSEK 4.3.

Tip

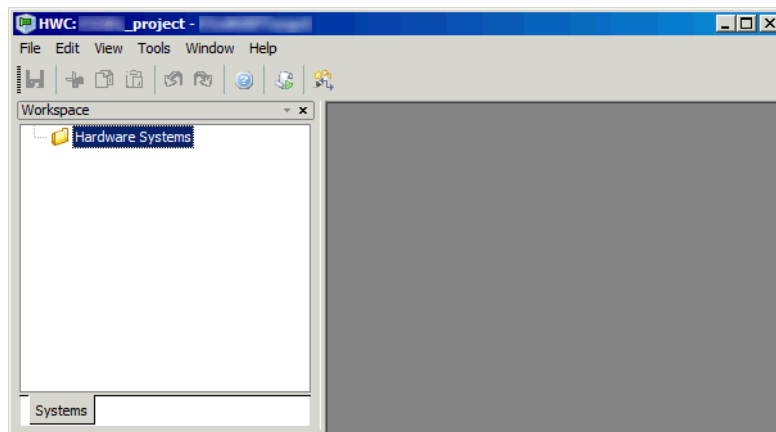
Only messages declared as "Exported" are available for communication.

9.5.3 Creating the Hardware Configuration

To start the hardware configuration:

- Create an empty hardware configuration and add it to ES1325_project (see also page 65).
- In the "Outline" tab of the project editor, double-click the HardwareConfiguration.

The Hardware Configurator opens. The title bar contains the project name and the selected experimental target. The Hardware Systems folder always exists, it forms the root of the hardware configuration tree.

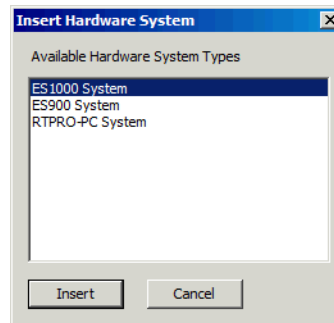


To specify the hardware configuration:

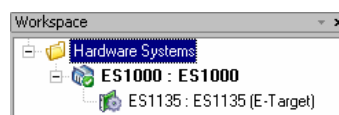
The hardware is described as a tree structure in the Workspace Browser.

- In the Hardware Configurator, right-click the `Hardware Systems` folder and select **Add Hardware System** from the context menu.

The "Insert Hardware System" window opens. This window lists the available hardware systems.



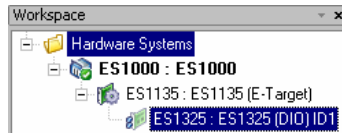
- Select the `ES1000` system entry.
- Click **Insert**.
- In the "Rename" dialog window, enter a name (default: `ES1000`) and click **OK**.
The hardware system is added below the `Hardware Systems` folder.
- Right-click the `Hardware Systems` folder and select **Insert Target** from the context menu.
The "Insert New Item" window opens. This window lists the items that can be inserted at the given level.
- In the "Available Items" list, select `ES1135` or `ES1130`.
Make sure that you select the target that is selected in the "Build" node of the Project Properties (cf. page 116).
- Click **Insert**.
- In the "Rename" dialog window, enter a name and click **OK**.
The target is added below the hardware system.



To integrate the ES1325:

- Right-click the target and select **Insert** from the context menu.
The "Insert New Item" window opens.
- Select the `ES1325 (DIO)` entry.
This entry describes the ES1325 interfaces.

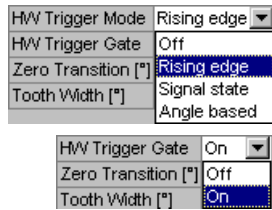
- Click **Insert**.
 - In the "Rename" dialog window, enter a name (default: ES1325) and click **OK**.
- The ES1325 board is added.



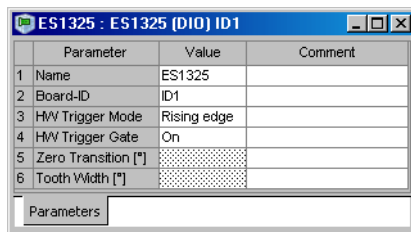
To configure the ES1325

You will use the ES1325 triggers. To configure trigger usage, proceed as follows.

- Double-click the ES1325 board.
The editor for the board opens. It lists all board parameters.
- In the board editor, click the "Value" column in the "HW Trigger Mode" row.
The table cell becomes a combo box.
- Select **Rising edge**.
- In the "Value" column in the "HW Trigger Gate" row, select **On**.
The hardware trigger is now determined.



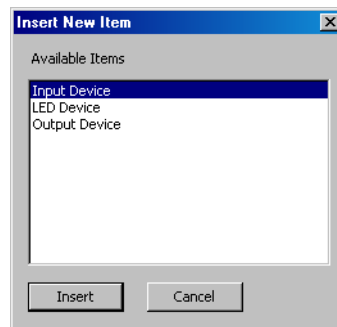
The board editor should now look as follows:



To create devices:

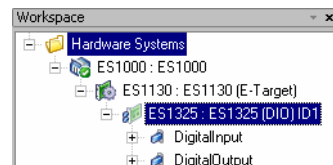
- Right-click the ES1325 board and select **Insert** from the context menu.

The "Insert New Item" window opens. It lists all ES1325 devices that can be added to the board.



- Add an **Input Device** and an **Output Device**.
- Click **OK**.

The item tree for the description of the sample system is now completely specified.



9.5.4 Configuring the Devices of the ES1325

The inputs and outputs now have to be configured. The outputs provide the signals which the inputs receive.

Outputs – ES1325-Output Device

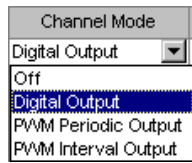
Below the `DigitalOutput` device, one signal group is created automatically for each output channel (`Ch<n>_out`, `<n> = 01 – 16`). `Ch01_out – Ch08_out` belong to the `Out_A` port; you will use only the signal groups `Ch01_out`, `Ch02_out`, and `Ch04_out`.

Signal Group Configuration: For each signal group, you select the channel mode (and with that, the available signals), assign a task, and make other necessary settings.

To configure the output signal groups:

- Double-click the `DigitalOutput` device.
The editor for the device opens.
- In the device editor, go to the "Signal Groups" tab.
This tab lists all signal groups and their parameters.

1. Selecting the channel mode



- Select the following modes for the different groups in the "Channel Mode" column:

Group	Channel Mode
Ch01_out, Ch02_out	Digital Output
Ch04_out	PWM Periodic Output

In the "Active State" column, the default values are entered automatically. As all groups use the default setting High, you do not have to make any changes here.

The necessary signals for the channel modes are created automatically.

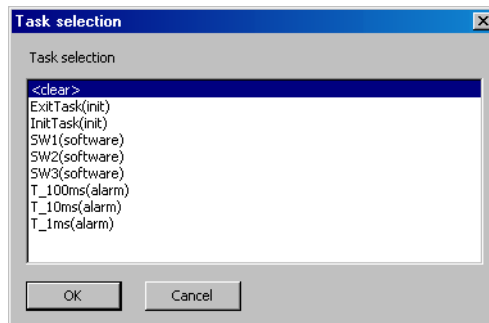
2. Assigning a task

In the "Task" column, you can now assign to each group a task in which signal transfer is to take place.

	Device	Name	Task
3	DigitalOutput	Ch04_out	
4	DigitalOutput	Ch05_out	

- In the row of the signal group you want to edit, click the button in the "Task" column.

The "Task selection" window opens. It lists all tasks of the project that can be selected.



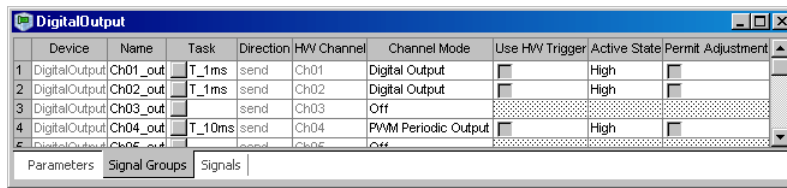
- Select the following tasks for data transfer from the task selection list.

Group	Task
Ch01_out, Ch02_out	T_1ms
Ch04_out	T_10ms

- Click **OK**.

The selected tasks are displayed in the "Task" column.

The "Signal Groups" tab should now look as follows.



Signal Configuration: The signals that were created during signal group configuration are contained in the "Signals" tab.

To rename signals:

The names of the signals for Ch04_out do not contain the channel number. For clarity, you will rename them.

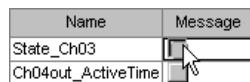
- In the rows with the signals of Ch04_out, double-click in the "Name" column.
- Rename the signals as follows.

Group	New signal name
Ch04_out	Ch04out_ActiveTime
Ch04_out	Ch05out_InactiveTime

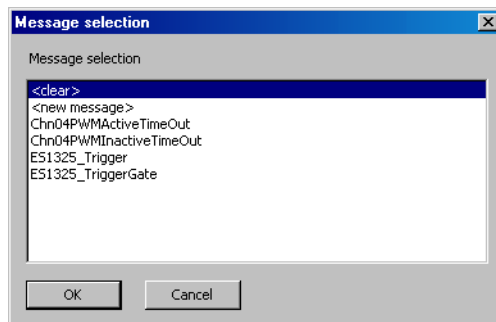
During signal mapping, ASCET messages from the project are assigned to the signals of the DigitalOutput signal group. A selection dialog can be opened via the "Message" parameter of each signal. This dialog contains only messages which have the attribute *Exported* and correspond to the transfer direction of the signal group (here: *send*); i.e.:

- Direction = send → send messages

To assign an ASCET message manually:



- Click the button in the "Message" column.
The "Message selection" window opens. It lists all messages in the project that can be selected.

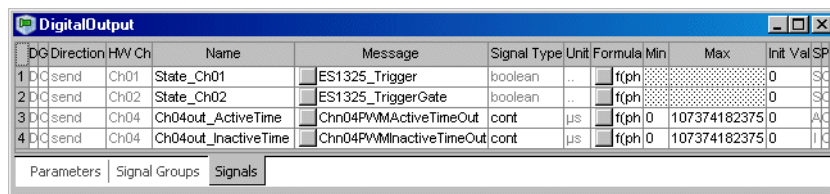


- Select the following ASCET messages for the signals.

Signal	ASCET message
State_Ch01	ES1325_Trigger
State_Ch02	ES1325_TriggerGate
Ch04out_ActiveTime	Chn04PWMActiveTimeOut
Ch04out_InactiveTime	Chn04PWMInactiveTimeOut

- Click **OK**.

Once you have assigned messages to all signals, the tab should look as follows:



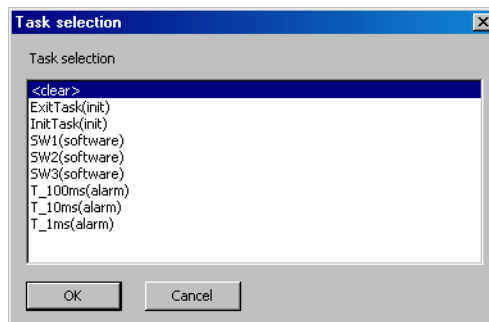
Inputs – ES1325- Input Device

Below the DigitalInput device, one signal group is created automatically for each input channel (Ch<n>_in, <n> = 01 – 16). Ch01_in – Ch08_in belong to the In_A port; you will use only the signal group Ch04_in.

To configure the DigitalInput device:

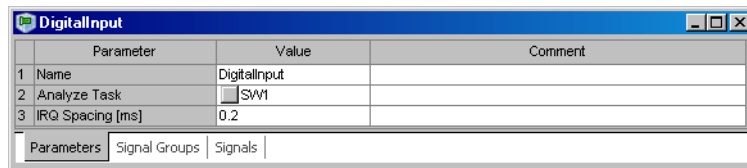
Since you will use the trigger, you need to select a software task for the device.

- Double-click the DigitalInput device.
The editor for the device opens.
- In the "Parameters" tab of the device editor, click the button in the "Analyze Task" row, "Value" column.
The "Task selection" window opens. It lists all tasks of the project that can be selected.



- Select the SW1 software task and click **OK**.
You can use the default settings for the other options.

The tab should now look as follows:



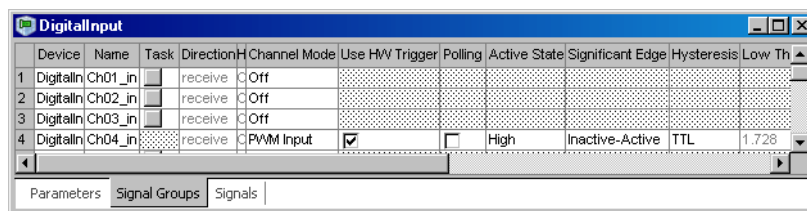
	Parameter	Value	Comment
1	Name	DigitalInput	
2	Analyze Task	SW1	
3	IRQ Spacing [ms]	0.2	

Signal Group Configuration: You select the channel mode (and with that, the available signals), assign a task and make other necessary settings for the Ch04_in signal group.

To configure the Ch04_in signal group:

- Double-click the `DigitalInput` device.
The editor for the device opens.
 - In the device editor, go to the "Signal Groups" tab.
This tab lists all signal groups and their parameters.
1. Selecting the channel mode
 - In the "Channel Mode" column, select the `PWM Input` mode.
The default values for this mode are entered in the columns "Active State", "Significant Edge", "Hysteresis", "Low Threshold [V]" and "High Threshold [V]".
The necessary signals are created automatically.
 2. Set up trigger usage
 - In the "Use Hardware Trigger" column, activate the option.
 - In the "Polling" column, deactivate the option.
The field in the "Task" column is reset and locked.

The "Signal Groups" tab should now look as follows:



	Device	Name	Task	Direction	Channel Mode	Use H/W Trigger	Polling	Active State	Significant Edge	Hysteresis	Low Th
1	DigitalIn	Ch01_in		receive	Off						
2	DigitalIn	Ch02_in		receive	Off						
3	DigitalIn	Ch03_in		receive	Off						
4	DigitalIn	Ch04_in		receive	PWM Input	<input checked="" type="checkbox"/>	<input type="checkbox"/>	High	Inactive-Active	TTL	1.728

Signal Configuration: The signals that were created during signal group configuration are contained in the "Signals" tab.

To rename signals:

The signal names do not contain the channel number. For clarity, you will rename them.

- Double-click in the "Name" column.

- Rename the signals as follows.

Group	New signal name
Ch04_in	Ch04in_ActiveTime
Ch04_in	Ch04in_InactiveTime

During signal mapping, ASCET messages from the project are assigned to the signals of the `DigitalInput` signal group. A selection dialog can be opened via the "Message" parameter of each signal. This dialog contains only messages which have the attribute *Exported* and correspond to the transfer direction of the signal group (here: *receive*); i.e.:

- Direction = receive → receive messages

To assign ASCET messages manually:

- Click the button in the "Message" column.
The "Message selection" window opens. It lists all messages in the project that can be selected.
- Select the following ASCET messages for the signals.

Signal

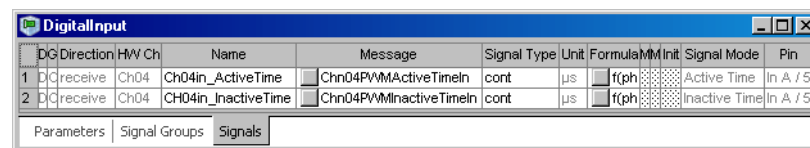
Ch04in_ActiveTime
Ch04in_InactiveTime

ASCET message

Chn04PWMActiveTimeOut
Chn04PWWInactiveTimeOut

- Click **OK**.

The "Signals" tab should now look as follows.



DG	Direction	HW	Ch	Name	Message	Signal Type	Unit	Formula	MM	Init	Signal Mode	Pin
1	D	C	receive	Ch04	Ch04in_ActiveTime	Chn04PWMActiveTimeIn	cont	µs	f(ph)		Active Time	In A / 5
2	D	C	receive	Ch04	CH04in_InactiveTime	Chn04PWWInactiveTimeIn	cont	µs	f(ph)		Inactive Time	In A / 5

9.5.5 Saving the Hardware Configuration and Generating Code

The created hardware configuration must be saved, and code must be generated. This is done with a single command.

To save the hardware configuration and generate code:



- In the Hardware Configurator, click on the **Save** button

or

- select **File** → **Save Workspace** to save the hardware configuration.

A hardware configuration file (* .hwx) is written to the working directory specified in the Hardware options (see section 2.1.1 on page 13).

Code is generated for the hardware configuration; information, warning and error messages are shown in the ASCET monitor window.

If no error occurred, the `HardwareConfiguration` component in the project editor is updated, and the hardware processes are assigned to the respective tasks.

You can export the hardware configuration to a file, see page 77.

9.5.6 Experimenting with the Sample Project

The hardware configuration is complete, and you can close the Hardware Configurator. The next step is to start the regular code generation for the experimental target from within the project editor.

Tip

It is not recommended to include the hardware configuration in the graphical display of the Project Editor. Since this module can change with each hardware configuration code generation process, it could result in an unfavorable representation.

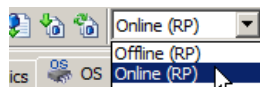
To generate code for the experimental target:

- In the project editor, select **Build** → **Build** to generate code for the entire project.
- Select **Build** → **View Generated Code** to view the generated code.

To experiment online:

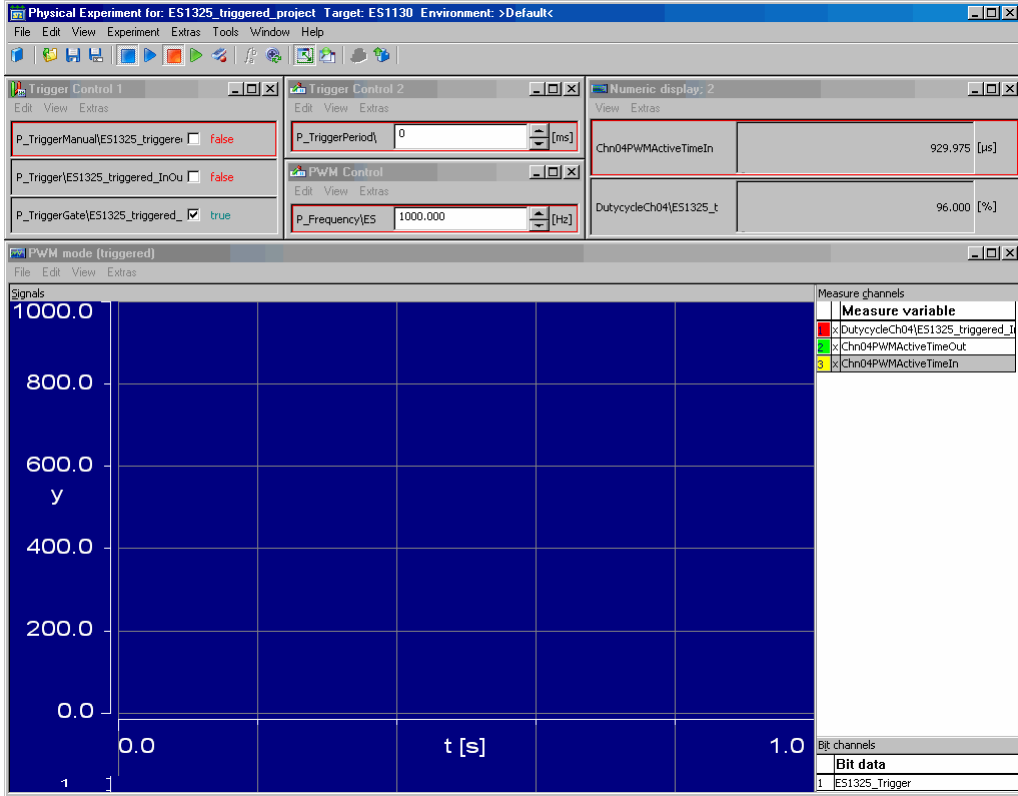
- Connect the inputs and outputs of the ES1325 according to section "Connections" on page 94.
- Switch on the power supply of the ES1000.x.
- Select **Online (RP)** from the "Experiment Target" combo box.

`Offline (RP)` is intended for offline experiments on the Target.



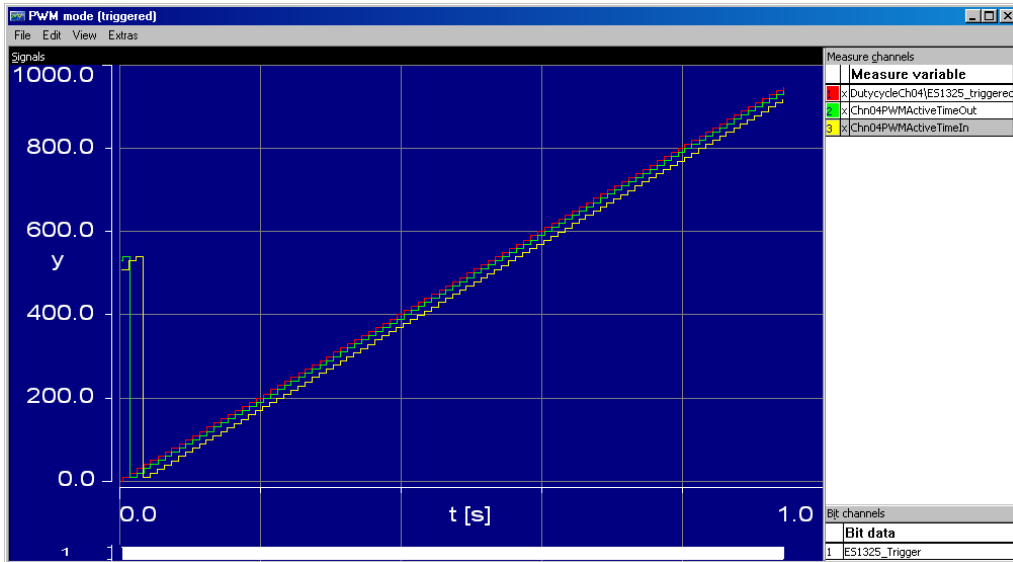
- Select **Build** → **Experiment**.

The "Physical Experiment" window and the pre-defined experiment environment (one oscilloscope, a numeric display and three calibration windows) open.



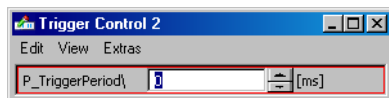
The default setting assumes automatic calculation of the first trigger signal and a constant second trigger signal (parameter P_TriggerManual is false, parameter P_TriggerGate is true).

The "PWM mode (triggered)" oscilloscope shows the variable `DutycycleCh04` (red), the output message `Chn04PWMActiveTimeOut` (green) and the input message `Chn04PWMActiveTimeIn` (yellow), as well as the `ES1325_Trigger` message, containing the first trigger signal, in the lower section.

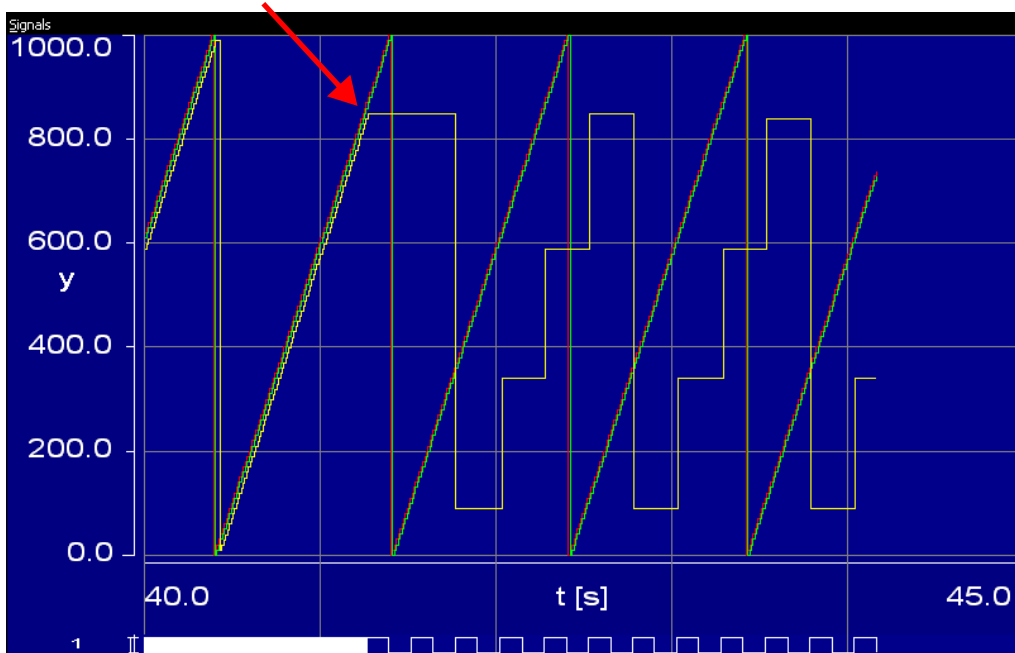


When you start measuring with the defined values, the green curve is one step behind the red curve, and the yellow curve is three steps behind the red curve.

If you increase the value of the `P_TriggerPeriod` parameter in the "Trigger Control 2" calibration window, the time between two rising edges increases. `Chn04PWMActiveTimeIn` is thus read in correspondingly less frequently.

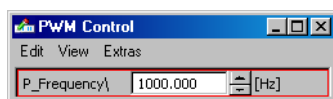


At the point marked in the following diagram, the `P_TriggerPeriod` parameter was set from 0 ms to 250 ms. The trigger signal at the bottom of the display area of the oscilloscope only delivers a rising edge every 250 ms; the `Chn04PWMActiveTimeIn` message is read in accordingly rarely and the relevant curve has large steps.

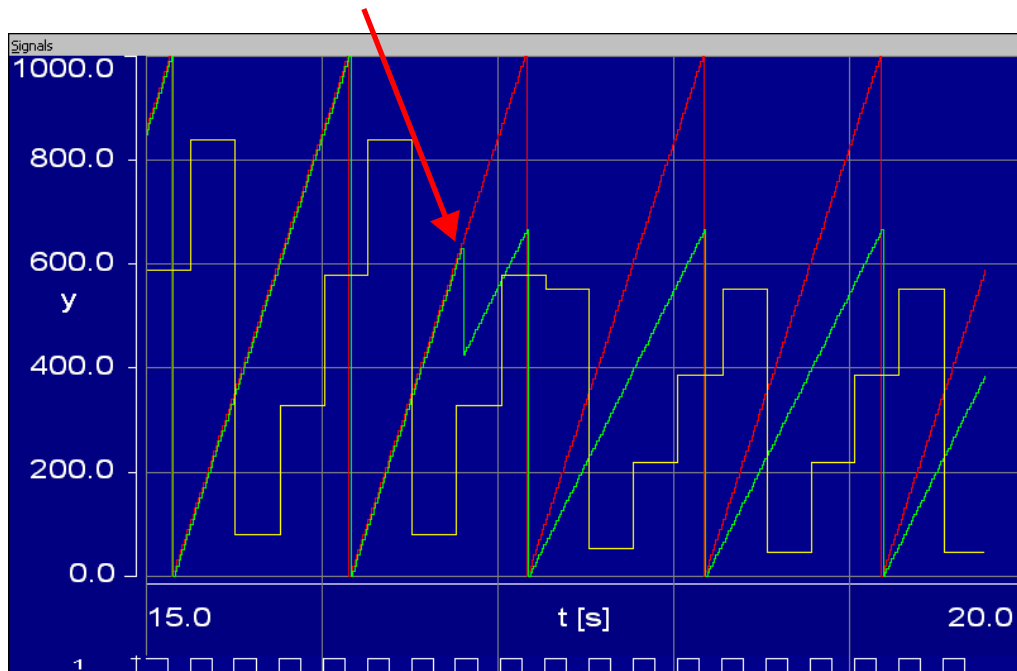


For this figure, the range of the x-axis in the "Extent" field of the "Display setup" window (to be reached from the oscilloscope using **Extras** → **Setup**) was set to 5 seconds.

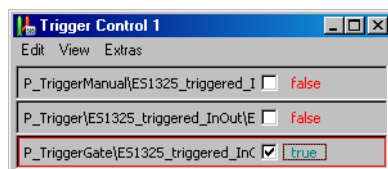
If you modify the parameter `P_Frequency` in the "PWM Control" calibration window, the frequency changes and hence the active and inactive time. The duty cycle remains the same.



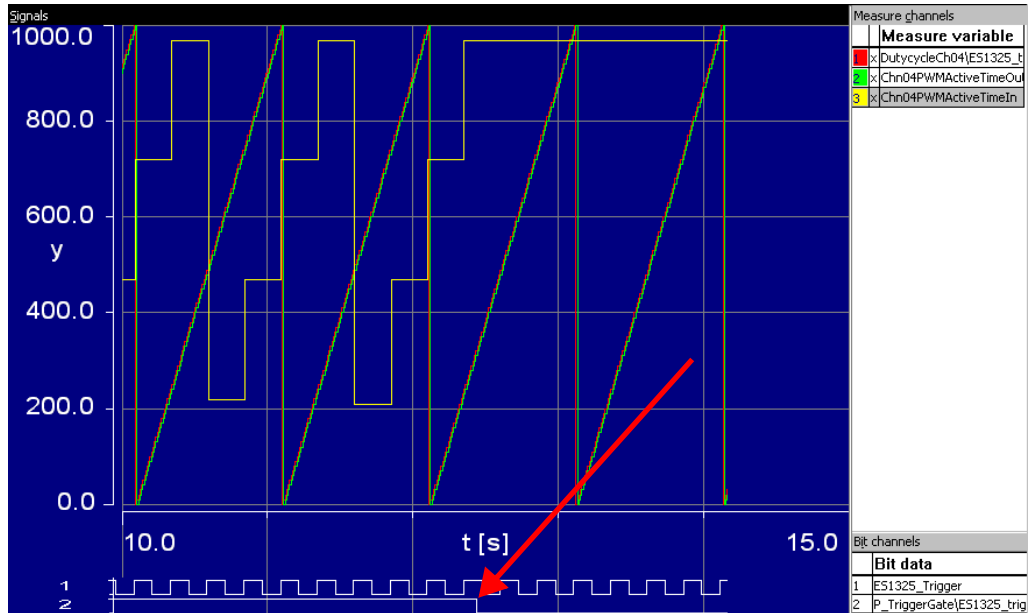
At the point marked in the following diagram, the frequency was set from 1000 Hz to 1500 Hz. The curve for `DutycycleCh04` remains unchanged, but the curves for `Chn04PWMActiveTimeOut` and `Chn04PWMActiveTimeIn` now have lower maxima.



You can change the trigger control in the "Trigger Control 1" calibration window.



If, for example, you set the `P_TriggerGate` parameter to `false`, `Chn04PWMAciveTimeIn` is no longer read in as the second trigger signal is `false`. The curve remains at the last value. The curves for `DutycycleCh04` and `Chn04PWMAciveTimeOut` remain unchanged.

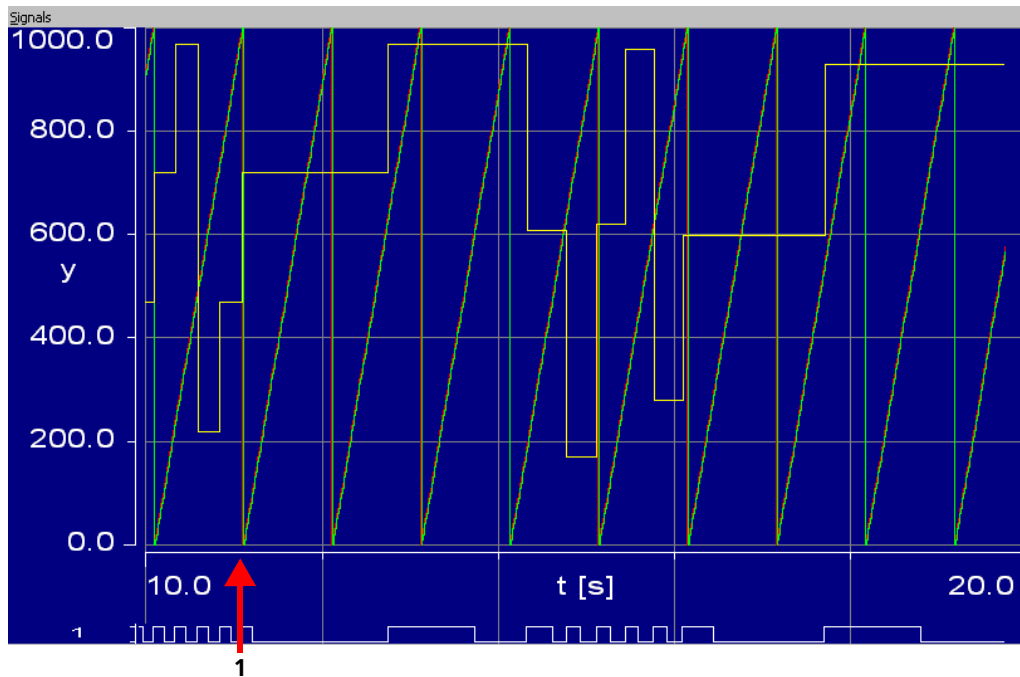


For this figure, the parameter `P_Frequency` was again set to 1000 Hz, and `P_TriggerGate` was added to the oscilloscope.

Set `P_TriggerGate` back to `true` to continue measuring `Chn04PWMAciveTimeIn`.

If you set the `P_TriggerManual` parameter in the "Trigger Control 1" window to `true`, you activate manual control of the first trigger signal. Each time you change the `P_Trigger` parameter from `false` to `true`, a trigger signal is generated and the `Chn04PWMAciveTimeIn` message read in.

At the point marked in the following diagram, manual triggering was activated. The course of the `ES1325_Trigger` signal shows how `P_Trigger` has been changed. `Chn04PWMActiveTimeIn` was read in with every rising edge of the trigger signal.



For this figure, `P_TriggerGate` has been removed from the oscilloscope. The range of the x-axis in the "Extent" field of the "Display setup" window (to be reached from the oscilloscope using [Extras](#) → [Setup](#)) was set to 10 seconds.

10 RTPRO-PC Tutorial

The example in the RTPRO-PC tutorial requires a notebook with ASCET-RP and RTPRO-PC installations, a suitable HSP version and an ES581.3 with appropriate cables.

You will configure RTPRO-PC, import and adapt the model, set up the hardware configuration, and generate the code.

10.1 Preparations

10.1.1 RTPRO-PC Configuration

To perform the tasks in this example, you must boot the notebook in ETAS RTPRO-PC and Windows or ETAS RTPRO-PC mode.

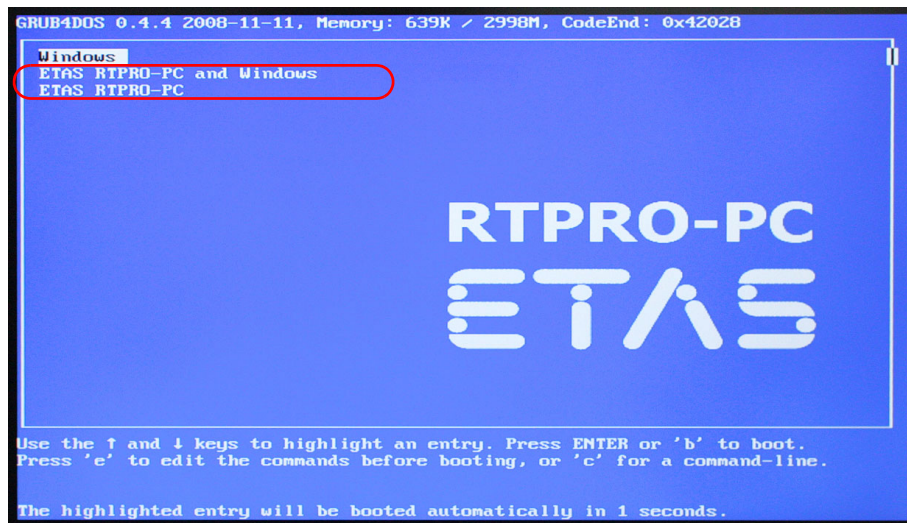


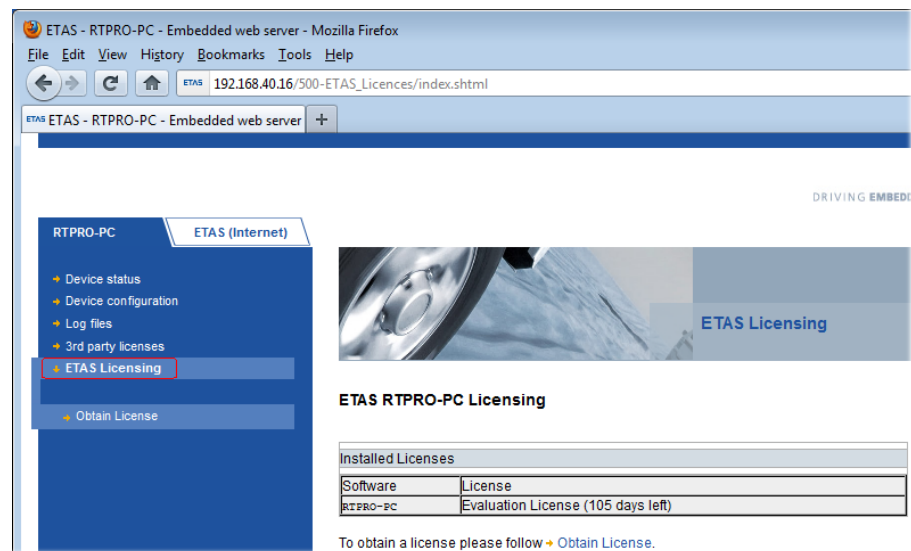
Fig. 10-1 RTPRO-PC: Boot configuration selection

Once the notebook has booted, you should check whether a valid RTPRO-PC license is available.

To check the RTPRO-PC license status:

- Launch the RTPRO-PC web interface.

The license status is shown on the start page, the "Device status" page, and the "ETAS Licensing" page.



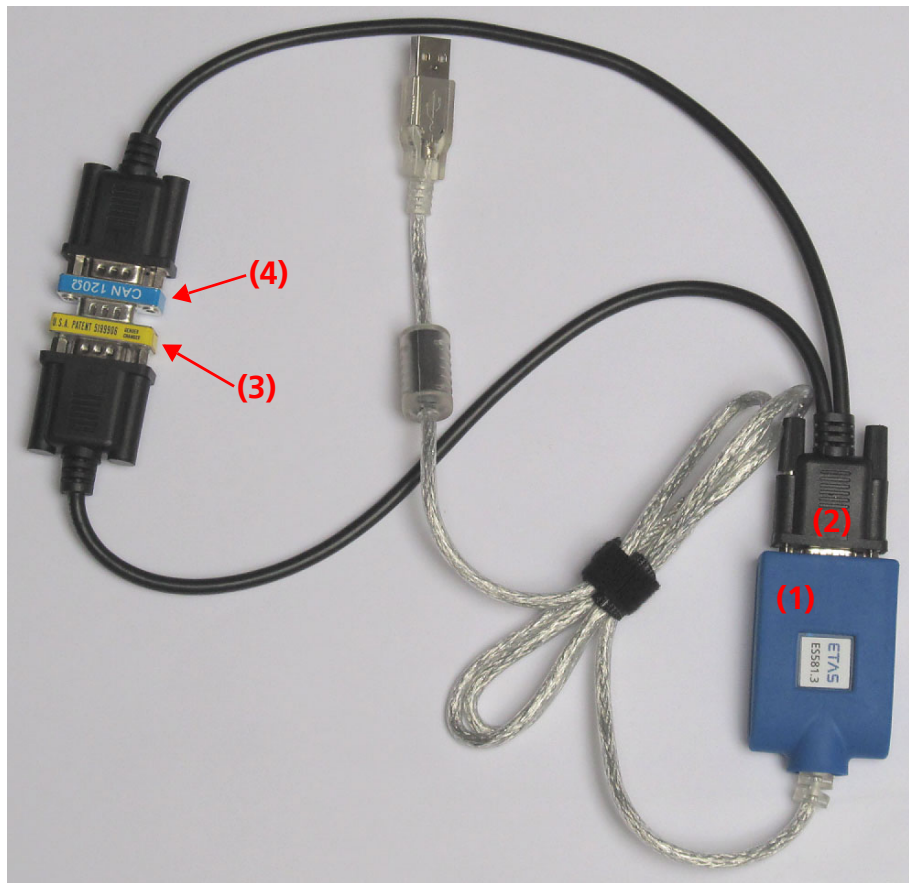
Next, you will connect and set up the ES581. ASCET-RP references ES581 devices via consecutive IDs. These IDs must be mapped to the ES581, referenced by serial number, actually connected to the notebook.

You do not have to install the ES581 driver, the driver is part of the RTOS image of RTPRO-PC.

You need the following equipment:

1. ES581.3
2. a Y cable (shipped with ES581)
3. a "gender changer" to connect the DSUB connectors of the Y cable

4. a termination for the CAN bus



To set up the ES581:

- Plug the ES581 into an USB port assigned to the simulation node.
See the RTPRO-PC user's guide for further information.
- Launch the RTPRO-PC web interface.
- Navigate to the "ES581 Device Settings" page.
- Open the combo box of the first ES581 device and select the serial number.

ID	Actual assignment	To change assignment
ES581 device #1 (ID1)	not used	enter number <input type="text"/> ▼
ES581 device #2 (ID2)	not used	not used 34525

The number is displayed in the text field above the combo box.

- Click **Save config.** to store your settings.

Tip

Make sure that the settings here match the ES581 settings in ASCET-RP.

Your settings are stored in an internal file on the USB stick used as persistent memory for the simulation node.

With that, RTPRO-PC configuration is complete.

10.1.2 ASCET-RP Preparations

Next, you start ASCET-RP and import the sample files for this task. These files are stored in the `Tutorial RTPROPC.*` file (* = `exp` or `ax1`). During the installation, the sample file is stored in the subdirectory `ASCET6.3\export` of your ASCET installation.

The sample files can be imported into a new or an existing database/workspace.

Tip

This tutorial is written under the assumptions that a new database is used.

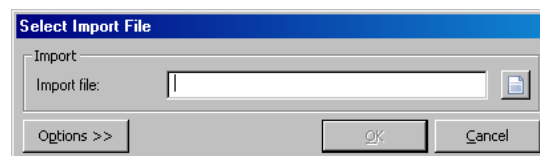
To create a new database:


- In the Component Manager, select the **File → New Database** menu option.
The "New Database" window opens.
- Enter the name for the new database.
- Click **OK**.
The database is created and opened.

Creating a new workspace is described in the ASCET online help.

To import the exercise example:

- In the Component Manager, select **File → Import**.
The "Select Import File" window opens.



- Use the  button to select the `Tutorial RTPROPC.*` file from the `ASCET6.3\export` subdirectory of your ASCET installation.
The **OK** button is now available.
You can use the default import options, so that no further action is required.
- Click **OK** to start the import.

- Make sure all components have been selected for the import in the "Import" window..
- Click **OK**.
The files are imported. A list of the imported database items is shown in the "Imported Items" window.
- Close the "Imported Items" window.

To activate the Hardware Configurator:

- In the Component Manager, select **Tools** → **Options** to open the ASCET options window.
- In the options window, open the "Hardware\Hardware Configuration" node.
- Deactivate the **Use legacy HWC Editor (deprecated)** option.
- Click **OK** to close the options window.

10.2 Adapting the Model

The module used here contains the same functionality as the module used in section 9.2. It is used without adaptations.

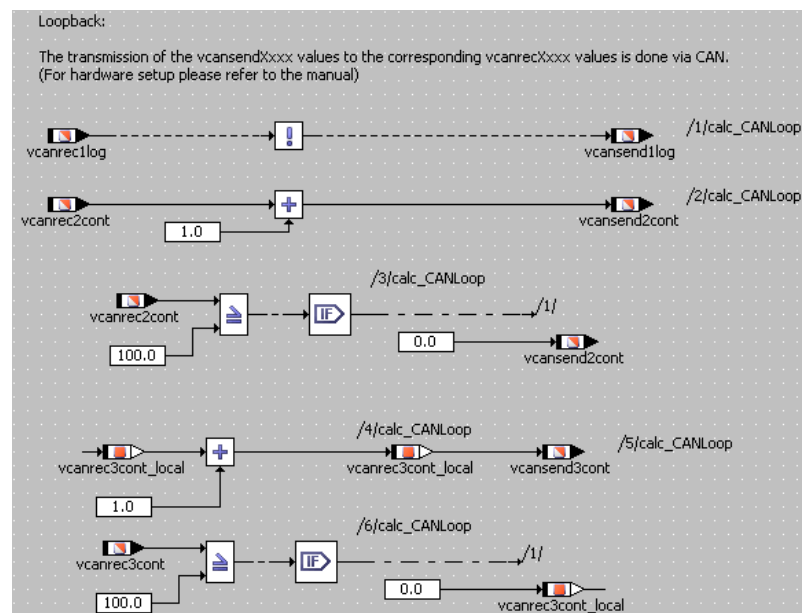


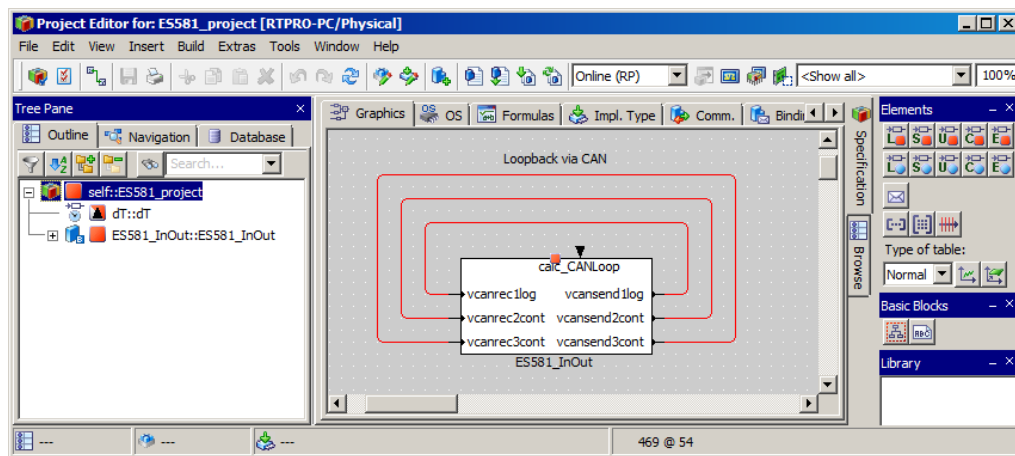
Fig. 10-2 RTPRO-PC – ES581_InOut module

The project used here needs some adaptations.

To open the exercise example:

- In the Component Manager, select the `... \RTIOTutorial\ES581_RTPROPC` folder.

- Open the ES581_project project.



- Click the **Project Properties** button.
The "Project Properties" window opens.
- In the "Build" node, select the target **RTPRO-PC**, Compiler and operating system are selected automatically.
- Close the "Project Properties" window.

Tip

Only messages declared as "Exported" are available for communication.

The predefined tasks are the same as used in section 9.2. The ES581 requires two further tasks.

To create the required tasks:

- In the project editor, open the "OS" tab.
- In the "Tasks" pane, create two tasks of type **Interrupt**.
Details are given in the ASCET online help.

10.3 Creating the Hardware Configuration

To prepare the hardware configuration:

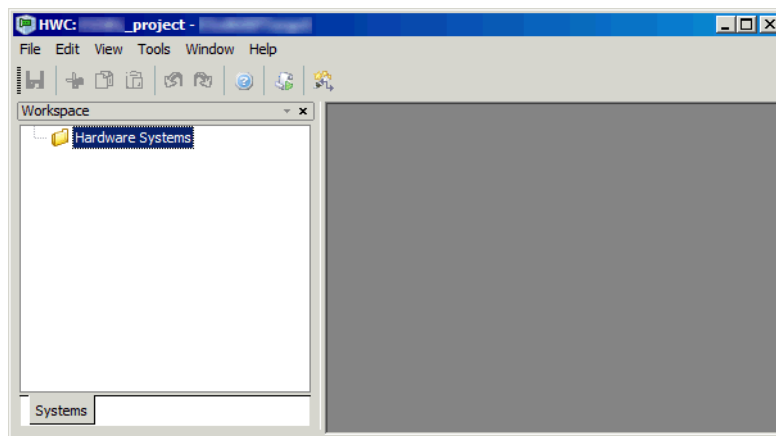
You have to create an empty hardware configuration in the Component Manager and add it to the project.

- In the Component Manager, select **Insert** → **Hardware Configuration**.
A new hardware configuration appears in the "1 Database" list. Its name is highlighted for in-place editing.
- If desired, edit the name (default: **HardwareConfiguration**).

- Press <ENTER> to confirm the name.
- Add the hardware configuration to the project, e.g., via **Insert Component** in the project editor.

To open the Hardware Configurator:

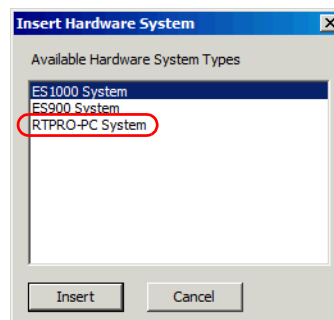
- In the "Outline" tab of the project editor, double-click the **Hardware Configuration**.
The Hardware Configurator opens. The title bar contains the project name and the selected experimental target. The **Hardware Systems** folder always exists, it forms the root of the hardware configuration tree.



To specify the hardware configuration:

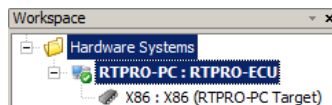
The hardware is described as a tree structure in the Workspace Browser.

- In the Hardware Configurator, right-click the **Hardware Systems** folder and select **Add Hardware System** from the context menu.
The "Insert Hardware System" window opens. This window lists the available hardware systems.



- Select the **RTPRO-PC** system entry.
- Click **Insert**.

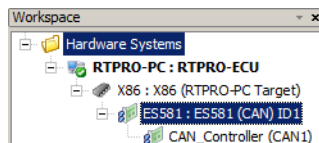
- In the "Rename" dialog window, enter a name (default: `RTPRO-PC`) and click **OK**.
The hardware system is added below the `Hardware Systems` folder.
- Right-click the hardware system and select **Insert Target** from the context menu.
Since only one target type is available for RTPRO-PC systems, the "Insert New Item" window is omitted and the "Rename" window opens.
- Enter a name and click **OK**.
The target is added below the hardware system.



To integrate the ES581:

1. Adding the ES581

- Right-click the target and select **Insert** from the context menu.
The "Insert New Item" window opens.
- Select the `ES581 (CAN)` entry.
This entry is used to describe the ES581 CAN interfaces.
- Click **Insert**.
- In the "Rename" dialog window, enter a name (default: `ES581`) and click **OK**.
The `ES581` board is added. One `CAN_Controller` is added automatically.



2. Adding a second CAN controller

- Right-click the `ES581` board and select **Insert** from the context menu.
Since only CAN controllers can be added, the "Insert New Item" window is omitted, and the "Rename" window opens immediately.
- Enter a name and click **OK**.

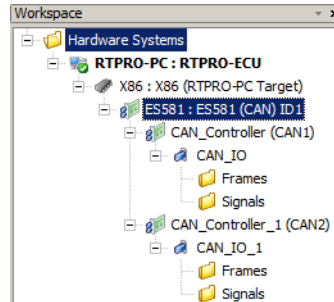
3. Adding a CAN_IO device to each controller

- Right-click a `CAN_Controller` and select **Insert** from the context menu.

The "Insert New Item" window opens.

- Select `CAN_IO` and click **Insert**.

The `CAN_IO` device is added and, below that, folders for `Frames` and `Signals`.



For this tutorial, the default settings for the ES581 board and the CAN controllers are used. You have to configure only the CAN IO devices and the frames and signals.

10.4 Configuring the CAN Channels

Next, the two CAN channels must be configured, i.e. the necessary tasks must be assigned to the `CAN_IO` devices, and CAN signals and frames must be created.

The first channel (`CAN_IO` in the Hardware Configurator) will send the signals which are received by the second channel (`CAN_IO_1`).

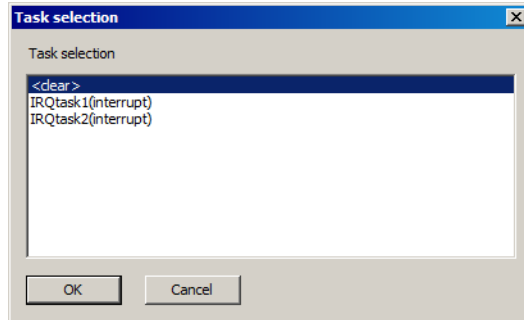
To configure the CAN IO devices:

- In the Hardware Configurator, double-click a `CAN_IO` device.

The editor for the device opens. The "Parameters" tab lists all parameters for the device.

- In the "Parameters" tab of the device editor, click on the button next to the "Interrupt Service Routine" option.

The "Task selection" window opens. It lists all interrupt tasks available in the project.



- Select a task and click **OK**.
Each interrupt task can be assigned to only one CAN IO device. If you assign the same task to a second device, the task is deassigned from the first device.

Channel 1

CAN Frames: The CAN messages are transmitted in form of CAN frames (signal groups).

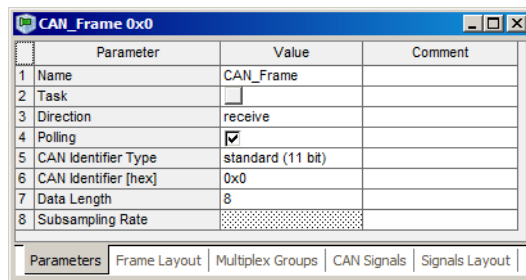
To create CAN frames:

- In the Hardware Configurator, right-click the **Frames** folder below the first CAN_IO device (CAN_IO) and select **Insert** from the context menu.
The "Rename" dialog window opens.
- Enter a name and click **OK**.
The CAN frame is added to the folder. Newly created frames always have the direction *Receive*.
- Add a second CAN frame.

To configure the first CAN frame:

- In the Hardware Configurator, double-click the first CAN frame.

The editor for the frame opens. The "Parameters" tab lists all parameters for the frame.



- In the "Parameters" tab of the frame editor, double-click in the "Name" row, "Value" column.
The table field changes to an input field.
- Name the frame `Frame_64`.
- Click the button in the "Task" row, "Value" column.
The "Task selection" window opens. It lists all tasks of the project that can be selected.
- Select the `T10ms` alarm task and click **OK**.
- Click in the "Direction" row, "Value" column.
The table field changes to a combo box.
- Select the direction `Send`.
- Double-click in the "CAN Identifier [hex]" row, "Value" column.
- Enter the value `0x64` for the identifier.
- Close the editor.

To configure the second CAN frame:

- Open the editor for the second CAN frame.
- Name the second frame `Frame_65`.
- Select the `T200ms` alarm task.
- Select the direction `Send`.
- Enter the identifier value `0x65`.
- Close the editor.

CAN signals: The CAN signals are first created in the `signals` folder, and then assigned to a CAN frame. The example uses two numeric and one logical signal.

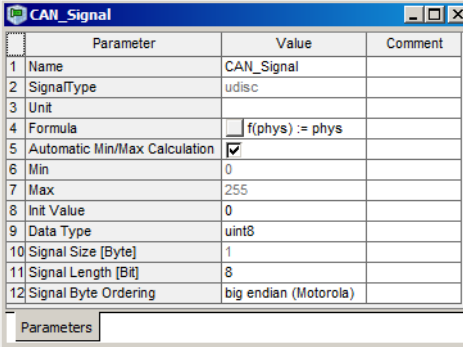
To create signals:

- In the Hardware Configurator, right-click the `signals` folder below the first CAN_IO device (`CAN_IO`) and select **Insert** from the context menu. The "Rename" dialog window opens.
- Enter a name and click **OK**. The CAN signal is added to the folder.
- Add two more CAN signals.

The logical and one numeric signal are transferred in the first CAN frame, the other numeric signal is transferred in the second CAN frame.

To configure a logical signal:

- In the Hardware Configurator, double-click the first CAN signal. The editor for the signal opens. The "Parameters" tab lists all signal parameters.



	Parameter	Value	Comment
1	Name	CAN_Signal	
2	SignalType	udisc	
3	Unit		
4	Formula	f(phys) := phys	
5	Automatic Min/Max Calculation	<input checked="" type="checkbox"/>	
6	Min	0	
7	Max	255	
8	Init Value	0	
9	Data Type	uint8	
10	Signal Size [Byte]	1	
11	Signal Length [Bit]	8	
12	Signal Byte Ordering	big endian (Motorola)	

- In the signal editor, name the signal `vcansend1log`.
- Click in the "Data Type" row, "Value" column. The table field changes to a combo box which lists the available types.
- From the combo box, select `boolean`.

To configure the numeric signals:

- Open the editor for the second CAN signal.
- Name the signal `vcansend2cont`.
- Select the "Data Type" `uint8`.
- Open the editor for the third CAN signal.
- Name the signal `vcansend3IRQ`.
- Select the "Data Type" `uint8`.
- Close the signal editors.

The configuration of the signals is not yet complete. Before the final configuration steps can be done, the signals must be assigned to CAN frames.

To assign CAN signals to CAN frames:

- Right-click the `Frame_64` frame and select **Insert** from the context menu.
The "Insert Item" window opens. It lists all available CAN messages.
- Select the `vcansend1log` signal and click **Insert**.
- Assign the `vcansend2cont` signal to the same CAN frame.
- Assign the `vcansend3IRQ` signal to the `Frame_65` CAN frame.

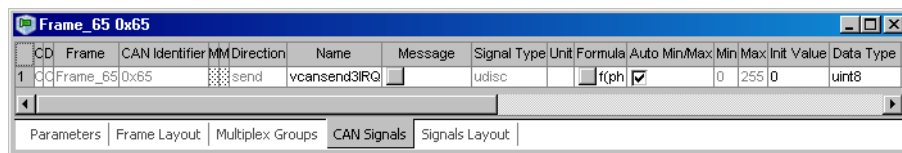
Now you can complete the signal configuration, i.e. assign ASCET messages to the CAN signals and edit the signals layout. However, these configuration options are only available when you edit the signals from the hierarchy branch beginning with the `Frames` folder.

During signal mapping, CAN signals and ASCET messages from the project are assigned to each other. A selection dialog can be opened via the "Message" parameter of each CAN signal assigned to a CAN frame. This dialog contains only messages featuring the *Exported* attribute. Make sure that you assign only ASCET messages that correspond to the frame direction, i.e.,

- frame direction = receive → receive-messages,
- frame direction = send → send-messages

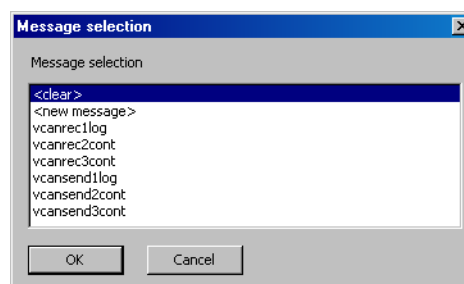
To assign ASCET messages manually:

- Double-click the `Frame_65` frame.
The editor for the frame opens.
- Go to the "CAN Signals" tab.



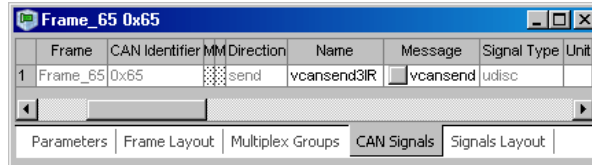
This tab contains the signal parameters of all signals assigned to the frame.

- Click the button in the "Message" column.
The "Message selection" window opens. It lists all messages in the project that can be selected.



- Select the ASCET message `vcansend3cont` and click **OK**.

The ASCET message is assigned to the `vcansend3IRQ` CAN signal.



- Close the frame editor.

To assign ASCET messages automatically:

The signals in `Frame_64` have the same names as two send-messages in the project. You can assign the messages automatically.

- Right-click the `Frame_64` frame and select **Automatic Signal Mapping** from the context menu.
The "Item Signal Mapping" window opens.



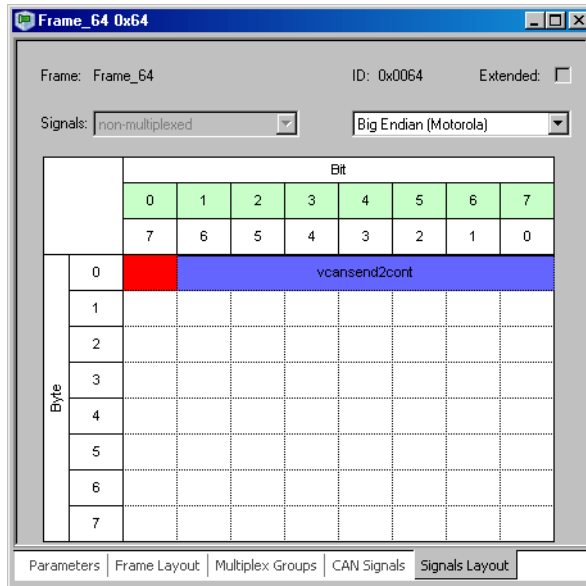
<F1> opens a detailed description.

- Activate the **Map unmapped signals** and **Create ASCET messages** options.
- Click **OK** to start the mapping.

The `vcansend1log` and `vcansend2cont` signals are assigned to the ASCET messages of the same name.

To edit the signals layout:

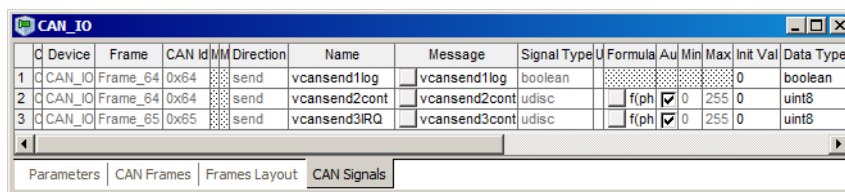
- In the frame editor for `Frame_64`, go to the "Signals Layout" tab.



<F1> opens a detailed description.

- Check the signal map for signal overlaps (i.e. two or more signals using the same bit); these are highlighted in red.
- Check for unwanted gaps (unoccupied bits) between signals.
- Remove any overlaps and/or unwanted gaps by dragging the associated signals.
- Close the frame editor.

This completes the configuration of the first CAN channel.



Channel 2

CAN frames: The second CAN_IO device needs two CAN frames with direction *Receive*.

To create and configure the CAN frames:

- In the Hardware Configurator, create two CAN frames in the `Frames` folder below the second CAN_IO device (`CAN_IO_1`).

- Assign the following parameters to the first CAN frame in `CAN_IO_1`:

Name	<code>Frame_64</code>
Task	<code>T10MS</code>
CAN Identifier [hex]	<code>0x64</code>

- Assign the following parameters to the second CAN frame in `CAN_IO_1`:

Name	<code>Frame_65</code>
Task	<code>Analyze</code>
Polling	<code>(deactivated)</code>
CAN Identifier [hex]	<code>0x65</code>

CAN signals: The second `CAN_IO` device needs three CAN signals.

To add and configure the CAN signals:

- In the Hardware Configurator, create three CAN signals (see page 144) in the `signals` folder below the second `CAN_IO` device (`CAN_IO_1`).

- Assign the following parameters to the first CAN signal in `CAN_IO_1`:

Name	<code>vcanrec1log</code>
Data Type	<code>Boolean</code>

- Assign the following parameters to the second CAN signal in `CAN_IO_1`:

Name	<code>vcanrec2cont</code>
Data Type	<code>uint8</code>

- Assign the following parameters to the CAN signal in `CAN_IO1_1`:

Name	<code>vcanrec3cont</code>
Data Type	<code>uint8</code>

- Assign the first two signals to the `Frame_64` frame (see page 145).

- Assign the third signal to the `Frame_65` frame.

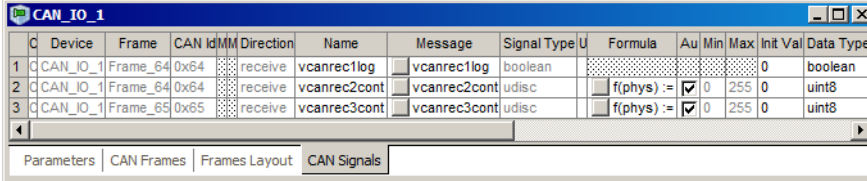
- Adjust the signals layout in `Frame_64` (see page 147).

- Assign messages to the signals.

For the three signals, receive-messages with the same names exist in the project so that all messages can be assigned automatically (see page 146).

Of course, the messages can also be assigned manually as described on page 145.

This completes the configuration of the second CAN channel.



C	Device	Frame	CAN Id	MM	Direction	Name	Message	Signal Type	U	Formula	Au	Min	Max	Init Val	Data Type
1	CAN_IO_1	Frame_64	0x64		receive	vcanrec1log	vcanrec1log	boolean						0	boolean
2	CAN_IO_1	Frame_64	0x64		receive	vcanrec2cont	vcanrec2cont	udisc		f(phys) :-	<input checked="" type="checkbox"/>	0	255	0	uint8
3	CAN_IO_1	Frame_65	0x65		receive	vcanrec3cont	vcanrec3cont	udisc		f(phys) :-	<input checked="" type="checkbox"/>	0	255	0	uint8

10.5 Saving the Hardware Configuration and Generating Code

The created hardware configuration must be saved, and code must be generated. This is done with a single command.

To save the hardware configuration and generate code:



- In the Hardware Configurator, click on the **Save** button

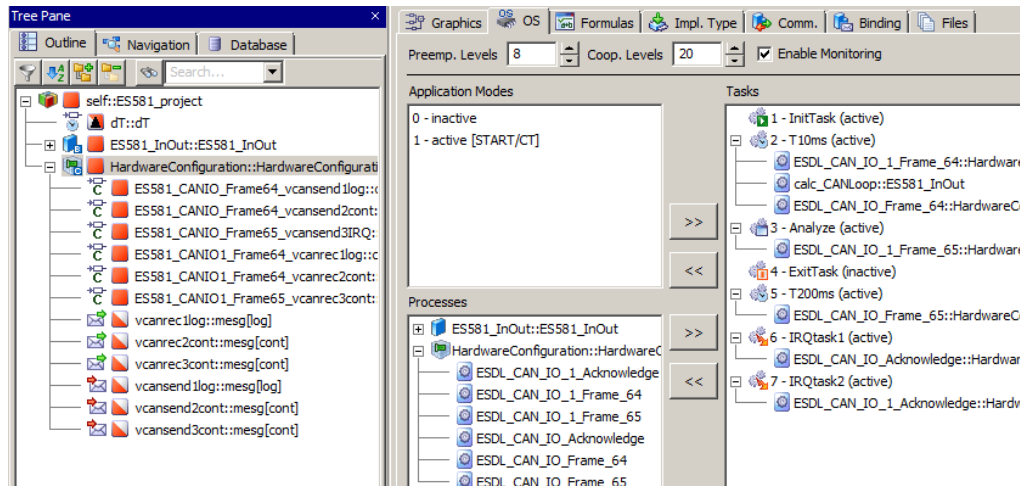
or

- select **File** → **Save Workspace** to save the hardware configuration.

A hardware configuration file (*.hwx) is written to the working directory specified in the Hardware options (see section 2.1.1 on page 13).

Code is generated for the hardware configuration; information, warning and error messages are shown in the ASCET monitor window.

If no error occurred, the HardwareConfiguration component in the project editor is updated, and the hardware processes are assigned to the respective tasks.



If desired, you can export the hardware configuration to a file, see "To export the hardware configuration as a file:" on page 77.

10.6 Experimenting with the Sample Project

The hardware configuration is complete, and you can close the Hardware Configurator. The next step is to start the regular code generation for the experimental target from within the project editor.

Tip

It is not recommended to include the hardware configuration in the graphical display of the Project Editor. Since this module can change with each hardware configuration code generation process, it could result in an unfavorable representation.

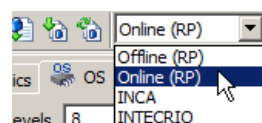
To generate code for the experimental target:

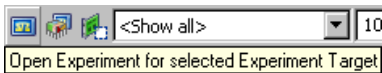
- In the project editor, select **Build** → **Build All** to generate code for the entire project.
- Select **Build** → **View Generated Code** to view the generated code.

To experiment online:

- Make sure that the DSUB connectors of the Y cable are connected, so that communication can take place.
- Select **Online (RP)** from the "Experiment Target" combo box.
The other entries are irrelevant for this tutorial.
- Select **Build** → **Experiment**.

Or





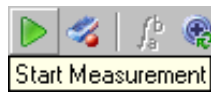
- Click the **Open Experiment for selected Experiment Target** button.

The "Physical Experiment" window and the pre-defined experiment environment consisting of two oscilloscopes open.



- In the "Physical Experiment" window, click the **Start OS** button.

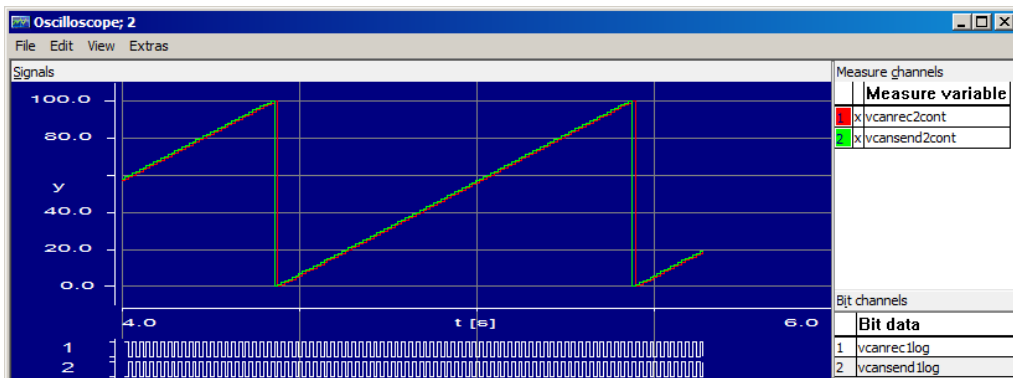
The operating system starts and the model is executed.



- In the "Physical Experiment" window, click the **Start Measurement** button.

The values of the ASCET messages are displayed on both oscilloscopes.

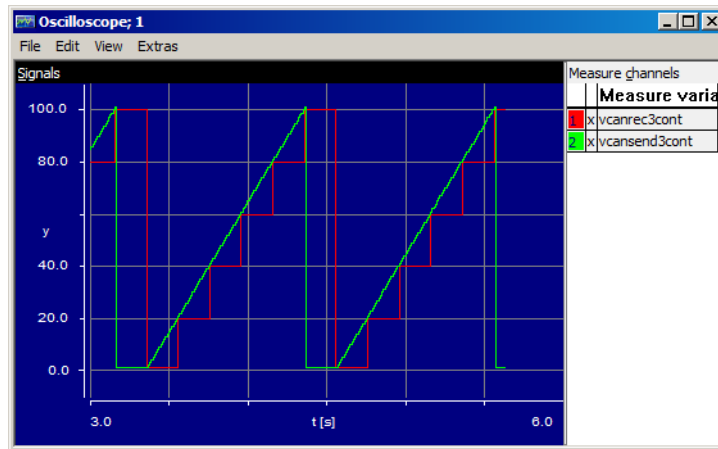
The top oscilloscope shows the `vcansend2cont` send-message (green, left curve) and the `vcanrec2cont` receive-message (red, right curve) in the "Signals" area. The bit display shows the `vcanrec1log` receive-message (top curve) and the `vcansend1log` send-message (bottom curve).



Characteristic and offset of the curves are generated as follows:

- The values for the send-messages are processed (`calcCANLoop` process in the `T10ms` task) and sent (`ESDL_CAN_IO_Frame64` process in the `T10ms` task) in the 10-ms raster.
- The processed value is read in again via receive-message (`ESDL_CAN_IO_1_Frame64` process in the `T10ms` task), but delayed by one time unit.

The bottom oscilloscope shows the `vcansend3cont` send-message (green curve with small steps) and the `vcanrec3cont` receive-message (red curve with large steps).



The `vcansend3cont` send-message is processed in the 10-ms raster (`calc-CANLoop` process in the `T10ms` task), which causes the green curve to respond as smoothly as the curve of `vcansend2cont`. The value is sent in the 200-ms raster (`ESDL_CAN_IO_1_Frame65` process in the `T200ms` task) and received again in the interrupt mode via receive-message after being triggered by the `Analyze` task. The graduated curve for `vcanrec3cont` responds accordingly.

11 Appendix A: Bypass Concept

11.1 ETK Bypass Concept Description

With an ETK equipped ECU, the ECU code must be prepared to set up data structures and communication with the ETK to enable communication between the rapid prototyping system used for the ETK bypass and the ECU itself.

Independent of the ECU implementation of these drivers, some safety issues common to the concept of the bypass must be considered.

11.2 Bypass Input

Like for measurement, the Distab13 (and Distab 12 for hook-based bypass) mechanism is used to provide ECU variables as inputs for the bypass.

The **DIS**play **TABLE** for the Distab13 contains a sorted list of addresses of variables in the ETK Flash. 8, 4, 2 and 1 byte values are supported. The addresses are ordered corresponding to the size of the values they point to. The ECU driver parses the table and writes the contents of the addresses to a table of return values in the ETK RAM. This table is ordered in the same manner as the address list: first, all 8 byte values, then the 4 byte values, etc.

With this approach, INCA and ASCET-RP are given access to values of the internal memory of the microcontroller. Also, collecting the data in a table allows using block modes for transferring the data to the PC.

Fig. 11-1 gives an overview on the data layout for Distab13.

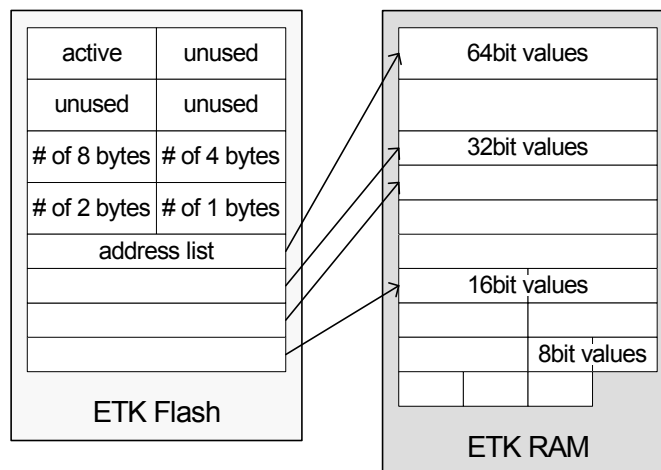


Fig. 11-1 Distab13 data structure

For each bypass raster that contains hooks for the hook-based bypass, an instance of the Distab is created and a Distab process is called. For the service-based bypass, one instance of the Distab data structure exists for each trigger where a service is provided and configured to read ECU values as input for the bypass calculation.

For the hook-based bypass, the number and name of Distabs implemented in the ECU code, and their size, e.g. the number of bytes per channel, is set by the ECU software setup and documented in the A2L description.

For service-based bypass, all tables are allocated dynamically in a given working memory.

11.3 Hook-Based Bypass

For the *hook-based bypass*, the input values of the bypass are gathered with the same Distab13 mechanism as the measurements. After the bypass input data is written to the ETK RAM, the bypass calculation is triggered. In addition, a channel for writing back bypass results to the ETK where they can be retrieved by the ECU is introduced.

For each bypass input channel, an output channel is provided. The size of these channels, as well as their names, also have to be documented in the A2L description. The prototyping tool (INTECRIO or ASCET-RP) can define the number of variables written back to the ECU, depending on the bypass experiment setup. Each variable written to by the bypass must be prepared in the ECU software by applying a hook to prevent the ECU from writing to this value if the bypass is active. The hook code is specific to the ECU and the variable it is applied to. No service is provided within this sample implementation for this task. The values prepared have to be documented in the A2L file by an `IF_DATA ASAP1B_BYPASS` description.

The following figure describes the hook-based bypass principle. The hook indicates the possibility to toggle between the results of the original function (Fn) and the bypass function (Fn*).

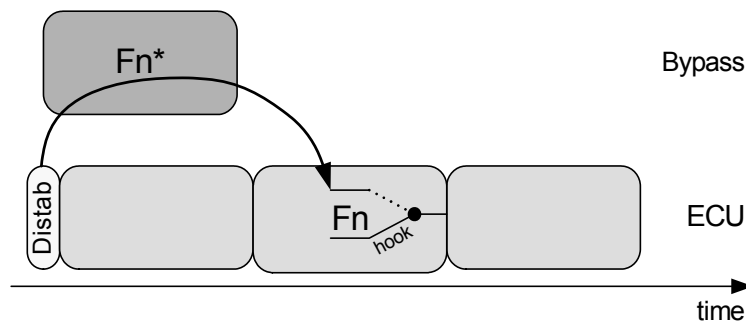


Fig. 11-2 Hook-Based Bypass: Principle

11.4 Service-Based Bypass

Tip

ASCET-RP does not support service-based bypass on an ETK with 8 Mbit/s.

For the *service-based bypass*, both the input values and the output values of the bypass function are transmitted with the same Distab13 mechanism. After the bypass input data is written to the ETK RAM, the bypass calculation is triggered. In addition, a channel for writing back bypass results to the ETK where they can be retrieved by the ECU is introduced. Here, each bypassed ECU process uses and calls its own Distab.

This service also contains an inverted Distab mechanism to write back bypass outputs to the ECU. The ECU does not need to apply hooks to the variables written to, since the service simply overwrites the values with the bypass outputs.

ASCET-RP sets up a Distab-like sorted address table with the addresses of the ECU variables to be written to, and writes the corresponding values in a table of the ETK Flash. The part of the ECU service that writes the bypass outputs to the ECU parses the address table and gets the corresponding values from the data table and writes the values to the ECU addresses.

The following figure describes the service-based bypass principle.

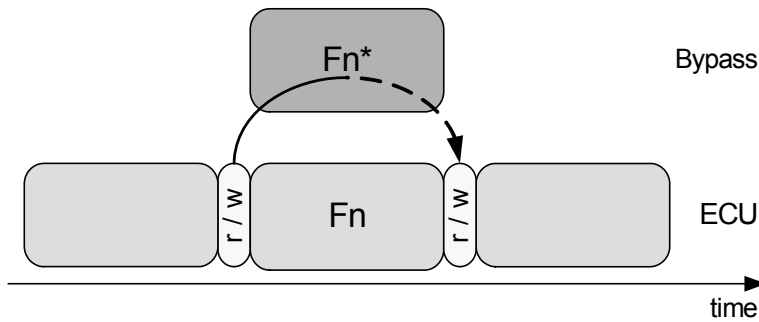


Fig. 11-3 Service-Based Bypass: Principle (The dashed line indicates that bypass data can be written back at a later time as well.)

The current version of ASCET-RP supports several versions of service-based bypass (SBB). The following table lists the supported SBB versions for each target (+: supported, -: not supported).

	SBB V2.0	SBB V2.1	SBB V3.*
ES1000 / ES1232 / ETK	+	-	-
ES 910.2 / supported ETKs	+	+	-
ES 910.3 / supported ETKs	+	+	+
<i>INTECRIO does not support service-based bypass on an ETK with 8 Mbit/s.</i>			

Tab. 11-1 Supported SBB versions

SBB version	supported AML versions	supported Distab types
V2.0 + V3.0	ETK AML 1.2.0 – 1.7.0	Distab13 - Distab16
	XETK AML 1.0.0 – 2.0.0	
	SBB AML ≥ 2.0.0	
V2.1	ETK AML not supported	Distab17
	XETK AML ≥ 2.0.0	
	SBB AML ≥ 3.1.1	
V3.1	ETK AML not supported	Distab17
	XETK AML ≥ 2.0.0	
	SBB AML ≥ 2.0.0	

Tab. 11-2 SBB versions and AML versions

11.5 Safety Considerations

With calculating a bypass function on a rapid prototyping system and feeding data back into the ECU, the same care needs to be taken for development and use of bypass software as for ECU software. The bypass output may directly or indirectly influence the output channels of the ECU. The same applies, of course, if the rapid prototyping system uses own output channels.

Thus, it is highly recommended that the bypass functions include range checks and validations algorithms for the bypass outputs.

11.5.1 Bypass Input Data

To perform proper calculations, the bypass obviously needs consistent and valid input data. The ECU software must ensure this and prevent activation of the bypass if the ECU software detects incorrect or invalid inputs. This also includes ECU states like initialization and afterrun, or error modes the ECU might run in. In other words, activation of and data transfer to the bypass must be covered by the safety mechanisms of the ECU.

11.5.2 Bypass Calculation

The ECU software must be aware whether the bypass is active and must provide measures to react on bypass failures, for example missing calculations or unexpected shut off of the bypass system.

Failsafe measures might be using the alternative output values of the bypassed ECU functions, using constant fallback values, or even resetting the ECU, depending on the bypassed ECU function. This is entirely under responsibility of the ECU provider who integrates the bypass drivers.

Some implementations of the ECU bypass drivers, as for the service-based bypass, allow the deactivation of the bypassed ECU function by the bypass user. In this case, the results of the bypassed ECU function obviously cannot be used as fallback. This must be considered when setting up a safety strategy.

11.5.3 Bypass Output Data

The ECU provider must guarantee that any value sent back by the bypass system leads to a predictable behavior of the ECU – the bypass output values must undergo the same range check and validation as the values calculated within the bypass.

As said before, the implementation of the ECU drivers must, in any case, ensure that bypass failures can be detected by the ECU and valid and safe fallback values are available at any time.

11.5.4 Message Copies

If the ECU software contains message copies, the bypass must be aware of them.

The usual implementations of the *hook-based bypass* are an exception to that rule. Here, the hooks (and thus the messages written to) are known before compilation, so that the hook code can take care of and use message copies if needed—individual code and addresses are used at each hook.

With the *service-based bypass*, users cannot choose variables and decide which message to write to before they set up the bypass experiment in ASCET-RP. Thus, the services in the ECU are generic code and do not know about specific message copies.

This requires two steps:

1. The ECU software provider must provide this message copy information in the A2L file (usually in encrypted and password-protected form).
2. If the message copy information is encrypted, the user of the bypass system receives a password from the ECU software provider. He must enter this password to decrypt the information and use it for system configuration.

If one of these two steps is missing (usually a wrong password is entered), the bypass system has no knowledge of message copies and reads from / writes to the original variable address, as it is declared in the `MEASUREMENT` declaration of the A2L file. For receive variables, this yields old data values. For send variables, the data value written by the bypass model may be overwritten by other parts of the ECU software. Both cases may cause bypass malfunction!

Another principal problem can arise with ECU software that contains message copies. The original code to create the message copies for an ECU task was based on the given message usage and generated appropriate code. By writing variable values into the ECU via bypass methods, the data flow changes, and a new or different message copy may become necessary. This can result in wrong variable values in the ECU software even at locations which are not directly related to the bypassed functions. Whether writing to a certain variable at a certain location (e.g. service point) may be dangerous or not, can be answered only by the ECU software supplier. This information cannot be declared in the A2L file.

11.6 Service-Based Bypass Specifics

Unlike the hook-based bypass where either the ECU or the bypass writes to the bypassed variable, the service-based bypass has an inherent possibility of data inconsistency, since both the ECU and the bypass write to the same value consecutively. Due to ECU realtime constraints, interrupts cannot be disabled to protect the sequence of writing the results of the ECU function and then writing the variable values of the bypass.

So, if a preemptive task of higher priority interrupts the tasks containing the bypass service, it will see the ECU value instead of the bypass value. The probability of this inconsistency depends on the distance between the two writes.

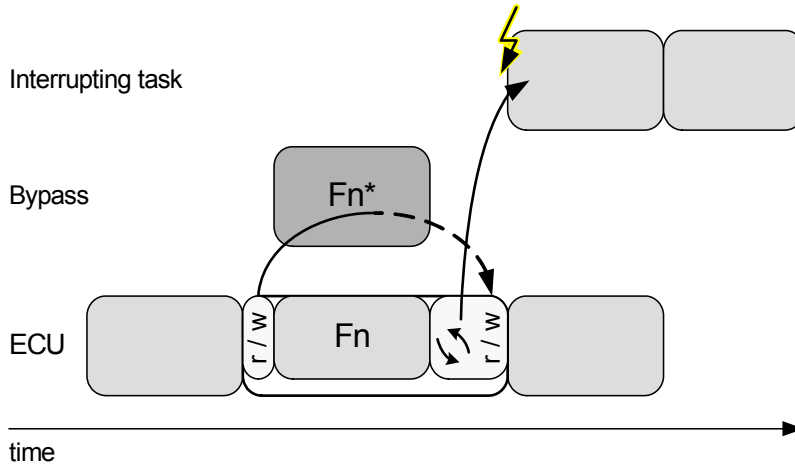


Fig. 11-4 Possible data inconsistency (the small arrows (↻) indicate a waiting time for bypass results)

A countermeasure to this problem is disabling the ECU function, so that only the bypass is writing to the bypassed ECU values (see below). Be aware that disabling the ECU function implies other safety constraints in case of bypass failures as discussed below.

11.6.1 Service Processes for the SBB Implemented as Service Functions

For SBB, the way of ECU implementation is to replace the ECU process to be bypassed by a container process that contains service function calls before and after it calls the original ECU process. This allows to call the ECU process under certain conditions only, e.g. to deactivate it in case of possible data consistency problems. To simulate the timing behavior of the disabled ECU process, a delay time can be configured.

Therefore, the suggested ECU implementation looks like this:

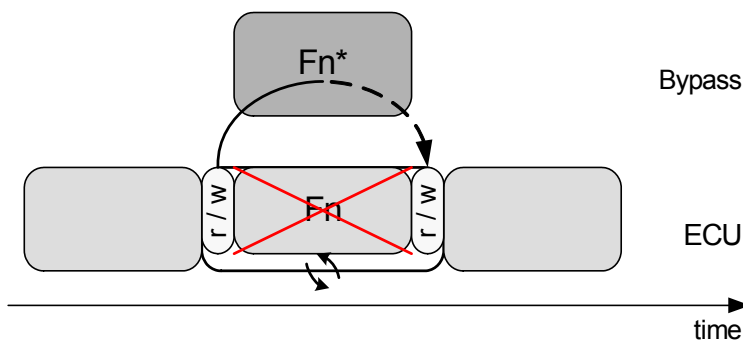


Fig. 11-5 Suggested SBB implementation

For setting up the bypass experiment in the Hardware Configurator and implementing the service point in the ECU software as container process, a service point is defined as

1. receiving data from the ECU (pre read action),


2. waiting for data to be sent (a timeout is defined),
3. sending data to the ECU (pre write action),
4. conditionally executing the original ECU process,
5. receiving data from the ECU (post read action),
6. waiting for data to be sent (a timeout is defined),
7. sending data to the ECU (post write action).

Each pre and post action can be freely configured or activated / deactivated.

11.6.2 Controlling the ECU Behavior from ASCET-RP

Upon setting up the ASCET-RP experiment, an initial setting of the control variables can be done in ASCET-RP. These values are written to the ETK on experiment initialization.

The ECU function can be controlled by the user in several ways (if the ECU drivers also provide this functionality):

- The ECU function can be deactivated in the service point editor of the Hardware Configurator ( column, see Fig. 11-6)
- The detection of a bypass error can be defined
- The bypass error behavior of the ECU code can be influenced

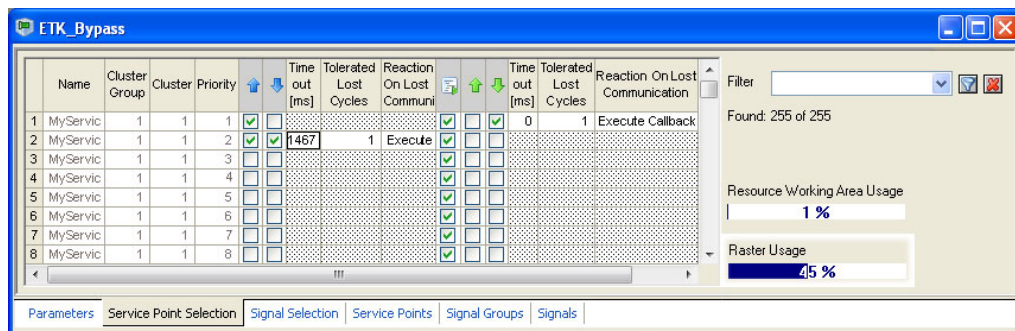


Fig. 11-6 Controlling ECU function execution from the Hardware Configurator
(The columns "Cluster Group" and "Cluster" are only available for SBB V3.
The "Raster Usage" display is only available for SBB V2.)

11.6.3 Summary

The implementation and integration of the ECU drivers must take care of the following:

- As the service-based bypass only overwrites ECU values with bypass values without preventing the ECU from writing these values, there is a possibility of data inconsistency which can lead to unpredictable behavior of the ECU.
- The Hardware Configurator provides a configuration variable to set the maximal number of tolerated lost cycles, e.g. how many ECU calculation cycles without receiving bypass values are tolerated before this is regarded as an error. But it is up to the provider of the ECU software to make sure missing bypass output values are detected.

- The Hardware Configurator provides a configuration variable to set a specific error behavior (if also supported by the ECU implementation) But it is up to the provider of the ECU software to make sure bypass failures or bypass deactivation can be detected by the ECU software! The configuration setting in the ASCET-RP GUI can then be used to choose between different provided error behaviors in the ECU.
- The Hardware Configurator allows disabling the bypassed ECU process. In this case, no ECU values can be used as fallback values for bypass failures! It is up to the provider of the ECU software to make sure sensible data is written to the variables if both the ECU process and the bypass is disabled.

12 **Appendix B: Compiler Switches and API Functions**

The descriptions of compiler switches and API functions are now available in the ASCET online help, section "INTECRIO Connectivity / ASCET-RP".

13 Appendix C: ETAS Network Manager

The ETAS Network Manager is used for creating a configuration that will be used by the ETAS IP Manager. The IP Manager is responsible for dynamic IP addressing of the ETAS hardware used in your network.

13.1 Overview

ETAS software supports different configurations for hardware access via Ethernet:

- Using multiple network adapters:
 - one network adapter for the company network,
 - one network adapter for the ETAS hardware.
- Using one network adapter:
 - automatic toggling between the company network and the ETAS hardware.

Tip

You do not require a separate network adapter to connect the ETAS hardware to your PC. You can use the same network adapter both for the company network and the ETAS network.

The ETAS Network Manager supports you in selecting the network adapter for the ETAS hardware.

The ETAS Network Manager gives you an overview of the network adapters available for your PC and the type of IP address assignment. If more than one network adapter is available in the system, you can select the network adapter to use for connecting the ETAS hardware to your PC. You can also specify the address range for the IP assignment for the ETAS hardware.

You do not need administrator rights to select the network adapter and the network environment configuration for the ETAS hardware. You can toggle between the ETAS network and the company network without rebooting your PC.

Tip

With Network Manager, you cannot create or modify the configuration for the network adapter. Instead modify the network settings of your PC via the Control Panel (see the documentation for your operating system). Please note that this requires administrator rights.

13.2 ETAS Hardware Addressing

The ETAS network allows you to connect several devices (including those that are the same type) to your PC. The connected devices are identified in the local ETAS network by their unique IP address.

An IP Manager integrated in the ETAS software looks up which IP addresses are available in a pre-configured address pool and assigns available IP addresses to the connected ETAS hardware.

The address range for the address pool is specified using the ETAS Network Manager.

13.3 Network Adapter Addressing

13.3.1 Type of Network Adapter Addressing

The type of network adapter addressing done within the company network depends on the operating system being used and the network adapter configuration:

Operating System	Type of Network Adapter Addressing			
	Manual	DHCP	DHCP+APIPA	DHCP+ alternative IP address
Windows Vista	yes	yes	yes	yes
Windows 7	yes	yes	yes	yes

The ETAS network supports the following types of network adapter addressing:

Operating System	Type of Network Adapter Addressing			
	Manual	DHCP	DHCP+APIPA	DHCP+ alternative IP address
Windows Vista	yes	no	yes	yes
Windows 7	yes	no	yes	yes

If you wish to use the network adapters both for the company network and the ETAS network, you cannot use the network adapters that exclusively support DHCP addressing for this dual operation.

Tip

DHCP can be used only in combination with APIPA or an alternative IP address!

13.3.2 Addressing the Network Adapter Manually

Addressing a network adapter depends on the operating system.

For instructions on addressing your PC's network adapter, see the documentation for your operating system.

To address the network adapter manually, you need administrator rights. Please contact your system administrator, if necessary.

If the network adapter is addressed manually, i.e., it has a static IP address, it may happen that you accidentally end up searching for or initialize ETAS hardware, although the PC is connected to the company network. The Network Manager allows you to stipulate that if this happens, you are to receive a warning before an IP address is assigned to an ETAS hardware.

13.3.3 Addressing the Network Adapter via DHCP

Addressing via DHCP requires that the DHCP server be available. Should the DHCP server not be available, or if there is no DHCP server (as in the ETAS network), the network adapter has not been configured.

In this instance, each operating system has a feature that automatically assigns the network adapter an IP address:

Windows Vista / Windows 7

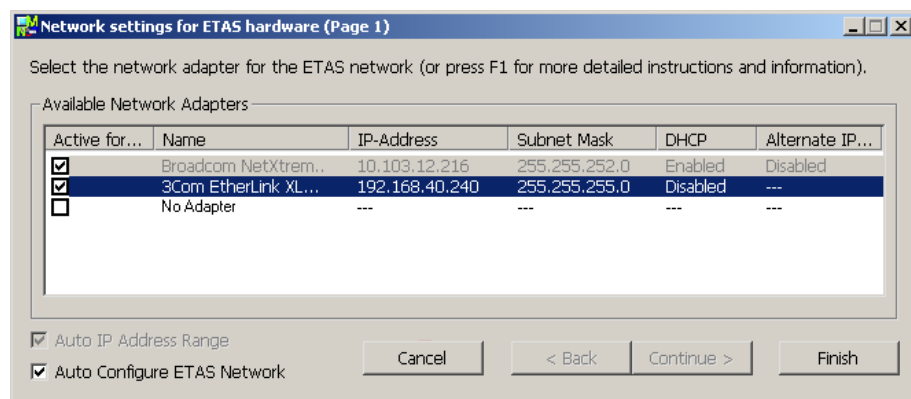
Windows Vista and Windows 7 automatically check whether there is a connection to the DHCP server. If there is none, it either assigns the IP address automatically via APIPA, or it uses the user-specified alternative IP address. The ETAS network always uses either the APIPA address or the alternative IP address.

When toggling between the DHCP network and ETAS hardware, make sure that the operating system is able to detect a connection failure because only then will reconfiguration be initiated. This may take up to 10 seconds. It takes the operating system 60 seconds to entirely reconfigure from a DHCP address to an APIPA address or to the alternative address. If the network adapter is once again connected to the DHCP network, configuring to a DHCP address takes place right after the connection has been detected.

Addressing a network adapter via DHCP without alternative addressing is not supported.

13.4 User Interface

13.4.1 "Network settings for ETAS hardware (Page 1)" Dialog Window



The following information on the available network adapters is displayed:

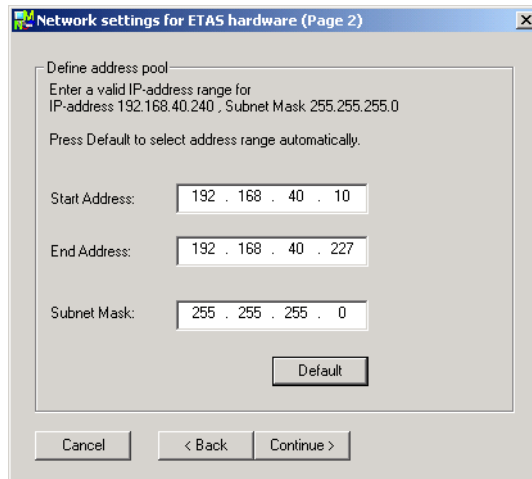
- **Active for ETAS Network** column
This column is only visible if the "Auto Configure ETAS network" checkbox is ticked. In the checkbox in this column you can determine which network adapters shall be enabled for autoconfiguration by the ETAS Network Manager.
- **Name** column
Name of the network adapter. This entry cannot be edited in this window.

- **IP Address** column
IP address of the network adapter. This entry cannot be edited in this window.
- **Subnet Mask** column
Setting for the subnet mask. This entry cannot be edited in this window.
- **DHCP** column
Shows whether the network adapter is configured for DHCP:
 - Enabled
The network adapter is configured for DHCP.
 - Disabled
The network adapter is configured with a fixed IP address.
- **Alternate IP Configuration** column
Shows the alternative IP address of the network adapter if it is configured for DHCP. This indication depends on the operating system being used.
 - APIPA
Automatic Private IP Addressing: method for automating the IP configuration for network connections
 - ---
An alternative IP address does not exist.
 - User defined
The user can define a user-specific alternative IP address (Windows Vista / Windows 7).
- **Auto IP Address Range** checkbox
If you tick this checkbox, the next configuration step is skipped, and the ETAS Network Manager automatically assigns default IP address ranges that will be used by the selected network adapter for addressing the ETAS hardware. If the IP address range is automatically changed by the IP Manager, a message is displayed in the system tray.
- **Auto Configure ETAS Network** checkbox
If you tick this checkbox, you can enable or disable several network adapters at once for auto-configuration through the IP Manager.
When you tick the checkbox, the "Auto IP Address Range" checkbox is activated. In the list of available network adapters, the "Active for ETAS Network" column is inserted, where you can determine which network adapters shall be available for auto-configuration through the ETAS Network Manager.
The ETAS Network Manager will go through the list (top-down) and use the first adapter which has a valid IP configuration for ETAS¹ and configure the IP address range automatically. If a configured network adapter fails, e.g. because the network adapter is being disabled or physically not

¹. An IP configuration is valid if the network adapter either uses a fix IP address, or if DHCP and APIPA are enabled.

available, the IP Manager will configure the next available network adapter automatically, and a message is displayed in the system tray, indicating the new configuration.

13.4.2 "Network settings for ETAS hardware (Page 2)" Dialog Window



In general, all values can be modified by directly typing them in the corresponding field, or by selecting the default setting from a list box.

The following network parameters can be set:

- **Start Address**
The first IP address in the IP address range for the ETAS hardware.
- **End Address**
The last IP address in the IP address range for the ETAS hardware.
- **Subnet Mask**
Associated Subnet Mask.

Reserved IP Addresses

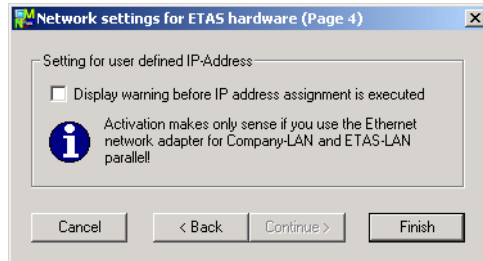
The following IP addresses are reserved for certain ETAS hardware in the IP address range that the ETAS hardware (192.168.40.1 - 192.168.40.254 with Subnet Mask 255.255.255.0) is currently using:

IP_Address	ETAS Hardware
192.168.40.10	ES1120
192.168.40.11	ES1130
192.168.40.12	ES780
192.168.40.13	Reserved
192.168.40.14	LABCAR-RTPC
192.168.40.15	ES1135

These addresses are assigned exclusively to these devices and thus may not be used for other ETAS hardware. This has to be taken into consideration when defining the address pool.

13.4.3 "Network settings for ETAS hardware (Page 4)" Dialog Window

This dialog window appears only if the selected network adapter is addressed manually.



The following parameters can be set:

- **Display warning before IP address assignment is executed**

Use this check box to specify that a warning be displayed before an IP address is assigned to an ETAS hardware device.

Tip

Enabling this warning is useful only if you want to run the PC both in the company network or on an ETAS measurement module in the ETAS network using this network adapter.

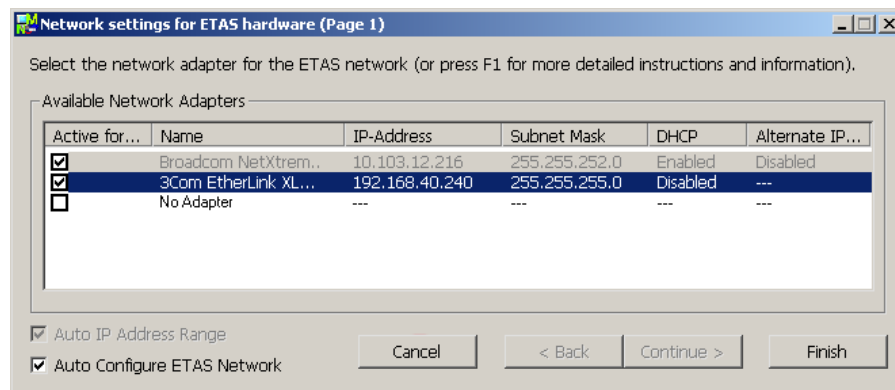
13.5 Configuring Network Addresses for ETAS Hardware

13.5.1 Adapter with Fixed IP Address

To start the Network Manager:

- In the Windows Start menu, go into the program folder of your ETAS software (below **Start** → **Programs** → **ETAS**), and select **ETAS Network Settings**.

The "Network settings for ETAS hardware (Page 1)" dialog window opens.



To select the network adapter:

- In the "Available Network Adapters" field, select the network adapters you want to use for the company network and the ETAS network.
You can select only network adapters of those types that are supported by the ETAS network.
- Click the **Continue** button.
The "Network settings for ETAS hardware (Page 2)" dialog window opens.

To define the address pool:

- Click the entry you want to modify in the "Start Address", "End Address" or "Subnet Mask" field.
- Edit the value directly (text input).

Or

- Click the **Default** button.
The Network Manager automatically enters the address range and the setting for the subnet mask. You may accept these settings or overwrite them.
- Click the **Continue** button.
The "Network settings for ETAS hardware (Page 4)" dialog window opens.

To set a user-defined IP address:

- Activate the **Display warning before IP assignment is executed** option if you want to specify that a warning be displayed before an IP address is assigned to the ETAS hardware.

Tip

Enabling this warning is useful only if you want to run the PC both in the company network or on an ETAS measurement module in the ETAS network using this network adapter.

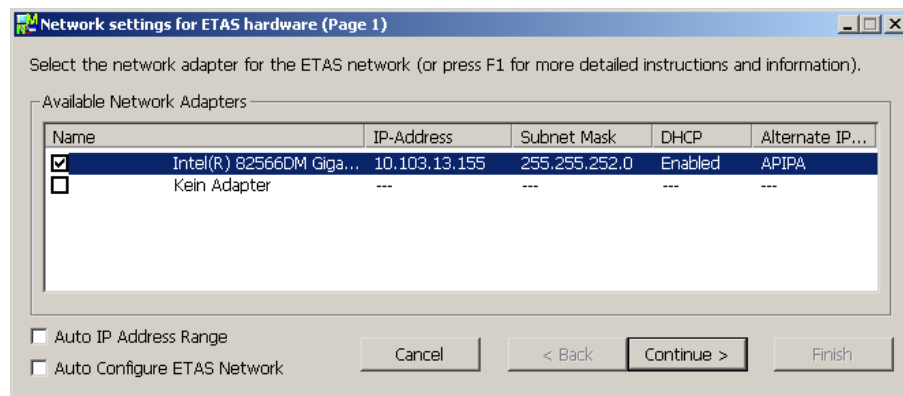
- Click the **Finish** button.
The configuration is finished and the dialog box is closed. The settings are saved.
- Restart the ETAS software to make the changes effective.
Restarting is necessary only if the ETAS software did not automatically invoke the configurator during a hardware search or initialization.

13.5.2 Adapter in DHCP Environment

To start the Network Manager:

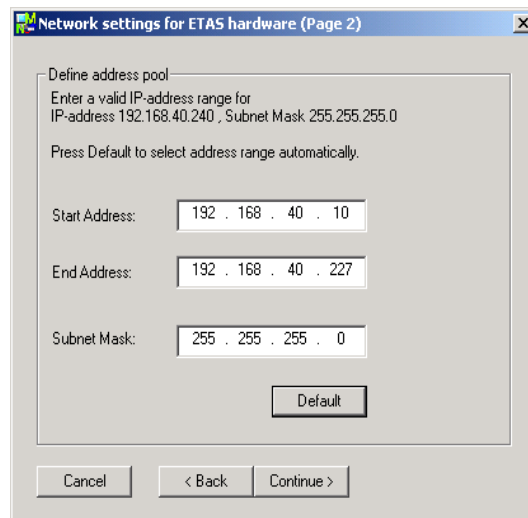
- In the Windows Start menu, go into the program folder of your ETAS software (below **Start** → **Programs** → **ETAS**), and select **ETAS Network Settings**.

The "Network settings for ETAS hardware (Page 1)" dialog window opens.



To select the network adapter:

- In the "Available Network Adapters" field, select the network adapters you want to use for the ETAS network.
You can select only those network adapters whose addressing type the ETAS network supports.
- Click the **Continue** button.
The "Network settings for ETAS hardware (Page 2)" dialog window opens.



To define the address pool:

- Click the entry you want to modify in the "Start Address," "End Address" or "Subnet Mask" field.
 - Edit the value using the keyboard (text entry).
- Or
- Click the **Default** button.
The Network Manager automatically enters the address range and the setting for the subnet mask. You may accept these settings or overwrite them.
 - If you address the network adapter via DHCP using an APIPA or alternative IP address, click the **Finish** button.
The configuration is finished and the dialog window is closed. The settings are saved.
 - Restart the ETAS software to make the changes effective.
Restarting is necessary only if the ETAS software did not automatically invoke the configurator during a hardware search or initialization.

13.6 Troubleshooting Ethernet Hardware Access

In certain cases you might have problems accessing ETAS hardware via the Ethernet interface. For instance, if APIPA, a mechanism for IP addressing, has not been enabled on your system, you cannot select a network adapter in the Network Manager. You can find descriptions of potential problems and their solutions in "Appendix D: Troubleshooting General Problems" on page 171.

14 Appendix D: Troubleshooting General Problems

This chapter gives some information of what you can do when problems arise that are not specific to an individual software or hardware product.

14.1 Problems and Solutions

14.1.1 Network Adapter cannot be selected via Network Manager

Cause: APIPA is disabled

The alternative mechanism for IP addressing (APIPA) is usually enabled on all Windows systems. Network security policies, however, may request the APIPA mechanism to be disabled. In this case, you cannot use a network adapter which is configured for DHCP to access ETAS hardware. The ETAS Network Manager displays a warning message.

The APIPA mechanism can be enabled by editing the Windows registry. This is permitted only to users who have administrator privileges. It should be done only in coordination with your network administrator.

To enable the APIPA mechanism:

- Open the Registry Editor:
 - Windows 7:
Click **Start** and then click **Run**. Enter `regedit` and click **OK**.
 - Windows Vista:
Click **Start**, enter `regedit` in the entry field, and press <ENTER>.

The registry editor is displayed.

- Open the folder `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\`
- Select **Edit** → **Find** to search for the key `IPAutoconfigurationEnabled`.

If you cannot find any instances of the registry key mentioned, the APIPA mechanism has not been disabled on your system, i.e. there is no need to enable it. Otherwise proceed with the following steps.

- Set the value of the key `IPAutoconfigurationEnabled` to 1 to enable the APIPA mechanism.
You may find several instances of this key in the Windows registry which either apply to the TCP/IP service in general or to a specific network adapter. You only need to change the value for the corresponding network adapter.
- Close the registry editor.

- Restart your workstation in order to make your changes take effect.

14.1.2 Search for Ethernet Hardware fails

Cause: The versions of the Hardware and the ETAS MC Software are not compatible

If you are using ETAS hardware with ETAS MC software, you can use the ETAS HSP Update Tool to check the firmware version of your hardware:

- Make sure you use the ETAS HSP Update Tool with the latest HSP (Hardware Service Pack) version.
- Also use the HSP Update Tool to check whether the hardware is compatible with the MC software used.
- Make sure any additional drivers for that hardware are installed correctly.

You can get the required HSP from the ETAS internet pages under www.etas.com.

If you still cannot find the hardware using the HSP Update Tool, check whether the hardware offers a Web interface and whether you can find using this interface. Otherwise check whether one of the following causes and solutions might apply.

Cause: Personal Firewall blocks Communication

For a detailed description on problems caused by personal firewalls and possible solutions see "Personal Firewall blocks Communication" on page 174.

Cause: Client Software for Remote Access blocks Communication

PCs or notebooks which are used outside the ETAS hardware network sometimes use a client software for remote access which might block communication to the ETAS hardware. This can have the following causes:

- A firewall which is blocking Ethernet messages is being used (see "„Cause: Personal Firewall blocks Communication" on page 172)
- By mistake, the VPN client software used for tunneling filters messages. As an example, Cisco VPN clients with versions before V4.0.x in some cases erroneously filtered certain UDP broadcasts.

If this might be the case, please update the software of your VPN client.

Cause: ETAS Hardware hangs

Occasionally the ETAS hardware might hang. In this case switch the hardware off, then switch it on again to re-initialize it.

Cause: ETAS Hardware went into Sleep Mode

In order to save power, some ETAS devices will go to sleep mode if they do not see that they are connected to another device/computer.

To solve that, connect your Ethernet cable from your computer to the "HOST"/"Sync In" port on the device. After the device turns on, connect to the device using the web interface and change the settings so that the device stays always on. Consult the device's manual for details on how to do that.

Cause: Network Adapter temporarily has no IP Address

Whenever you switch from a DHCP company LAN to the ETAS hardware network, it takes at least 60 seconds until ETAS hardware can be found. This is caused by the operating system's switching from the DHCP protocol to APIPA, which is being used by the ETAS hardware.

Cause: ETAS Hardware had been connected to another Logical Network

If you use more than one PC or notebook for accessing the same ETAS hardware, the network adapters used must be configured to use the same logical network. If this is not possible, it is necessary to switch the ETAS hardware off and on again between different sessions (repowering).

Cause: Device driver for network card not in operation

It is possible that the device driver of a network card is not running. In this case you will have to deactivate and then reactivate the network card.

To deactivate and reactivate the network card (Win Vista):

- To deactivate the network card first select in the Windows start menu the following item:
 - Windows Vista:
Control Panel → Network and Internet → Network and Sharing Center → Manage Network Connections
- Right-click on the used network adapter and select **Disable** in the context menu.
- In order to reactivate the network adapter right-click on it again and select **Enable**.

To deactivate and reactivate the network card (Win 7):

- To deactivate the network card, select **Control Panel → Device Manager** from the Windows start menu.
- In the Device Manager, open the tree structure of the entry **Network Adapters**.
- Click on the used connection to open its "<connection name> Status" dialog window.
- Right-click on the used network adapter and select **Disable** in the context menu.
- In order to reactivate the network adapter right-click on it again and select **Enable**.

Cause: Laptop power management deactivates the network card

The power management of a laptop computer can deactivate the network card. Therefore you should turn off power monitoring on the laptop.

To switch off power monitoring on the laptop:

- From the Windows Start Menu, select
 - Windows Vista:
Control Panel → System and Maintenance → Device Manager.
 - Windows 7:
Control Panel → Device Manager.
- In the Device Manager open the tree structure of the entry **Network Adapters**.
- Right-click on the used network adapter and select **Properties** in the context menu.
- Select the **Power Management** tab and deactivate the **Allow the computer to turn off this device to save power** option.
- Select the **Advanced** tab. If the property **Auto-sense** is included, deactivate it also.
- Click **OK** to apply the settings.

Cause: Automatic disruption of network connection

It is possible after a certain period of time without data traffic that the network card automatically interrupts the Ethernet connection. This can be prevented by setting the registry key `autodisconnect`.

To set the registry key `autodisconnect`:

- Open the Registry Editor.
- Select under `HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\lanmanserver\parameters` the Registry Key `autodisconnect` and change its value to `0xffffffff`.

14.1.3 Personal Firewall blocks Communication

Cause: Permissions given through the firewall block ETAS hardware

Personal firewalls may interfere with access to ETAS Ethernet hardware. The automatic search for hardware typically cannot find any Ethernet hardware at all, although the configuration parameters are correct.

Certain actions in ETAS products may lead to some trouble if the firewall is not properly parameterized, e.g. upon opening an experiment in ASCET or searching for hardware from within INCA or HSP.

If a firewall is blocking communication to ETAS hardware, you must either disable the firewall software while working with ETAS software, or the firewall must be configured to give the following permissions:

- Outgoing limited IP broadcasts via UDP (destination address 255.255.255.255) for destination ports 17099 or 18001
- Incoming limited IP broadcasts via UDP (destination IP 255.255.255.255, originating from source IP 0.0.0.0) for destination port 18001

- Directed IP broadcasts via UDP to the network configured for the ETAS application, destination ports 17099 or 18001
- Outgoing IP unicasts via UDP to any IP in network configured for the ETAS application, destination ports 17099 through 18020
- Incoming IP unicasts via UDP originating from any IP in the network configured for the ETAS application, source ports 17099 through 18020, destination ports 17099 through 18020
- Outgoing TCP/IP connections to the network configured for the ETAS application, destination ports 18001 through 18020

Tip

The ports that have to be used in concrete use cases depend on the hardware used. For more precise information on the port numbers that can be used please refer to your hardware documentation.

The Windows operating systems come with a built-in personal firewall. In addition, it is very common to have personal firewall software from third party vendors, such as Symantec, McAfee or Blacklce installed. The proceedings in configuring the ports might differ for each personal firewall software used. Therefore please refer to the user documentation of your personal firewall software for further details.

As an example for a firewall configuration, you will find below a description on how to configure the widely used Windows XP firewall if the hardware access is prohibited under Windows XP with Service Pack 2.

Solution for Windows XP Firewall, Users with Administrator Privileges

If you have administrator privileges on your PC, the following dialog window opens if the firewall blocks an ETAS product.



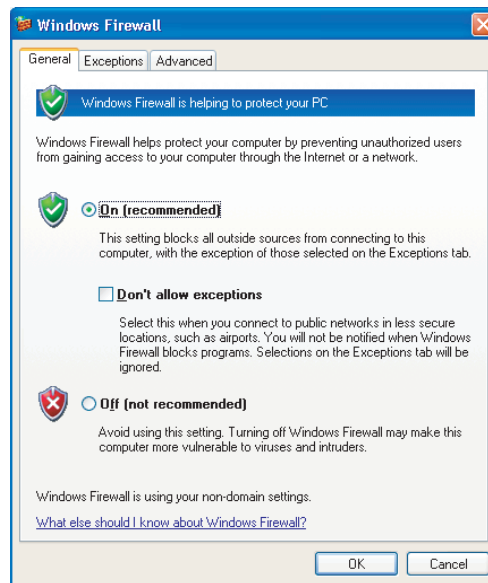
To unblock a product:

- In the "Windows Security Alert" dialog window, click on **Unblock**.
The firewall no longer blocks the ETAS product in question (in the example: ASCET). This decision survives a restart of the program, or even the PC.

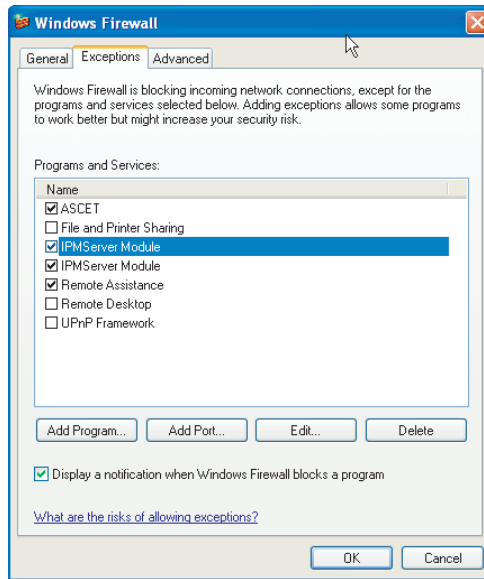
Instead of waiting for the "Windows Security Alert" dialog window, you can unblock ETAS products in advance.

To unblock ETAS products in the firewall control:

- From the Windows Start Menu, select **Settings** → **Control Panel**.
- In the control panel, double-click the **Windows Firewall** icon to open the "Windows Firewall" dialog window.

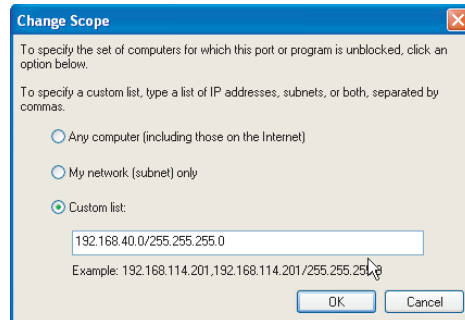


- In the "Windows Firewall" dialog window, open the "Exceptions" tab.



This tab lists the exceptions not blocked by the firewall. Use **Add Program** or **Edit** to add new programs, or edit existing ones.

- Make sure that the ETAS products and services you want to use are properly configured exceptions.
 - Open the "Change Setup" window.



- To ensure proper ETAS hardware access, make sure that at least the IP addresses 192 . 168 . 40 . xxx are unblocked.
- Close the "Change Setup" window with **OK**.

- Close the "Windows Firewall" dialog window with **OK**.

The firewall no longer blocks the ETAS product in question. This decision survives a restart of the PC.

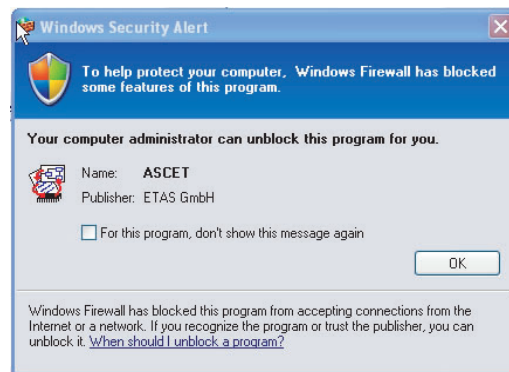
Solution for Windows XP Firewall, Users without Administrator Privileges

This section addresses users with restricted privileges, e.g., no system changes, write restrictions, local login.

Working with an ETAS software product requires "Write" and "Modify" privileges within the ETAS, ETASData, and ETAS temporary directories. Otherwise, an error message opens if the product is started, and a database is opened. In that case, no correct operation of the ETAS product is possible because the database file and some *.ini files are modified during operation.

The ETAS software has to be installed by an administrator anyway. It is recommended that the administrator assures that the ETAS program/processes are added to the list of the Windows XP firewall exceptions, and selected in that list, after the installation. If this is omitted, the following will happen:

- The "Window Security Alert" window opens when one of the actions conflicting with a restrictive firewall configuration is executed.



To unblock a program (no Admin privileges):

- In the "Windows Security Alert" dialog window, activate the option **For this program, don't show this message again**.
- Click **OK** to close the window.

An administrator has to select the respective ETAS software in the "Exceptions" tab of the "Windows Firewall" dialog window to avoid further problems regarding hardware access with that ETAS product.

15 ETAS Contact Addresses

ETAS HQ

ETAS GmbH

Borsigstraße 14

70469 Stuttgart

Germany

Phone: +49 711 3423-0

Fax: +49 711 3423-2106

WWW: www.etas.com

ETAS Subsidiaries and Technical Support

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

ETAS subsidiaries WWW: www.etas.com/en/contact.php

ETAS technical support WWW: www.etas.com/en/hotlines.php

Index

A

- ASCET options
 - "Hardware Configuration" node 13
 - "Hardware Connection" node 14
 - "Hardware" node 12
 - "Signal Mapping" node 13
- ASCET Rapid Prototyping 12
- Automatic Mapping 13

B

- bypass
 - concept 153–159
 - ETK (see also *ETK Bypass*) 45
 - hook-based 154
 - safety 155
 - service-based 154, 157
- bypass offset 54

C

- CAN database
 - import 37
- Compiler
 - GNU Cross ~ 16
 - precompiled header 15
 - QCC 16
 - select 14
 - use own ~ 15
- Converting old projects 18

D

- daisy chain
 - import configuration 37
- DISTAB method 48
- dT 19

E

- ERCOS^{EK} 7
- ES1000
 - memory pages 20
- ES1130
 - dT 19
- ES1135
 - dT 19
- ES910
 - dT 19
 - memory pages 20
- ETAS Contact Addresses 179
- ETAS Network 161
 - Addressing 162
 - Configure network adapter 166
 - DHCP 163
 - manual addressing 162
 - Network Manager 163
 - Reserved addresses 165
 - Search error 170
- ETAS network
 - Hardware Connection 14
- Ethernet interface 14

- ETK Bypass 45
 - ASCET project 46
 - data exchange 48
 - DISTAB method 48
 - Hardware Configuration 46
 - how it works 47
- Experimental target 7
- Experimenting with ASCET 20–28
 - acquisition task 24
 - assign element to measurement window 24
 - C code Debugger 27
 - open environment 23
 - run online~ 22–27
 - setting up an experiment 25
 - standalone mode 27
 - start experiment 25
 - start measurement 25
 - stop experiment 26
 - stop measurement 26
- Experimenting with INCA 28–35
 - Back-Animation 33
 - INCA database path 29
 - initiating a transfer 28
 - selecting a device 31
 - selecting a project 32
 - selecting a workspace 30
 - starting a transfer 33
- EXPORT Subdirectory 9
- F**
- FlexRay
 - import configuration 37
- G**
- GNU Cross Compiler 16
- H**
- hardware configuration
 - add hardware description 38
 - create (Hardware configurator) 37
 - migrate *.hwx 40, 41
- Hardware Configurator 36
 - add hardware description to hardware configuration 38
 - automatic mapping options 13
 - create configuration 37
 - create hardware description file 37
 - enable 36
 - migrate 1 project 40
 - migrate all projects 41
 - open 39
 - set up project 39
 - use INTECRIO 36
- Hardware Connection
 - with ETAS Network Manager 14
 - without ETAS Network Manager 14
- hardware description file
 - add to hardware configuration 38
 - create 37
- hardware options 12
 - "Hardware Configuration" node 13
 - "Hardware Connection" node 14
 - "Signal Mapping" node 13
- hardware systems 43
- hook-based bypass 154
- I**
- INCA
 - see *Experimenting with INCA*
- INTECRIO
 - Experimenting with ~ 35
 - use as Hardware Configurator 36
- L**
- LIN
 - import configuration 37
- N**
- network adapter configuration
 - in DHCP environment 168
 - with fixed IP address 166
- Network configuration
 - s. ETAS network
- O**
- Online experiment
 - acquisition task 24
 - assign element to measurement window 24
 - open experiment environment 23
 - running 22–27
 - select hardware 22
 - standalone 27
 - start 22
- P**
- precompiled header 15
- Product liability disclaimer 7

project

- add hardware configuration 39
- migrate all ~ to Hardware Configurator 41
- migrate to Hardware Configurator 40

Q

QCC compiler 16

R**RTPRO-PC**

- dT 19
- memory pages 20
- tutorial 133

S**Safety Instructions**

- technical state 8

service-based bypass 154

- specifics 157

T**target**

- set up interfaces (with ETAS Network Manager) 14
- set up interfaces (without ETAS Network Manager) 14

Target directory 12

tutorial 59–152

- ES1222 (CAN-IO) 62
- ES1303 81
- ES1325 (no trigger) 90
- ES1325 (trigger) 115
- RTPRO-PC 133