
ASCET Rapid Prototyping V5.4

Benutzerhandbuch

Copyright

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

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

© **Copyright 2006** ETAS GmbH, Stuttgart

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

Dokument EC012401 R5.4.2 DE

Inhalt

1	Einleitung	11
1.1	Komponenten	11
1.2	Installation	11
1.3	Aufbau des Handbuchs	12
1.4	Vereinbarungen	12
1.4.1	Dokumentationsvereinbarungen	12
1.4.2	Typografische Konventionen	13
2	Konfigurieren experimenteller Targets	15
2.1	Die Hardware-Optionen	16
2.2	Hardware-Anbindung mit dem ETAS Network Manager	18
2.2.1	Das Hardware-Auswahlfenster	19
2.3	Einrichten der Schnittstellen ohne ETAS Network Manager	21
2.4	Compiler auswählen	23
2.4.1	Verwendung eines eigenen Compilers	23
2.4.2	Umstellen auf den GNU Cross Compiler	24
2.4.3	Umstellen auf den Diab Data Compiler	27
2.5	Konfigurieren des Compilers	28
2.5.1	Allgemeine Argumente	28
2.5.2	Initialisierungseinträge	28

2.5.3	Compiler-Einträge.....	29
2.5.4	Linker-Einträge.....	29
2.5.5	Loader-Einträge.....	29
2.6	Einstellungen in der Datei ascetsd.ini.....	29
3	Hinweise zur Benutzung von ASCET-RP.....	31
3.1	Vorverarbeitung vorhandener Datenbanken.....	31
3.2	Umwandeln von Projekten für ES1000.1 nach ES1000.2/ ES1000.3.....	31
3.3	Verwendung von dT.....	33
4	Rapid-Prototyping-Experimente.....	35
4.1	Experimentieren mit ASCET.....	35
4.1.1	Die Benutzeroberfläche.....	35
4.1.2	Durchführen von Online-Experimenten.....	37
4.1.3	Standalone-Modus.....	43
4.2	Experimentieren mit INCA.....	44
4.3	Experimentieren mit INTECRIO.....	54
5	Realtime-Input-Output-Package.....	67
5.1	Einleitung.....	67
5.2	Architektur des RTIO-Package.....	67
5.2.1	Hardware Configuration Modul.....	68
5.2.2	Hardware Configuration Editor.....	69
6	Vorbereitende Maßnahmen.....	71
6.1	Hardware – Experimentalsystem ES1000.x.....	71
6.2	Besonderheiten der ES1135.....	73
6.2.1	Non-volatile RAM (NVRAM).....	73
6.2.2	Watchdog.....	83
6.2.3	LEDs.....	85
6.3	System-Software.....	86
6.3.1	System Root Path.....	86
6.3.2	C-Code-Modul.....	87
6.3.3	Projekt.....	87
7	HWC-Editor.....	89
7.1	Öffnen des HWC-Editors.....	89
7.2	Bedienelemente.....	90
7.2.1	Schaltflächen.....	90
7.2.2	Liste „Items“.....	92
7.2.3	Konfigurationsregister.....	93
7.2.4	Hauptmenü.....	95

7.2.5	Kontextmenü in der Liste „Items“	110
7.3	Konfigurationsregister	111
7.3.1	Allgemeine Hinweise	111
7.3.2	Standardoptionen im Register „Globals“	114
7.3.3	Standardoptionen im Register „Groups“	117
7.3.4	Standardoptionen im Register „Signals“	118
7.3.5	Standardoptionen im Register „Mappings“	120
8	Codegenerierung	123
8.1	HWC-Modul	123
8.1.1	Elemente	124
8.2	Prozessreihenfolge	126
9	Der ETK-BYPASS (ES1200/ES1201/ES1231/ES1232)	127
9.1	Definition des Begriffs ETK-Bypass	127
9.2	Hardwarekonfiguration eines ETK-Bypass	128
9.3	ASCET Projekt für den ETK-Bypass	129
9.4	Funktionsweise des ETK-Bypass	130
9.5	Datenaustausch zwischen Steuergerät und ETAS Experimentalsystem ..	131
9.6	Vorab benötigte Informationen und Daten	139
10	HWC-Items	143
10.1	Implementierte Items	143
10.2	ES1135-LED	144
10.2.1	Globals (ES1135-LED Device)	144
10.2.2	Groups (ES1135-LED Device)	145
10.2.3	Signals (ES1135-LED Device)	145
10.2.4	Mappings (ES1135-LED Device)	146
10.3	ES1201-ETK	146
10.3.1	Globals (ES1201-ETK Subsystem)	146
10.3.2	Globals (ETK-CTRL Subsystem)	147
10.3.3	Globals (ETK-BYPASS Device)	149
10.3.4	Groups (ETK-BYPASS Device)	158
10.3.5	Signals (ETK-BYPASS Device)	161
10.3.6	Mappings (ETK-BYPASS Device)	164
10.4	ES1222-CAN (CAN-IO)	164
10.4.1	Globals (ES1222-CAN Subsystem)	165
10.4.2	Globals (CAN-CTRL Subsystem)	166
10.4.3	Globals (CAN-IO Device)	169
10.4.4	Groups (CAN-IO Device)	175
10.4.5	Signals (CAN-IO Device)	177

	10.4.6	Mappings (CAN-IO Device)	180
10.5		ES1222-CAN (CAN Bypass Protokoll CBP)	180
	10.5.1	Lizenzrechtlicher Hinweis zum CAN Bypass Protokoll (CBP) . . .	180
	10.5.2	Hardwarekonfiguration eines CAN-Bypass	182
	10.5.3	Globals (CAN-Bypass Device)	183
	10.5.4	Groups (CAN-Bypass Device)	186
	10.5.5	Signals (CAN-Bypass Device)	188
	10.5.6	Mappings (CAN-Bypass Device)	189
10.6		ES1223-LIN	189
	10.6.1	Globals (ES1223-LIN Subsystem)	190
	10.6.2	Globals (LIN-CTRL Subsystem)	190
	10.6.3	Globals (LIN-IO Device)	192
	10.6.4	Groups (LIN-IO Device)	194
	10.6.5	Signals (LIN-IO Device)	195
	10.6.6	Mappings (LIN-IO Device)	196
	10.6.7	Laufzeitverhalten	196
10.7		ES1231.1-ETK	196
	10.7.1	Globals (ES1231-ETK Subsystem)	197
	10.7.2	Globals (ETK-CTRL Subsystem)	197
	10.7.3	Globals (ETK-BYPASS Device)	197
	10.7.4	Groups (ETK-BYPASS Device)	197
	10.7.5	Signals (ETK-BYPASS Device)	198
	10.7.6	Mappings (ETK-BYPASS Device)	199
10.8		ES1232-ETK	199
	10.8.1	Globals (ES1232-ETK Subsystem)	199
	10.8.2	ETK-CTRL-BAS Subsystem	200
	10.8.3	ETK-BYPASS Device	201
	10.8.4	100 Mbit/s für existierende Projekte (ETK-CTRL-BAS Subsystem)	201
	10.8.5	Globals (ETK-CTRL-ADV Subsystem)	206
	10.8.6	Globals (ETK-BYPASS-ADV Device)	211
	10.8.7	Groups (ETK-BYPASS-ADV Device)	213
	10.8.8	Signals (ETK-BYPASS-ADV Device)	217
	10.8.9	Mappings (ETK-BYPASS-ADV Device)	217
10.9		ES1300-AD	219
	10.9.1	Globals (ES1300-AD Device)	219
	10.9.2	Groups (ES1300-AD Device)	221
	10.9.3	Signals (ES1300-AD Device)	222
	10.9.4	Mappings (ES1300-AD Device)	222
10.10		ES1301-AD	222

10.10.1	Globals (ES1301-AD Device)	223
10.10.2	Groups (ES1301-AD Device)	224
10.10.3	Signals (ES1301-AD Device)	225
10.10.4	Mappings (ES1301-AD Device)	225
10.11	ES1303-AD	225
10.11.1	Globals (ES1303-AD Device)	226
10.11.2	Groups (ES1303-AD Device)	229
10.11.3	Signals (ES1303-AD Device)	231
10.11.4	Mappings (ES1303-AD Device)	231
10.12	ES1310-DA	231
10.12.1	Globals (ES1310-DA Device)	232
10.12.2	Groups (ES1310-DA Device)	233
10.12.3	Signals (ES1310-DA Device)	234
10.12.4	Mappings (ES1310-DA Device)	234
10.13	ES1320-CB (DIO)	234
10.13.1	Globals (ES1320-CB Subsystem)	235
10.13.2	Globals (DIO Device)	236
10.13.3	Groups (DIO Device)	237
10.13.4	Signals (DIO Device)	238
10.13.5	Mappings (DIO Device)	238
10.14	ES1325-DIO	238
10.14.1	Globals (ES1325-DIO Subsystem)	239
10.14.2	Globals (ES1325-Input Device)	243
10.14.3	Groups (ES1325-Input Device)	247
10.14.4	Signals (ES1325-Input Device)	253
10.14.5	Mappings (ES1325-Input Device)	253
10.14.6	Globals (ES1325-Output Device)	254
10.14.7	Groups (ES1325-Output Device)	255
10.14.8	Signals (ES1325-Output Device)	258
10.14.9	Mappings (ES1325-Output Device)	258
10.14.10	Globals (ES1325-LED Device)	259
10.14.11	Groups (ES1325-LED Device)	260
10.14.12	Signals (ES1325-LED Device)	261
10.14.13	Mappings (ES1325-LED Device)	261
10.15	ES1330-PWM	261
10.15.1	Globals (ES1330-PWM Subsystem)	262
10.15.2	Globals (PWM-COUNTER Device)	263
10.15.3	Groups (PWM-COUNTER Device)	266
10.15.4	Signals (PWM-COUNTER Device)	266
10.15.5	Mappings (PWM-COUNTER Device)	267

11	Tutorial	269
11.1	Tutorial – Experimentieren mit INTECRIO	272
11.1.1	Vorbereitungen	273
11.1.2	Transfer des Projekts	274
11.1.3	Experimentieren in INTECRIO	275
11.1.4	Back-Animation verwenden	278
11.2	Tutorial – ES1222 (CAN-IO)	282
11.2.1	Die ES1222 Karte	284
11.2.2	Beispielprojekt	286
11.2.3	Erstellen der Hardware-Konfiguration	287
11.2.4	HWC-Einstellungen für die ES1222 (CAN-IO) vornehmen	293
11.2.5	Hardware-Konfiguration speichern	304
11.2.6	Code für HWC-Modul erzeugen	306
11.2.7	Experimentieren mit dem Beispielprojekt	307
11.3	Tutorial – ES1303	311
11.3.1	Die ES1303 Karte	312
11.3.2	Beispielprojekt	313
11.3.3	Erstellen der Hardware-Konfiguration	314
11.3.4	HWC-Einstellungen für die ES1303 vornehmen	318
11.3.5	Hardware-Konfiguration speichern	325
11.3.6	Code für HWC-Modul erzeugen	325
11.3.7	Abschließende Aktionen	325
11.4	Tutorial – ES1325 (ohne Trigger)	326
11.4.1	Die ES1325 Karte	329
11.4.2	Beispielprojekt	331
11.4.3	Erstellen der Hardware-Konfiguration	332
11.4.4	HWC-Einstellungen für die ES1325 vornehmen	337
11.4.5	Hardware-Konfiguration speichern	354
11.4.6	Code für HWC-Modul erzeugen	354
11.4.7	Experimentieren mit dem Beispielprojekt	354
11.5	Tutorial – ES1325 (mit Trigger)	361
11.5.1	Die ES1325 Karte	363
11.5.2	Beispielprojekt	363
11.5.3	Erstellen der Hardware-Konfiguration	364
11.5.4	HWC-Einstellungen für die ES1325 vornehmen	369
11.5.5	Hardware-Konfiguration speichern	381
11.5.6	Code für HWC-Modul erzeugen	381
11.5.7	Experimentieren mit dem Beispielprojekt	381

12	ETAS Network Manager	389
12.1	Übersicht	389
12.2	ETAS-Hardware-Adressierung	390
12.3	Netzwerkadapter-Adressierung	390
12.3.1	Art der Netzwerkadapter-Adressierung	390
12.3.2	Manuelle Adressierung der Netzwerkadapter	391
12.3.3	Adressierung des Netzwerkadapters über DHCP	391
12.4	Bedienoberfläche	393
12.4.1	Dialogfenster „Network settings for ETAS hardware (Page 1)“	393
12.4.2	Dialogfenster „Network settings for ETAS hardware (Page 2)“	394
12.4.3	Dialogfenster „Network settings for ETAS hardware (Page 4)“	395
12.4.4	Dialogfenster „Network settings for ETAS hardware (Page 5)“	396
12.5	Netzwerkadapter für ETAS-Hardware konfigurieren	396
12.6	Probleme bei der Hardware-Anbindung über Ethernet	401
12.6.1	APIPA unter Windows 98 SE, 2000 oder XP deaktiviert	402
12.6.2	Personal Firewalls	403
13	Anhang: API-Funktionen	405
13.1	API-Funktionen (ERCOS ^{EK})	405
13.1.1	Betriebsmodi	406
13.1.2	Tasks	408
13.1.3	Systemzeit	409
13.1.4	Interruptbehandlung	411
13.1.5	dT-Abfrage	412
13.2	API-Funktionen (NVRAM)	413
13.3	API-Funktionen (Watchdog)	420
13.3.1	Watchdog-Konfiguration	421
13.3.2	Watchdog-Service	425
13.3.3	Interruptsteuerung	426
13.3.4	Watchdog Status	427
13.4	API-Funktionen (ES1135-LEDs)	429
13.5	Sonstige API-Funktionen	430
	Index	431

1 **Einleitung**

Zur Ausführung von Echtzeit-Software ist echtzeitfähige Experimentier-Hardware erforderlich. Das Softwarepaket ASCET **Rapid Prototyping V5.4** (ASCET-RP V5.4) dient zur Integration der ES1000, Experimental-Targets (E-Targets) in ASCET V5.1. Im Zusammenspiel mit I/O-Peripherie können auf Basis dieser Experimental-Targets leistungsstarke Entwicklungssysteme aufgebaut werden.

Neben der Lizenz für das verwendete Compiler Toolset enthält ASCET-RP Erweiterungen der ASCET Entwicklungsumgebung, u.a. zur homogenen Integration von Compiler und Linkeraufrufen, und den ERCOS^{EK} Betriebssystemkern für Experimental-Targets.

ASCET-RP enthält auch die Grundfunktionen der ASCET Basis-Software. Für die Modellierungsfunktionen wird außerdem ASCET-MD benötigt.

1.1 **Komponenten**

Die ASCET-RP V5.4 Installation enthält folgende Komponenten:

- Integration der Simulationscontroller ES1130 und ES1135¹;
- Integration der I/O-Boards inklusive ETK-Bypass und CAN-Bypass für die Targets ES1130 und ES1135 (RTIO-Package);
- GNU Compiler;
- Dokumentation und Beispiele.

1.2 **Installation**

Die ASCET-Basissoftware und die ASCET-RP-Software werden gemeinsam auf einer CD-ROM ausgeliefert. Um ASCET-RP V5.4 auf ihrem PC zu installieren, müssen Sie zunächst die ASCET-Basissoftware und anschließend die ASCET-RP-Software installieren.

Starten Sie dazu zuerst das Installationsprogramm `ASCET.exe` und anschließend das Installationsprogramm `ASCET-RP.exe` von der CD-ROM.

Details zur ASCET-RP V5.4 Installation finden Sie in der Release Note.

Beispieldateien

Die aus ASCET exportierten Beispieldatenbanken befinden sich nach der ASCET-RP V5.4 Installation im Unterverzeichnis `export` Ihrer ASCET Installation in den Dateien `RTIOTutorial.exp` und `INTECRIO_Tutorial.exp`.

¹ Wenn ein beliebiger Systemcontroller gemeint ist, wird im Handbuch die Bezeichnung `ES113x` verwendet.

1.3 Aufbau des Handbuchs

Das ASCET-RP V5.4 Handbuch besteht aus drei Hauptteilen:

- Allgemeingültiger Teil
- Realtime-Input-Output-Package (mit Bypass-Schnittstelle)
- Tutorial

Der allgemeingültige Teil richtet sich an alle ASCET-RP V5.4 Anwender. Sie finden hier Informationen über die Struktur, Installation und Benutzung des ASCET-RP V5.4.

In den darauffolgenden Kapiteln wird die Funktion und Bedienung des RTIO-Packages des ASCET-RP V5.4 vorgestellt und erklärt.

Das Tutorial enthält Beispielaufgaben zum Experimentieren mit INTECRIO sowie zur Arbeit mit verschiedenen Karten.

1.4 Vereinbarungen

1.4.1 Dokumentationsvereinbarungen

Anweisungen werden in einem aufgabenorientierten Format dargestellt, wie das folgende Beispiel veranschaulicht:

Erreichen eines Ziels:

- Führen Sie Aktion 1 aus.
Erklärungen werden unter einer Aktion gegeben.
- Führen Sie Aktion 2 aus.
- Führen Sie Aktion 3 aus.

Eine *Aufgabe* in diesem Handbuch ist eine Folge von Aktionen, die ausgeführt werden müssen, um ein bestimmtes Ziel zu erreichen. Der Titel einer Aufgabenstellung gibt für gewöhnlich das Ergebnis der Aktionen an, z. B. „Eine neue Komponente erstellen“, oder „Ein Element umbenennen“. Aufgabenstellungen enthalten oft Darstellungen des entsprechenden ASCET Fensters oder Dialogfensters, auf das sich die Aufgabe bezieht.

1.4.2 Typografische Konventionen

Die folgenden typografischen Konventionen werden in diesem Handbuch verwendet:

Wählen Sie Datei → Öffnen .	Menübefehle werden fett dargestellt.
Klicken Sie OK .	Schaltflächen werden fett dargestellt.
Drücken Sie <EINGABE>.	Tastaturbefehle werden in spitzen Klammern, in Kapitälchen dargestellt.
Das Dialogfenster „Datei öffnen“ erscheint.	Namen von Programmfenstern, Dialogfenstern, Feldern u.ä. werden in Anführungszeichen gesetzt.
Wählen Sie die Datei <code>setup.exe</code> aus.	Text in Auswahllisten auf dem Bildschirm, Programmcode, sowie Pfad- und Dateinamen werden in der Schriftart <code>Courier</code> dargestellt.
Eine Konvertierung zwischen den Datentypen logisch und arithmetisch ist <i>nicht</i> möglich.	Inhaltliche Hervorhebungen und neu eingeführte Begriffe werden <i>kursiv</i> gesetzt
Die OSEK-Gruppe (siehe http://www.osekvdx.org/) hat einige Standards entwickelt.	Links zu Internet-Dokumenten werden in <u>blauer Schrift und unterstrichen</u> dargestellt.

Wichtige Hinweise für den Anwender werden so dargestellt:

Hinweis

Wichtiger Hinweis für den Anwender.

2 Konfigurieren experimenteller Targets

ASCET Rapid Prototyping V5.4 (ASCET-RP V5.4) enthält die Compiler- und Linker-Werkzeuge, die zum Erzeugen ausführbarer Dateien für ein PowerPC-Target erforderlich sind, sowie eine Erweiterung der ASCET-Entwicklungsumgebung zur Integration der Hardware. Das Target selbst lässt sich in den Targetoptionen des Projekteditors auswählen. Auf diese Weise werden die Targets voll in ASCET integriert.

Die Beschreibung der Konfiguration des Compilers und des Linkers sowie der Schnittstelle zur tatsächlichen Target-Hardware erfolgt nicht direkt in ASCET, sondern wahlweise über den ETAS Network Manager oder mit Hilfe von *.ini-Dateien. Kapitel 2.1 beschreibt die Hardware-Optionen von ASCET-RP, die Anbindung der Hardware mit Hilfe des ETAS Network Managers ist in Kapitel 2.2 beschrieben. Kapitel 2.3 beschreibt die Konfiguration der Host-Schnittstelle über Änderungen der Datei `target.ini` sowie die Konfiguration der Ethernet-Schnittstelle.

Kapitel 2.4 und Kapitel 2.5 beschreiben Auswahl und Konfiguration des Compilers.

Kapitel 2.6 beschreibt einige Optionen der Datei `ascetsd.ini`, die sich nicht aus ASCET heraus ändern lassen. Zum Ändern dieser Einstellung muss die Datei bearbeitet werden.

Struktur der PowerPC E-Target Verzeichnisse

Durch die Installation des ASCET-RP V5.4 wird unter dem ASCET-Verzeichnis ein Targetverzeichnis mit verschiedenen PowerPC-Unterverzeichnissen angelegt. In diesen Unterverzeichnissen sind E-Target-spezifische Informations-, Konfigurations- und Bibliotheksdateien abgelegt.

In der nachfolgenden Tabelle sind die Unterverzeichnisse aufgelistet:

ASCET Unterverzeichnis	E-Target	Rechenknoten
..\Target\ES1130	ES1000.2/ ES1000.3 ^a	ES1130
..\Target\ES1135	ES1000.2/ ES1000.3	ES1135

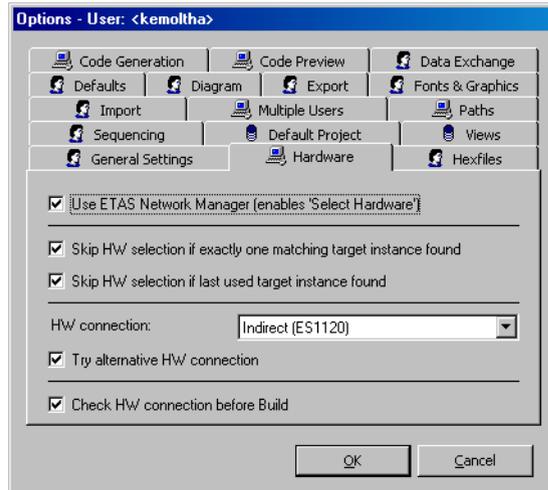
a: Wenn nicht explizit eine der beiden Varianten gemeint ist, wird im Handbuch die Bezeichnung *ES1000.x* oder *ES1000* verwendet.

Das Verzeichnis `..\Target\ES113x` enthält Dateien, die von beiden Rechenknoten genutzt werden.

Ein E-Target-abhängiges Umstellen des Systemhauptpfades (System Root Path) im Optionsfenster von ASCET ist nicht erforderlich.

2.1 Die Hardware-Optionen

ASCET-RP fügt für die Einrichtung der Hardware-Optionen im Optionsfenster von ASCET das Register „Hardware“ hinzu. Dort können Sie die folgenden Optionen einstellen. Sinnvollerweise wählen Sie die Einstellung, mit der Sie am häufigsten arbeiten.



- **Use ETAS Network Manager (enables 'Select Hardware')**

Mit dieser Option legen Sie fest, ob der ETAS Network Manager (siehe Kapitel 12) verwendet wird (aktiviert, Standardeinstellung) oder nicht. Wenn die Option aktiviert ist, werden im Projekteditor die Schaltfläche **Select Hardware** und die Menüfunktion **Extras → Select Hardware** verfügbar.

Hinweis

Die beiden folgenden Optionen sind nur von Bedeutung, wenn Sie den ETAS Network Manager verwenden.

- **Skip HW selection if exactly one matching target instance found**

Wenn die Option aktiviert ist (Standard), wird das Fenster „Experimental Target Hardware Selection“ nicht geöffnet, wenn beim Start des Experiments nur ein, zum Projekt passendes, Experimentaltarget gefunden wird.

Wenn die Option deaktiviert ist, hängt es von der folgenden Option ab,

ob das Fenster „Experimental Target Hardware Selection“ bei jedem Start des Experiments geöffnet wird. Dieses Fenster bietet alle an Ihrem Rechner angeschlossenen Experimentaltargets zur Auswahl an.

- **Skip HW selection if last used target instance found**

Wenn die Option aktiviert ist (Standard), wird das Fenster „Experimental Target Hardware Selection“ nicht geöffnet, wenn beim Start des Experiments das zuletzt mit diesem Projekt verwendete Target gefunden wird.

Wenn die Option deaktiviert ist, hängt es von der vorherigen Option ab, ob bei jedem Start des Experiments das Fenster „Experimental Target Hardware Selection“ geöffnet wird.

Hinweis

*Die beiden folgenden Optionen sind nur von Bedeutung, wenn Sie den ETAS Network Manager **nicht** verwenden.*

- **HW connection**

In diesem Kombikästchen wählen Sie aus, ob die ES1000 standardmäßig über die Steuereinheit ES1120 (`indirect (ES1120)`, Standard) oder über den Rechenknoten ES113x (`direct (ES113x)`) mit dem PC verbunden ist.

- **Try alternative HW connection**

Mit dieser Option legen Sie fest, ob auch eine Verbindung zu dem unter „HW connection“ nicht angegebenen Gerät gesucht werden soll (aktiviert, Standard) oder nur zu dem angegebenen Gerät (deaktiviert).

Hinweis

Die folgende Option ist in jedem Fall von Bedeutung.

- **Check HW connection before Build**

Mit dieser Option legen Sie fest, ob beim Start des Experiments (**Open Experiment**) *vor und nach* (aktiviert, Standard) oder *nur nach* (deaktiviert) dem Build-Prozess nach der Hardware gesucht wird, die in den Codegenerierungsoptionen des Projekts ausgewählt ist. Wird keine passende Hardware gefunden, wird eine Fehlermeldung ausgegeben.

Bei aktivierter Option können Sie den Fehler ohne Zeitverlust für den Build-Prozess korrigieren und eine geeignete Hardware hinzufügen.

Bei deaktivierter Option können Sie den Build-Prozess ohne Hardware durchführen, ohne dass eine Fehlermeldung erscheint.

2.2 Hardware-Anbindung mit dem ETAS Network Manager

Der ETAS Network Manager bietet Ihnen für die Hardware-Anbindung einige Vorteile.

- Sie können einen einzigen Netzwerkadapter für die ETAS-Hardware und Ihr Firmennetzwerk verwenden.
- Sie können individuelle Netzwerkadressen vergeben.
- Sie können Rechenknoten (ES113x) innerhalb eines ETAS-Hardware-Netzwerks auswählen.

Wie Sie mit dem ETAS Network Manager arbeiten, ist in Kapitel 12 beschrieben. An dieser Stelle finden Sie Informationen zur Hardware-Anbindung unter Verwendung des ETAS Network Managers.

Verwendung des ETAS Network Managers aktivieren:

- Wählen Sie im Komponentenmanager **Tools** → **Options**.
Das Dialogfenster „Options“ öffnet sich.
- Öffnen Sie das Register „Hardware“.
Die Optionen sind in Kapitel 2.1 beschrieben.
- Aktivieren Sie die Option **Use ETAS Network Manager (enable 'Select Hardware')**.
Wenn *nur* diese Option aktiviert ist, wird bei jedem Start eines Experiments das Hardware-Auswahlfenster „Experimental Target Hardware Selection“ geöffnet.
- Aktivieren Sie die Optionen **Skip HW selection if ***, wenn Sie unter den jeweiligen Bedingungen das Hardware-Auswahlfenster überspringen wollen.
- Aktivieren Sie die Option **Check HW connection before Build**, wenn die Hardware-Suche vor dem Build-Prozess stattfinden soll.

Zusätzlich zu dieser Automatik können Sie das Hardware-Auswahlfenster „Experimental Target Hardware Selection“ jederzeit aus dem Projekteditor manuell öffnen.

Hardware-Auswahlfenster manuell öffnen:

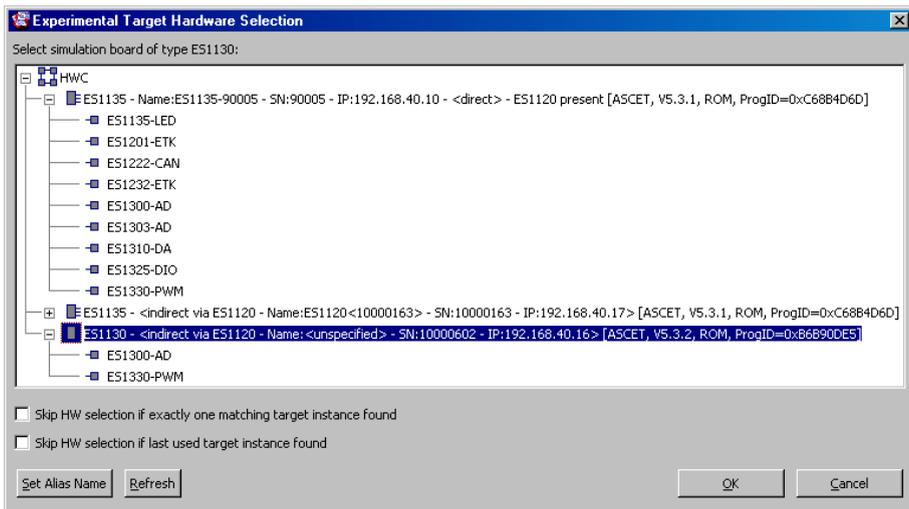
Hinweis

Die Schaltfläche **Select Hardware** und die Menüfunktion **Extras → Select Hardware** sind nur verfügbar, wenn die Option **Use ETAS Network Manager (enable 'Select Hardware')** aktiviert ist.

- Wählen Sie **Extras → Select Hardware** oder
- klicken Sie auf die Schaltfläche **Select Hardware**.
Das Hardware-Auswahlfenster öffnet sich.



2.2.1 Das Hardware-Auswahlfenster



Das Hardware-Auswahlfenster „Experimental Target Hardware Selection“ enthält folgende Elemente:

- Feld „Select simulation board of type <type>“¹

In diesem Feld werden unter dem Haupteintrag HWC (Symbol ) alle mit dem PC verbundenen Rechenknoten (ES113x, Symbol ) angezeigt. Die Bezeichnung des Rechenknotens enthält neben der Angabe,

¹. Der Namensteil <type> wird durch das Target vorgegeben, das im jeweiligen Projekt ausgewählt ist.

ob die ES113x direkt oder via ES1120 mit dem PC verbunden ist, verschiedene weitere Informationen; siehe Seite 20. Vorhandene Karten (Symbol ) werden unterhalb des Rechenknotens angezeigt.

Für das Experiment wählen Sie den Rechenknoten, den Sie in den Codegenerierungsoptionen des Projekts eingetragen haben..

- Optionen **Skip HW selection if exactly one matching target instance found** und **Skip HW selection if last used target instance found**

Diese beiden Optionen haben dieselbe Funktion wie die gleichnamigen Optionen im Register „Hardware“ des ASCET-Optionsfensters (s. Kapitel 2.1).

Die Einstellungen, die Sie hier vornehmen, werden ins Register „Hardware“ übernommen und umgekehrt.

- Schaltfläche **Set Alias Name**

Mit Hilfe dieser Schaltfläche können Sie einen beliebigen Namen für die ES113x oder ES1120 vergeben.

- Schaltfläche **Refresh**

Mit dieser Schaltfläche können Sie die Anzeige im Feld „Select simulation board of type <type>“ aktualisieren. Neu hinzugekommene oder angeschaltete Hardware wird anschließend angezeigt, entfernte/abgeschaltete Hardware verschwindet aus der Anzeige.

- Schaltflächen **OK** und **Cancel**

Mit **OK** übernehmen Sie die Auswahl, mit **Cancel** schließen Sie das Hardware-Auswahlfenster, ohne die Auswahl zu übernehmen.

Wenn der Rechenknoten direkt mit dem PC verbunden ist, sieht der Eintrag im Feld „Select simulation board of type <type>“ so aus:

```
ES113x - Name:<alias> - SN:<serial number> -  
IP:<IP address> - <direct> - ES1120 present  
[<SW>, <syslib version>, <boot mode>,  
ProgID=<ID>]
```

- ES113x ist die Bezeichnung des Rechenknotens.
- Name:<alias> ist der optionale Name, den Sie dem Rechenknoten geben können.
Wenn Sie keinen Namen vergeben, fehlt dieser Bestandteil.
- SN:<serial number> ist die Seriennummer der ES113x.
- IP:<IP address> ist die IP-Adresse der ES113x.
- <direct> zeigt an, dass die ES113x direkt mit dem PC verbunden ist.

- `ES1120 present` zeigt an, dass die ES1000 eine unverbundene ES1120 enthält.
Wenn die ES1000 keine ES1120 enthält, fehlt dieser Bestandteil.
- `<SW>` ist die Software, mit der Sie das Projekt auf die ES1000 geladen haben (z.B. ASCET oder INTECRIO).
- `<syslib version>` ist die verwendete Version der Hardware-Systembibliothek.
- `<boot mode>` zeigt an, ob das Projekt beim Einschalten der ES1000 aus dem Flash gestartet wurde (ROM), oder ob nach dem Einschalten ein Download stattfand (RAM).
- `ProgID=<ID>` ist der Identifier `<ID>`, der dem Projekt in der Software `<SW>` zugeordnet ist.

Wenn der Rechenknoten indirekt, also über die ES1120, mit dem PC verbunden ist, sieht der Eintrag so aus:

```
ES113x - <indirect via ES1120 - Name:<alias> -
      SN:<serial number> - IP:<IP address>
      [<SW>, <syslib version>, <boot mode>, ProgID=<ID>]
```

- `ES113x` ist die Bezeichnung des Rechenknotens.
- `<indirect via ES1120 ...>` zeigt an, dass die ES113x indirekt mit dem PC verbunden ist.
- `Name:<alias>` ist der optionale Name, den Sie der ES1120 geben können.
- `SN:<serial number>` ist die Seriennummer der ES1120.
- `IP:<IP address>` ist die IP-Adresse der ES1120.
- `<SW>`, `<syslib version>`, `<boot mode>` und `ProgID=<ID>` haben dieselbe Bedeutung wie bei einer direkten Verbindung.

2.3 Einrichten der Schnittstellen ohne ETAS Network Manager

Für spezielle Anwendungsfälle gibt es die Möglichkeit, wie in den früheren Versionen von ASCET-RP ohne den ETAS Network Manager zu arbeiten. In diesem Fall wird die Ethernet-Schnittstelle mit Hilfe der Datei `target.ini` des verwendeten Targets für ASCET eingerichtet. Die Dateien befinden sich in den Unterverzeichnissen `..\Target\ES1130` und `..\Target\ES1135`.

Festlegen der ES1000-Anbindung:

- Wählen Sie im Komponentenmanager **Tools** → **Options**.
Das Dialogfenster „Options“ öffnet sich. Die Optionen sind in Kapitel 2.1 beschrieben.
- Öffnen Sie das Register „Hardware“.
- Deaktivieren Sie die Option **Use ETAS Network Manager (enable 'Select Hardware')**.
- Wählen Sie im Kombikästchen „HW connection“ den geeigneten Eintrag für Ihre ES1000.
- Aktivieren Sie die Option **Try alternative HW connection**, wenn auch eine Verbindung zu dem unter „HW connection“ nicht angegebenen Gerät gesucht werden soll.
Wenn die Option deaktiviert ist, wird nur eine Verbindung zu dem angegebenen Gerät gesucht.
- Aktivieren Sie die Option **Check HW connection before Build**, wenn beim Start eines Experiments *vor* und nach dem Build-Prozess eine Hardware-Verbindung gesucht werden soll.
Wenn die Option deaktiviert ist, wird die Verbindung nur *nach* dem Build-Prozess gesucht.
- Klicken Sie auf **OK**, um die Einstellungen zu übernehmen.

Entsprechend Ihrer Auswahl wird für die ASCET-Experimentierumgebung aus der Datei `target.ini` im Unterverzeichnis des verwendeten Targets Ihrer ASCET-Installation die Variable für die IP-Adresse ausgewählt.

- Ist der ASCET Host-PC mit der Steuereinheit (ES1120) der ES1000.x verbunden, wird folgende Variable verwendet:

– **ES1130**

```
IndirectIpAddress=192.168.40.10  
;Default IP-Address for ES1120.x
```

– **ES1135**

```
IndirectIpAddress=192.168.40.10  
;Default IP-Address for ES1120.x
```

- Ist der ASCET Host-PC mit dem Rechenknoten (ES113x) der ES1000.x verbunden, wird folgende Variable verwendet:

– **ES1130**

```
DirectIpAddress=192.168.40.11  
;Default IP-Address for ES1130.x
```

– **ES1135**

```
DirectIpAddress=192.168.40.15  
;Default IP-Address for ES1135.1
```

Eine Ethernet-Dokumentation mit Hinweisen zur Installation und Konfiguration der Netzwerkschnittstelle Ihres PCs zum Anschluss des ES1000.x-Systems wird von ETAS bereitgestellt. Wenn Sie die ASCET-Manuals installiert haben, befinden sich die folgende Ethernet-Dokumentation im Unterverzeichnis `..\ETAS\ETASManuals\ASCET V5.1\ES1000:`

- `ES1000 Ethernet Installationanweisung.pdf` (Installation)

2.4 Compiler auswählen

Für die Targets ES1130 und ES1135 ist der GNU Cross Compiler (in der Programmoberfläche: `GNU-C (PowerPC)`) in ASCET-RP integriert. Für das Target ES1130 kann alternativ auch der Diab Data Compiler verwendet werden (nicht im Lieferumfang enthalten).

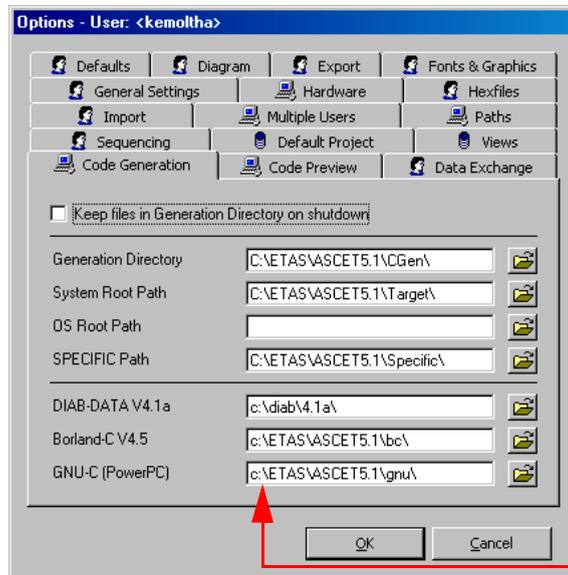
Target	Compiler
ES1130	GNU Cross Compiler, Diab Data V4.1a
ES1135	GNU Cross Compiler

Tab. 2-1 Übersicht Target/Compiler

2.4.1 Verwendung eines eigenen Compilers

Standardmäßig wird mit ASCET-RP V5.4 die MS-DOS Version des GNU Cross Compilers ausgeliefert. Falls ein eigener Compiler verwendet werden soll, so ist dies unter Berücksichtigung der folgenden Punkte möglich.

1. Unter ASCET muss im Fenster „Options“ der Pfad der Codegenerierungsoption GNU-C (PowerPC) auf den neuen Compiler eingestellt werden.



2. In der Datei `comptool.ini` des entsprechenden Targets (ES1130 oder ES1135) muss die Variable `RelativeToolPath` von `RelativeToolPath=\powerpc-eabi\bin` auf (beispielsweise) `RelativeToolPath=\win32\bin` verändert werden.

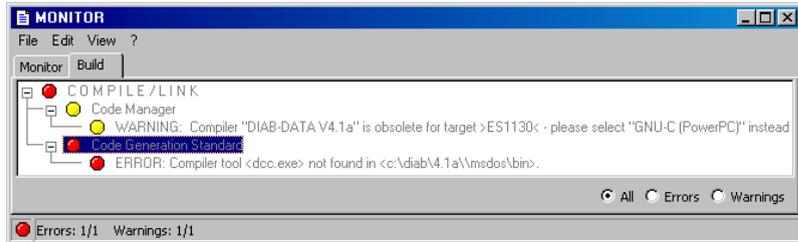
Hinweis

Pfade und Systemvariablen sollten nicht auf eine andere als die unter ASCET zu verwendende Version des Compilers verweisen.

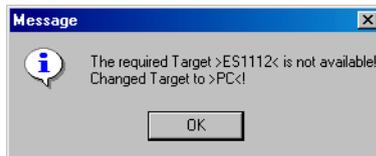
2.4.2 Umstellen auf den GNU Cross Compiler

Auch wenn kein Diab Data Compiler installiert ist, wird im ASCET-Optionsfenster im Register „Code Generation“ ein Standardpfad angegeben. Ebenso kann der Diab Data Compiler in den Targetoptionen eines Projekts ausgewählt werden.

Wenn Sie ein älteres Projekt bearbeiten, das das Target ES1130 mit dem Diab Data Compiler verwendet, wird daher beim Öffnen *keine* Fehlermeldung angezeigt. Erst beim Aufruf des Compilers während der Generierung von ausführbarem Code wird im ASCET-Monitorfenster eine Fehlermeldung angezeigt.



Wenn Sie ein älteres Projekt bearbeiten, das das Target ES1112 verwendet, wird bereits beim Öffnen eine Fehlermeldung ausgegeben und das nicht mehr vorhandene Target durch das Target PC ersetzt.



Sie müssen in beiden Fällen das Projekt auf eine geeignete Kombination aus Target und Compiler umstellen. Gehen Sie dazu wie folgt vor.

Umstellen auf den GNU-Compiler:



- Wählen Sie im Projekteditor die Schaltfläche **Specify Code Generation Options**.

Das Fenster „Settings for:“ öffnet sich.

- Wählen Sie im Register „Build“ das Target **ES1130** oder **ES1135** und den Compiler **GNU-C (PowerPC)**.
- Klicken Sie **OK**.

Je nachdem, welches Target das Projekt verwendet hat, müssen Sie noch den C-Code umkopieren. Das Verfahren ist unter „Umkopieren von Betriebssystemeinstellungen und C-Code:“ beschrieben.

- Wählen Sie abschließend **Component** → **Touch** → **Recursive**, damit alle Bestandteile des Projektes beim nächsten Durchlauf neu kompiliert werden.

Umkopieren von Betriebssystemeinstellungen und C-Code:

- Wählen Sie im Projekteditor die Registerkarte „OS“.
- Wählen Sie im Projekteditor **Operating System** → **Copy From Target**.

Das Fenster „Selection Required“ öffnet sich.



- Wählen Sie im Fenster „Selection Required“ das ursprüngliche Target des alten Projekts aus.
 - Klicken Sie **OK**.
- Der Betriebssystem-Code wird vom Target ES1130 oder ES1135 kopiert.
- Wählen Sie im Projekteditor **Extras** → **Copy C-Code From**.

Das Fenster „Selection Required“ öffnet sich.

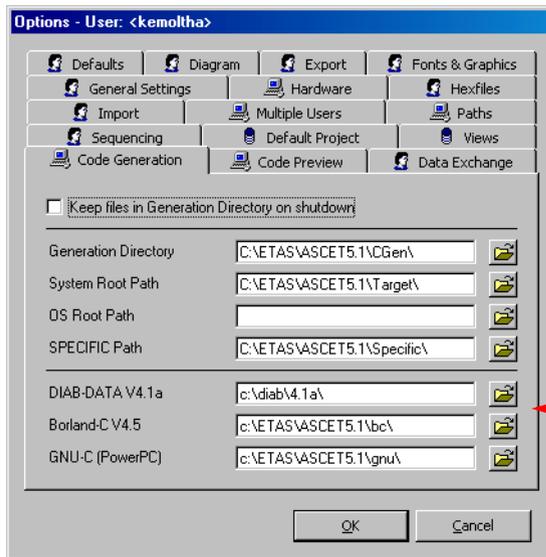


- Wählen Sie im Fenster „Selection Required“ die Kombination aus Target und Experiment, die im alten Projekt verwendet wurde.
- Klicken Sie **OK**.

Der Code wird von der ursprünglichen Kombination aus Target und Experiment zu den aktuellen Einstellungen kopiert.

2.4.3 Umstellen auf den Diab Data Compiler

Bis zur Version 4.2 des TIPEXP wurde standardmäßig die MS-DOS Version des Diab Data 4.1a Compilers ausgeliefert. Der Diab Data Compiler kann für das Target ES1130 auch mit ASCET-RP V5.4 verwendet werden. Er muss allerdings separat erworben und installiert werden.



Hinweis

Das Target ES1135 kann **nicht** mit dem Diab Data-Compiler zusammen verwendet werden.

2.5 Konfigurieren des Compilers

Für jedes mit ASCET installierte Target (einschließlich des Targets PC) wird ein Targetverzeichnis erstellt. Dieses Targetverzeichnis enthält alle targetspezifischen Dateien. Darunter befindet sich auch die Compiler-Konfigurationsdatei `comptool.ini`. Bei Änderungen an dieser Datei muss man Vorsicht walten lassen, da sie Auswirkungen auf das Gesamtverhalten der Compiler- und Linker-Arbeitsweise hat.

2.5.1 Allgemeine Argumente

Der Compiler verwendet mehrere Argumente, die wie folgt definiert sind:

- `%bp%` ist der Compiler-Basispfad, z. B. `c:\ETAS\ASCET5.1\bc`
- `%lp%` ist der Compiler-Bibliothekspfad, z. B. `c:\ETAS\ASCET5.1\bc\libs`
- `%ip%` ist der Compiler-include-Pfad, z. B.
`c:\ETAS\ASCET5.1\bc\include`
- `%tp%` ist der Compiler-Tool-Pfad, z. B. `c:\ETAS\ASCET5.1\bc\bin`
- `%tarbp%` ist der Targetbasispfad, z. B. `c:\ETAS\ASCET5.1\target`
- `%tarbsp%` ist der targetspezifische Pfad, z. B.
`c:\ETAS\ASCET5.1\target\pc`
- `%fn%` ist der Name der zu bearbeitenden Datei
- `%LD%` ist/sind das/die Zeilenbegrenzungszeichen
- `%SP%` ist das Leerzeichen

Einträge mit mehreren Argumenten müssen in einer durch Kommas abgetrennten Liste erfolgen. Leerzeichen davor und danach werden ignoriert (werden sie benötigt, muss `%SP%` eingefügt werden).

2.5.2 Initialisierungseinträge

- `DOSVariableSettings`: Benötigen die Compiler/Linker-Werkzeuge eine DOS-Variable, so kann sie hier eingestellt werden, z. B.
`DOSVariableSettings=%lp%` für die INMOS-Werkzeuge.
- `FilesInWorkingDirectory`: Einige Compiler benötigen spezielle Dateien im Arbeitsverzeichnis. Dieses Arbeitsverzeichnis (gewöhnlich mit `cgen` bezeichnet) wird jedoch bei jedem Schließen von ASCET gelöscht, wenn in den ASCET-Optionen (Register „Code Generation“) nichts anderes angegeben wurde (siehe Kapitel 2.2.1 im ASCET-Benutzerhandbuch). Die hier angegebenen Dateien werden vor dem Kompilieren automatisch in dieses Verzeichnis kopiert, z. B.
`FilesInWorkingDirectory = %tarbp%\dos4gw.exe`

2.5.3 Compiler-Einträge

- `CompilerCall1`: Damit wird der Compiler-Aufruf spezifiziert, der nach jedem Kompilierungslauf ausgeführt wird. Sie können mehrere Compiler-Aufrufe mit `CompilerCall2`, `CompilerCall3` usw. spezifizieren.
- `CompilerErrorKeywords`: Enthält die Fehlerausgabe des Compilers einen der vorgegebenen Strings, werden diese ausgewählt und in ASCET angezeigt.
- `GlobalIncludeFiles`: Hier werden die include-Dateien spezifiziert, die bei jedem Kompilieren zur Anwendung kommen. Die Header-Datei wird vor jedem Kompilieren gelesen. Hier kann der Benutzer seine eigenen Header-Dateien einfügen.
- `CompilerResultExtension`: Die Erweiterung des Kompilierergebnisses, gewöhnlich `.obj`.

2.5.4 Linker-Einträge

- `LinkCall1`: Damit wird der Linker-Aufruf spezifiziert, der nach dem Linking-Lauf ausgeführt wird. Sie können mehrere Linker-Aufrufe mit `CompilerCall2`, `CompilerCall3` usw. spezifizieren.
- `LinkerErrorKeywords`: Enthält die Fehlerausgabe des Linkers einen der vorgegebenen Strings, werden diese ausgewählt und in ASCET angezeigt.
- `LinkerResultExtension`: Die Erweiterung des Linking-Ergebnisses, z.B. `.dll`.
- `ObjectLibraries`: Die Bibliotheken werden bei jedem Linker-Lauf mit der Ausgabe verbunden.
- `ObjectFiles`: Diese Dateien werden bei jedem Linker-Lauf mit der Ausgabe verbunden.

2.5.5 Loader-Einträge

Loader-Einträge werden noch nicht unterstützt.

2.6 Einstellungen in der Datei `ascetsd.ini`

Die Datei `ascetsd.ini` finden Sie im Verzeichnis `ETASData\ASCET5.1` Ihrer ASCET-Installation. Folgende Einstellungen in der Datei lassen sich nicht aus ASCET heraus verändern:

- `Bubbledelaytime`: Spezifiziert den Zeitverzug in Millisekunden vor dem Erscheinen einer Werkzeugspitze.

- `MeasureCycleTime`: Spezifiziert die Zeit in Millisekunden zum Übertragen von Messwerten für numerische und Balkenanzeigen an den Host. Bei auftretenden Problemen mit L1-Kommunikation zwischen Host und Target ist dieser Wert zu erhöhen.
- `ExitWithPrompter`: Wenn `yes` vorgegeben wird, werden Sie bei jedem Verlassen der Experimentierumgebung aufgefordert, die Umgebung zu speichern. Wird `no` vorgegeben, erscheint keine Aufforderung (und die Umgebung wird nicht gespeichert).

Alle sonstigen Einstellungen sind entweder intern (sie sollten nicht verändert werden) oder lassen sich aus ASCET heraus ändern. Allgemein ist bei Veränderungen an den `*.ini`-Dateien Vorsicht geboten, da diese das Gesamtverhalten des Werkzeugs beeinflussen können.

3 Hinweise zur Benutzung von ASCET-RP

3.1 Vorverarbeitung vorhandener Datenbanken

ASCET-Datenbanken, die mit ASCET-Versionen vor V4.1.1 erstellt wurden, müssen zunächst mindestens mit einer ASCET-Version V4.1.1 gespeichert werden, bevor sie mit ASCET V5.1 geöffnet und konvertiert werden können.

Hinweis

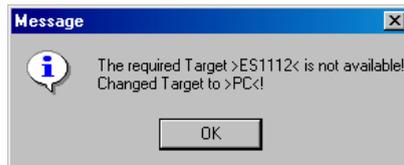
Ausführliche Hinweise zum Konvertieren sehr alter ASCET-Projekte (mit TIP-Exp V3.x und älter, Target PPC) befinden sich z.B im TIPExp V4.4 Handbuch.

3.2 Umwandeln von Projekten für ES1000.1 nach ES1000.2/ ES1000.3

ASCET-Projekte, die für das E-Target ES1000.1 erstellt wurden, müssen für das E-Target ES1000.2/ ES1000.3 umgestellt werden. Für die Umwandlung müssen folgende Schritte ausgeführt werden.

Umwandlung eines ASCET-Projekts für ES1000.1 nach ES1000.2/ ES1000.3

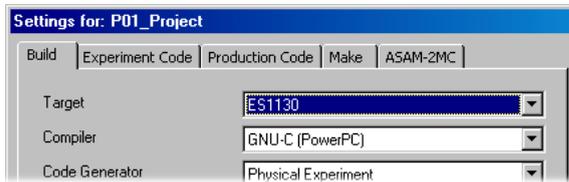
- Laden Sie das ASCET-Projekt.
Eine Meldung wird angezeigt, dass das Target ES1112 nicht mehr verfügbar ist.



- Bestätigen Sie die Meldung mit **OK**.
Das Projekt wird geöffnet; anstelle des nicht vorhandenen Targets wird das Target >PC< ausgewählt.



- Klicken Sie auf die Schaltfläche **Specify Code Generation Options**.

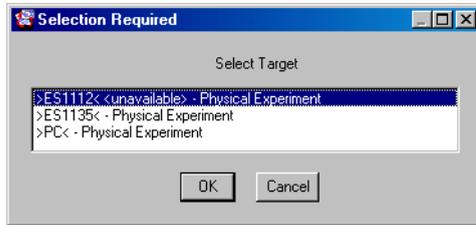


- Wählen Sie im Fenster „Settings for“, Register „Build“, folgende Optionen:
Target: >ES1130< oder >ES1135< und
Compiler: GNU-C (PowerPC).
- Klicken Sie auf **OK**, um das Fenster „Settings for“ zu schließen.
- Wählen Sie im Projekteditor das Register „OS“.
- Wählen Sie im Projekteditor **Operating System** → **Copy From Target**.



- Wählen Sie im Fenster „Selection Required“ das ursprüngliche Target und klicken Sie **OK**.
Der Betriebssystem-Code wird vom Target ES1112 zum Target ES1130 oder ES1135 kopiert.

- Wählen Sie im Projekteditor **Extras** → **Copy C-Code From**.



- Wählen Sie im Dialogfenster „Selection Required“ die Kombination aus Target **>ES1112<** und Experiment, die im alten Projekt verwendet wurde.
- Klicken Sie **OK**.

Der Code wird von der ursprünglichen Kombination aus Target und Experiment zu den aktuellen Einstellungen kopiert.

Das Projekt kann nun für das ES1000.2/ ES1000.3 System bearbeitet werden.

3.3 Verwendung von dT

Für das Target ES113x ist das Betriebssystem ERCOS^{EK} implementiert. ERCOS^{EK} bietet die Möglichkeit, auf die Zeit dT , die zwischen dem letzten und vorletzten Aufruf der laufenden Task verstrichen ist, zuzugreifen. Dabei bezieht sich dT immer auf die Task, in der die Größe verwendet wird (siehe Abb. 3-1).

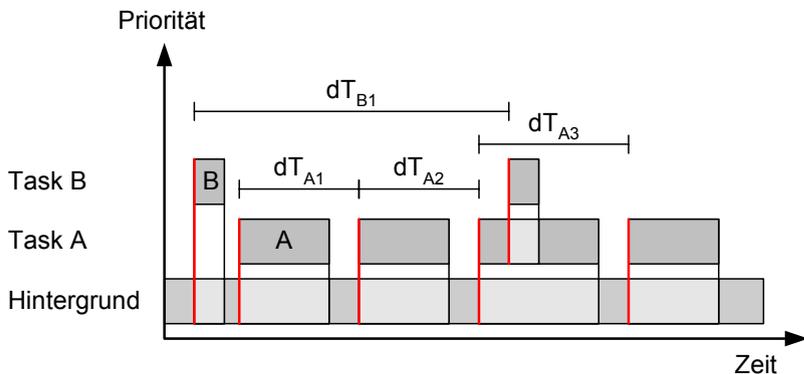


Abb. 3-1 dT – Schema

In ERCOS^{EK} ist `dT` eine globale Variable vom Typ `uint32`. Sie wird in einer der ERCOS^{EK} Headerdateien deklariert und enthält den Wert für die jeweils aktuelle Task in Einheiten von System-Ticks.



Von ASCET aus kann auf `dT` zugegriffen werden. Dazu gibt es in den Editoren die Schaltfläche **dT**, mit deren Hilfe ein Element (`xea164`) angelegt wird, das die Zeit in Einheiten von Sekunden enthält.

Wenn ein Anwender im C-Code-Editor dieses Element nicht erzeugt, aber trotzdem auf `dT` zugreift, gibt es keine Fehlermeldung, weil `dT` in den ERCOS^{EK} Dateien deklariert ist. Da aber `dT` in ERCOS^{EK} und `dT` in ASCET verschiedene Einheiten haben (System-Ticks bzw. Sekunden), stimmen die Berechnungen nicht. Der Anwender sollte also unbedingt darauf achten, mit der Schaltfläche **dT** das entsprechende Element zu erzeugen.

4 Rapid-Prototyping-Experimente

Dieses Kapitel beschreibt die verschiedenen Möglichkeiten, ein Rapid-Prototyping-Experiment durchzuführen.

4.1 Experimentieren mit ASCET

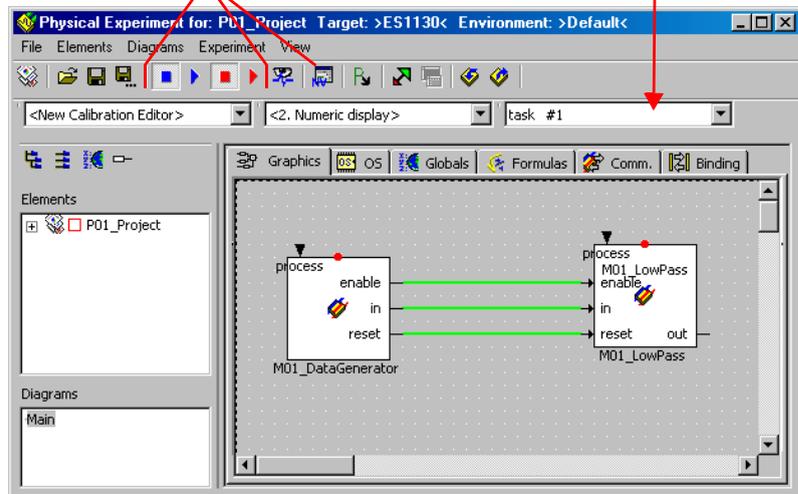
Wenn Sie das Rapid Prototyping-Experiment in ASCET durchführen wollen, können Sie im Projekteditor zwischen Online- und Offline-Experiment wählen. Details finden Sie im ASCET-Benutzerhandbuch, Kapitel „Experimentieren mit Projekten“; die Experimentierumgebung von ASCET ist im ASCET-Benutzerhandbuch im Kapitel „Die Experimentierumgebung“ ausführlich beschrieben.

An dieser Stelle werden nur die Besonderheiten des Online-Experiments beschrieben.

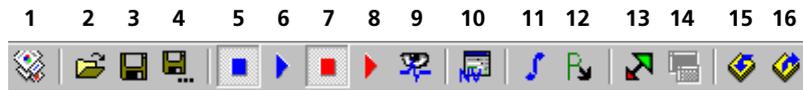
4.1.1 Die Benutzeroberfläche

Die Benutzeroberfläche des Online-Experiments ist der des Offline-Experiments sehr ähnlich. Die Schaltflächen zur Steuerung des Experiments und des NVRAM-Cockpits (nur ES1135), das Kombikästchen „Task“ und die Funktionen im Menü **Experiment** sind jedoch anders als im Offline-Experiment.

Steuerschaltflächen (Experiment, NVRAM-Cockpit) Kombikästchen „Task“



Schaltflächen



1. Exit to Component (beendet das Experiment und ruft den Projekteditor auf)
2. Load Environment (lädt eine Experimentierumgebung, d.h. vordefinierte Mess- und Verstellfenster mit zugewiesenen Größen)
3. Save Environment (speichert die aktuelle Experimentierumgebung)
4. Save Environment As (speichert die aktuelle Experimentierumgebung unter einem beliebigen Namen)
5. Stop ERCOS (stoppt das Betriebssystem und damit das Experiment)
6. Start ERCOS (startet das Betriebssystem und damit das Experiment)
7. Stop Measurement (stoppt die Messung, d.h. die Datenanzeige)
8. Start Measurement (startet die Messung, d.h. die Datenanzeige)
9. Open Data Logger
10. Open NVRAM Cockpit (öffnet das NVRAM-Cockpit, s. Seite 79)
Diese Schaltfläche ist *nur* vorhanden, wenn Sie als Target die ES1135 ausgewählt haben.
11. Open CT Solver (öffnet ein Fenster, in dem Sie die Integrationsmethode konfigurieren können)
Diese Schaltfläche ist *nur* vorhanden, wenn Sie mit einem CT-Block oder einem hybriden Projekt experimentieren.
12. Update Dependent Parameters (aktualisiert die Werte abhängiger Parameter)
13. Expand / Collapse Window (blendet die Komponentenanzeige ein/aus)
14. Always on top (hält das Experimentierfenster immer im Vordergrund)
15. Navigate down to child component (zeigt die ausgewählte aufgenommene Komponente)
16. Navigate up to parent component (zeigt die übergeordnete Komponente)

Menü **Experiment**

- *Data Logger*
Öffnet den Data-Logger.

- *NVRAM Cockpit*
(Nur verfügbar, wenn in den Codegenerierungsoptionen das Target ES1135 gewählt wurde.)
Öffnet das NVRAM-Cockpit.
- *Stop ERCOS*
Beendet das Betriebssystem.
- *Start ERCOS*
Startet das Betriebssystem.
- *Stop Measurement*
Beendet die Messung.
- *Start Measurement*
Startet die Messung.
- *Open Target Debugger*
Öffnet das Debugger-Fenster für C-Code-Komponenten.
- *Update Calibration Windows*
Aktualisiert die Inhalte der Verstellfenster.
- *Close Calibration Windows*
Schließt alle geöffneten Verstellfenster.
- *Close Measure Windows*
Schließt alle geöffneten Messfenster.

Die übrigen Elemente der Benutzeroberfläche entsprechen denen des Offline-Experiments; sie sind im ASCET-Benutzerhandbuch, Kapitel „Die Experimentierumgebung“, beschrieben.

4.1.2 Durchführen von Online-Experimenten

Sie starten die Online-Experimentierumgebung für ein Projekt aus dem Projekteditor.

Online-Experiment starten:

- Öffnen Sie das gewünschte Projekt oder die Komponente.
- Wenn Sie mit einer Komponente experimentieren wollen, öffnen Sie das zugehörige Standardprojekt.

- Wählen Sie in den Codegenerierungsoptionen des Projekts oder Standardprojekts das Target ES1130 oder ES1135.

Im Kombikästchen „Experiment Target“ sind die Einträge *Offline (RP)* und *Online (RP)* verfügbar.

Die Schaltflächen **Open Experiment for selected Experiment Target** und **Reconnect to Experiment of selected Experiment Target** sind nun verfügbar.



- Wählen Sie aus dem Kombikästchen „Experiment Target“ den Eintrag *Online (RP)*.

Der Eintrag *Offline (RP)* ist für Offline-Experimente auf dem Target gedacht.

- Wählen Sie **Component** → **Open Experiment**

oder

- klicken Sie auf die Schaltfläche **Open Experiment for selected Experiment Target**.



Hardware auswählen (mit ETAS Network Manager):

Hinweis

Wenn Sie ohne den ETAS Network Manager arbeiten, überspringen Sie diese Anleitung und lesen Sie im Abschnitt „Verhalten im Fehlerfall:“ auf Seite 39 weiter.

Wenn Sie in den Hardware-Optionen (s. Kapitel 2.1) die Option **Use ETAS Network Manager (enables 'Select Hardware')** aktiviert haben, öffnet sich beim Start des Online-Experiments unter bestimmten Umständen das Hardware-Auswahlfenster (s. Kapitel 2.2.1).

- Wählen Sie im Feld „Select simulation board of type <type>“ die Hardware aus, mit der Sie arbeiten wollen.

Die Schaltfläche **OK** wird aktiviert.

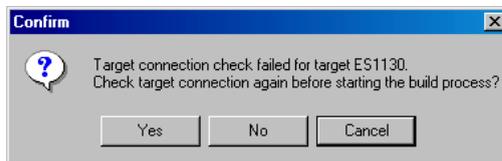
- Nehmen Sie ggf. weitere Einstellungen vor.

- Schließen Sie das Fenster mit **OK**.

Das System prüft, ob die gewählte Hardware vorhanden ist und mit dem Target übereinstimmt, das Sie in den Codegenerierungsoptionen des Projekts eingegeben haben. Ist das der Fall, wird die Experimentierumgebung geöffnet.

Verhalten im Fehlerfall:

Wird keine Übereinstimmung zwischen gewählter und vorhandener Hardware festgestellt, erscheint folgende Fehlermeldung.



- Klicken Sie auf **Yes**, um erneut nach einer Verbindung zum Target zu suchen.

Oder

- Klicken Sie auf **No**, um den Build-Prozess ohne Verbindung zur Hardware zu starten.

Wenn Sie mit dem ETAS Network Manager arbeiten, wird nach dem Build-Prozess das Hardware-Auswahlfenster erneut geöffnet.

Wenn Sie ohne den ETAS Network Manager arbeiten, werden Sie nach den Build-Prozess gefragt, ob Sie erneut nach einer geeigneten hardware suchen wollen oder nicht.

Oder

- Klicken Sie auf **Cancel**, um das Experiment abubrechen.

Öffnen der Experimentierumgebung für das Online-Experiment:

Die Standard-Experimentierumgebung für die Komponente wird unmittelbar nach dem Start des Experiments (s. Seite 37) geöffnet, wenn Sie ohne ETAS Network Manager arbeiten, bzw. nach erfolgreicher Hardware-Auswahl (s. Seite 38), wenn Sie mit dem ETAS Network Manager arbeiten.

- Wurden mehrere Umgebungen gespeichert, wählen Sie, welche geöffnet wird. Details dazu siehe Abschnitt „Laden und Speichern von Umgebungen“ im ASCET-Benutzerhandbuch.

Wenn Ihr Projekt mehrere Tasks enthält, werden Sie unter Umständen aufgefordert, für jeden Messwert eine Erfassungstask auszuwählen.



- Wählen Sie im Fenster „Selection Required“ eine Task und klicken Sie **OK**.

Später können Sie die Erfassungstask im Kombikästchen „Task“ auswählen.

Das Einrichten eines Online-Experiments umfasst nur das Einrichten der Mess- und Verstellfenster. Die Mess- und Verstellfenster im Online-Experiment sind dieselben wie die im Offline-Experiment. Ihre Verwendung sind im ASCET-Benutzerhandbuch, Kapitel „Die Experimentierumgebung“, beschrieben.

Nach dem Einrichten des Experiments können Sie dieses starten. Während das Online-Experiment läuft, können Sie die Anzeigeeoptionen an allen Mess- und Verstellfenstern verändern, Mess-/Verstellfenster öffnen und schließen und Datenwerte mit dem Verstellsystem verändern.

Experiment und Messung starten:

- Öffnen Sie die Experimentierumgebung für das Projekt, mit dem Sie experimentieren möchten.
- Wählen Sie **Experiment** → **Start ERCOS**
oder



- klicken Sie auf die Schaltfläche **Start ERCOS**. Das Betriebssystem und damit das Experiment wird gestartet. Messdaten werden noch nicht angezeigt.

- Wählen Sie **Experiment** → **Start Measurement**

oder

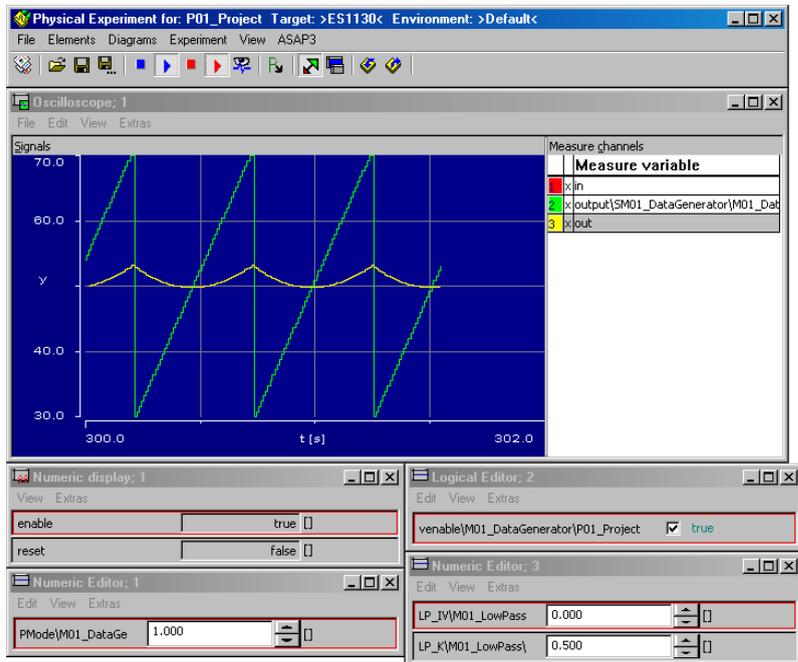
- klicken Sie auf die Schaltfläche **Start Measurement**.



Hinweis

Die Messung kann das Echtzeitverhalten des Betriebssystems beeinflussen.

Die Messung wird gestartet, und es werden alle im Mess-System eingerichteten Werte in den entsprechenden Fenstern angezeigt.



Messung beenden:

- Wählen Sie **Experiment** → **Stop Measurement**

oder



- Klicken Sie auf die Schaltfläche **Stop Measurement**.

Die Messung wird beendet, das Experiment läuft jedoch weiter. Beim Neustarten der Messung wird die Zeitachse auf den dann aktuellen Wert gesetzt.

Alle Einstellungen bleiben aktiv. Die Messdaten bleiben im Oszilloskopfenster erhalten, Sie können die Daten analysieren.

Experiment beenden:

- Wählen Sie **Experiment** → **Stop ERCOS**

oder



- Klicken Sie auf die Schaltfläche **Stop ERCOS**.

Eine laufende Messung wird beendet. Das Betriebssystem und damit das Experiment wird angehalten und geht in einen inaktiven Modus über. Der inaktive Modus des Betriebssystems kann eine Task mit Trigger-modus `init` enthalten, die beim Anhalten des Betriebssystems ausgeführt wird. Diese kann beispielsweise zum Rücksetzen externer Hardware genutzt werden. Beim erneuten Start des Betriebssystems durchläuft es wiederum den Startmodus und die entsprechende `init`-Task. Es gibt keine Pausenfunktion des Betriebssystems.



- Klicken Sie auf die Schaltfläche **Exit to Component**, um die Experimentierumgebung zu verlassen und zum Projekteditor zu gelangen.
- Wählen Sie **File** → **Exit**, um die Experimentierumgebung zu verlassen und das Projekt zu schließen.

Wie beim Offline-Experiment können Sie aus der Experimentierumgebung heraus jederzeit die Implementierung des Projekts ansehen, mit dem Sie experimentieren. Dabei spielt es keine Rolle, ob das Experiment läuft oder angehalten wurde.

Ebenfalls wie beim Offline-Experiment bieten in C-Code spezifizierte Komponenten zusätzliche Möglichkeiten zur Anzeige von Debugger-Informationen oder Fehlermitteilungen während des Experimentierens. Sie können Debugger- oder Fehlermitteilungen in Ihrem C-Code einbetten. Debugger-Informationen werden im Debugger-Fenster angezeigt, das sich während des Experimentierens öffnen lässt. Fehlermitteilungen werden im Monitorfenster von ASCET angezeigt. Der Debugger im Online-Experiment funktioniert wie der Debugger im Offline-Experiment (siehe Abschnitt „Durchführen von Experimenten“ im ASCET-Benutzerhandbuch).

4.1.3 Standalone-Modus

Wenn Sie über ein geeignetes experimentelles Target verfügen, können Sie ASCET-Experimente im Standalone-Modus ohne Experimentierumgebung durchführen. Ein solches experimentelles Target ist mit Flash-Speicher auszurüsten und muss darauf hochgefahren werden können.

Experiment in den Flash-Speicher laden:



- Wählen Sie aus dem Kombikästchen „Experiment Target“ den Eintrag **Online (RP)**.
- Wählen Sie **Component** → **Flash Target**.

Wenn Sie in den Hardware-Optionen (s. Kapitel 2.1) die Option **Use ETAS Network Manager (enables 'Select Hardware')** aktiviert haben, öffnet sich an dieser Stelle unter bestimmten Umständen das Hardware-Auswahlfenster (s. Kapitel 2.2.1).

Was Sie in diesem Fenster tun müssen, ist im Abschnitt „Hardware auswählen (mit ETAS Network Manager):“ auf Seite 38 beschrieben.

Der durch ASCET generierte Code wird anstatt in den RAM in den Flash-Speicher des experimentellen Targets geschrieben. Ein Startprogramm zum Hochfahren aus dem Flash-Speicher ist im Code integriert. Die Hardware des Targets führt dann das ASCET-Modell nach jedem Reset aus.

Experimentieren im Standalone-Modus:

- Wählen Sie **Component** → **Reconnect To Experiment**

oder

- klicken Sie auf **Reconnect to Experiment of selected Experiment Target** zum Umschalten auf ein laufendes Online-Experiment.



Reconnect to Experiment of selected Experiment Target

Wenn Sie in den Hardware-Optionen (s. Kapitel 2.1) die Option **Use ETAS Network Manager (enables 'Select Hardware')** aktiviert haben, öffnet sich an dieser Stelle unter bestimmten Umständen das Hardware-Auswahlfenster (s. Kapitel 2.2.1).

Was Sie in diesem Fenster tun müssen, ist im Abschnitt „Hardware auswählen (mit ETAS Network Manager):“ auf Seite 38 beschrieben.

Die Online-Experimentierumgebung wird gestartet, als ob das Experiment ganz von Anfang an gestartet würde.

Sie können auch auf Online-Experimente umschalten, die nicht im Standalone-Modus laufen und die Sie früher abgeschaltet haben. Zum Abschalten eines laufenden Experiments verlassen Sie einfach die Experimentierumgebung, ohne vorher das auf dem experimentellen Target laufende Online-Experiment zu unterbrechen.

4.2 Experimentieren mit INCA

Sie können mit Ihrem Projekt nicht nur in ASCET, sondern auch in INCA (ab Version 4.0.4), zusammen mit dem Add-On INCA-EIP experimentieren. Hierfür gibt es im Projekteditor eine Funktionalität, die einen bequemen Transfer des Experimentalprojekts ermöglicht.

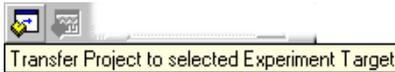
Transfer aufrufen:

- Öffnen Sie das Projekt, mit dem Sie experimentieren wollen.
- Wählen Sie im Projekteditor in den Codegenerierungsoptionen das Target **ES1130** oder **ES1135** und den Compiler **GNU-C (PowerPC)**.



- Wählen Sie aus dem Kombikästchen „Experiment Target“ den Eintrag **INCA**.

Die Schaltflächen **Open Experiment for selected Experiment Target** und **Reconnect to Experiment of selected Experiment Target** sind nun verfügbar.

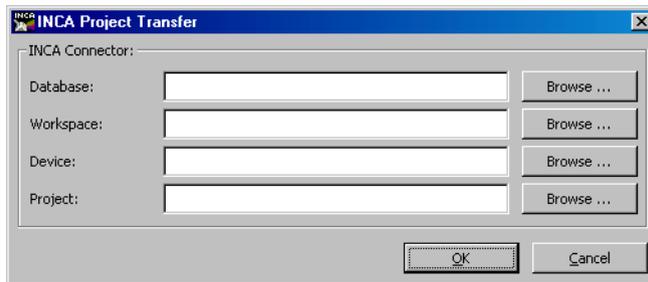


- Klicken Sie auf die Schaltfläche **Transfer Project to selected Experiment Target**

oder

- wählen Sie **Component** → **Transfer Project**.

Das Fenster „INCA Project Transfer“ öffnet sich.



In diesem Fenster stellen Sie die zu verwendende INCA-Datenbank sowie die Arbeitsumgebung und das Projekt innerhalb der INCA-Datenbank ein.

Wenn Sie auf eine der Schaltflächen **Browse** oder auf **OK** klicken, wird INCA gestartet, falls es nicht bereits ausgeführt wird. Haben Sie mehr als eine INCA-Version installiert, wird diejenige Version gestartet, die zuletzt installiert wurde – auch wenn dieses nicht die Version mit der höchsten Versionsnummer ist.

Hinweis

Wenn Sie eine zu alte (d.h. V3.x oder älter) oder gar keine Version von INCA installiert haben, wird eine Fehlermeldung angezeigt, der Transfer bricht ab.

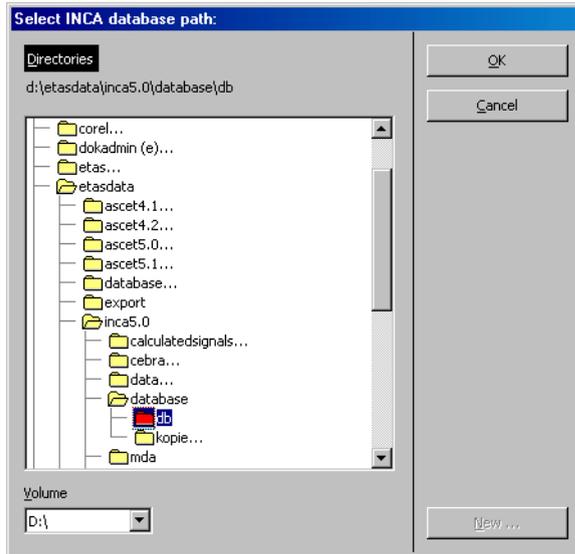
INCA-Datenbankpfad einstellen:

- Geben Sie im Fenster „INCA Project Transfer“ im Feld „INCA Database“ einen existierenden Datenbankpfad ein

oder

- klicken Sie auf die Schaltfläche **Browse**, um nach einer Datenbank zu suchen.

Das Fenster „Select INCA database path“ öffnet sich. Die Datenbanken sind durch rote Ordnersymbole gekennzeichnet.



- Wählen Sie einen INCA-Datenbankpfad aus und klicken Sie **OK**.

Hinweis

*In diesem Fenster werden ASCET-Datenbanken genauso markiert wie INCA-Datenbanken. Achten Sie unbedingt darauf, auch wirklich eine **INCA-Datenbank** auszuwählen.*

Oder

- klicken Sie auf die Schaltfläche **New**, um ein neues Verzeichnis zu erstellen, in dem anschließend eine INCA-Datenbank für den Projekttransfer angelegt wird. Das neue, leere Verzeichnis wird gelb dargestellt.

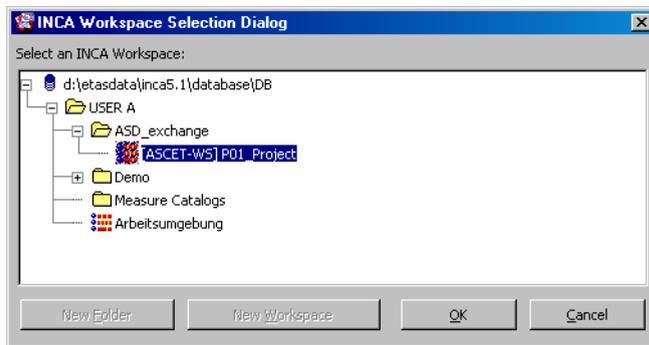
INCA-Arbeitsumgebung auswählen:

- Geben Sie im Fenster „INCA Project Transfer“ im Feld „INCA Workspace“ Pfad und Name einer existierenden Arbeitsumgebung ein

oder

- klicken Sie auf die Schaltfläche **Browse**, um nach einer Arbeitsumgebung zu suchen.

Das Fenster „INCA Workspace Selection Dialog“ öffnet sich. Es zeigt die Arbeitsumgebungen in der ausgewählten INCA-Datenbank an.



- Wählen Sie eine vorhandene Arbeitsumgebung aus

oder

- legen Sie eine neue Arbeitsumgebung in einem neuen oder existierenden Ordner an. Das Vorgehen ist auf Seite 47 beschrieben.
- Wenn Sie eine Arbeitsumgebung ausgewählt haben, klicken Sie auf **OK**.

Ordner/Arbeitsumgebung im Fenster „Workspace Selection Dialog“ anlegen:

1. Ordner anlegen

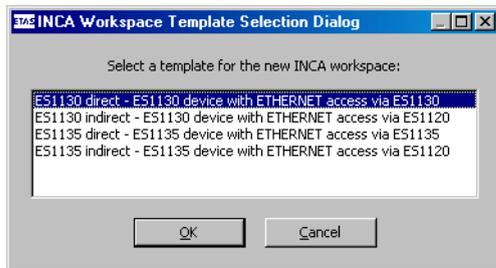
- Markieren Sie im Fenster „INCA Workspace Selection Dialog“ den Datenbanknamen oder einen Ordner.

Die Schaltfläche **New Folder** wird aktiviert.

- Klicken Sie auf **New Folder**.
- Geben Sie im Fenster „Create New INCA Folder“ einen Namen ein und klicken Sie **OK**.
Der neue Ordner wird angelegt.

2. Arbeitsumgebung anlegen

- Markieren Sie einen Ordner.
Die Schaltfläche **New Workspace** wird aktiviert.
- Klicken Sie auf **New Workspace**.
Das Fenster „INCA Workspace Template Selection Dialog“ öffnet sich.



- Wählen Sie ein Template für die Arbeitsumgebung und klicken Sie auf **OK**.
Das Eingabefenster „Create New INCA Workspace“ öffnet sich.



Der Name [ASCET-WS] <ascet project name> ist vorgegeben.

- Geben Sie einen Namen für die Arbeitsumgebung ein und klicken Sie **OK**.
Die neue Arbeitsumgebung wird im ausgewählten Ordner angelegt und ausgewählt.

INCA-Gerät auswählen:

- Geben Sie im Fenster „INCA Project Transfer“ im Feld „INCA Device“ einen Gerätenamen ein

oder

- klicken Sie auf die Schaltfläche **Browse**, um ein Gerät auszuwählen.

Das Fenster „INCA Device Selection Dialog“ öffnet sich. Es enthält alle geeigneten Geräte.



- Wählen Sie ein Gerät und klicken Sie auf **OK**.

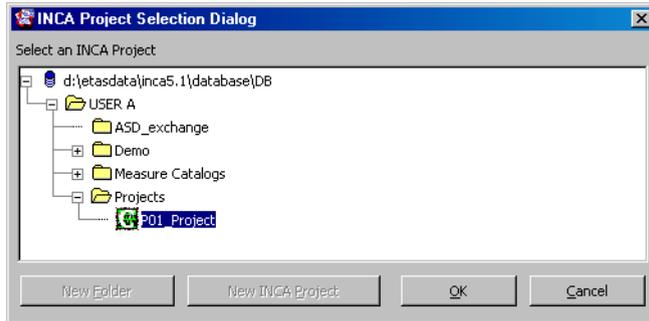
INCA-Projekt auswählen:

- Geben Sie im Fenster „INCA Project Transfer“ im Feld „INCA Project“ Pfad und Name eines existierenden Projekts ein

oder

- klicken Sie auf die Schaltfläche **Browse**, um nach einem Projekt zu suchen.

Das Fenster „INCA Project Selection Dialog“ öffnet sich. Es zeigt die Projekte in der ausgewählten INCA Datenbank an.



- Wählen Sie ein vorhandenes Projekt aus *oder*
- legen Sie ein neues Projekt in einem neuen oder existierenden Ordner an. Das Vorgehen ist auf Seite 50 beschrieben.
- Wenn Sie ein Projekt ausgewählt haben, klicken Sie auf **OK**.

Hinweis

Das ausgewählte Projekt wird beim Start des Transfers ohne weitere Warnung durch das zu transferierende Projekt ersetzt.

Ordner/Projekt im Fenster „INCA Project Selection Dialog“ anlegen:

1. Ordner anlegen

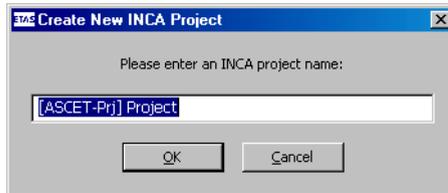
- Um einen neuem Ordner anzulegen, gehen Sie vor wie unter „Ordner anlegen“ auf Seite 47 beschrieben.

2. Projekt anlegen

- Markieren Sie einen Ordner.
Die Schaltfläche **New INCA Project** wird aktiviert.

- Klicken Sie auf **New INCA Project**.

Das Eingabefenster „Create New INCA Project“ öffnet sich. Der Name [ASCET-Prj] <ascet project name> ist vorgegeben.



- Geben Sie einen Namen für das Projekt ein und klicken Sie auf **OK**.

Das neue Projekt wird im ausgewählten Ordner angelegt und ausgewählt.

Transfer starten:

- Wenn Sie im Fenster „INCA Project Transfer“ alle Einstellungen vorgenommen haben, klicken Sie auf **OK**.

Der Transfer des ASCET-Projekts nach INCA wird durchgeführt. Dabei wird zunächst der ASAM-MCD-2MC-Code generiert und gegebenenfalls das Projekt neu gebaut (d.h. Code generiert, kompiliert und gelinkt).

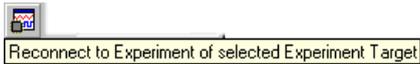
Wenn Sie im Feld „INCA Project“ ein existierendes Projekt ausgewählt haben, wird dieses überschrieben.

Nach Beendigung des Transfers können Sie das Experiment in INCA durchführen. Einzelheiten dazu entnehmen Sie der INCA- und INCA-EIP-Dokumentation.

Zusätzlich steht Ihnen über die Back-Animation in ASCET eine spezielle Experimentierumgebung zur Verfügung, in der Sie wie gewohnt Werte verstellen können. Das Mess-System dieser Experimentierumgebung funktioniert ebenfalls wie gehabt, hat aber verglichen mit Offline- und Online-Experimenten in ASCET einen reduzierten Umfang: Oszilloskop, Recorder und Data-Logger stehen nicht zur Verfügung. Diese brauchen eine synchrone Messung, die beim Experimentieren mit INCA für die Back-Animation nicht gegeben ist. Verwenden Sie stattdessen die entsprechenden Instrumente von INCA.

Back-Animation verwenden:

- Starten Sie das INCA-Experiment mit Ihrem Projekt.
- Vergewissern Sie sich, dass im Projekteditor von ASCET im Kombikästchen „Experiment Target“ der Eintrag **INCA** gewählt ist.
- Klicken Sie auf die Schaltfläche **Reconnect to Experiment of Selected Experiment Target**



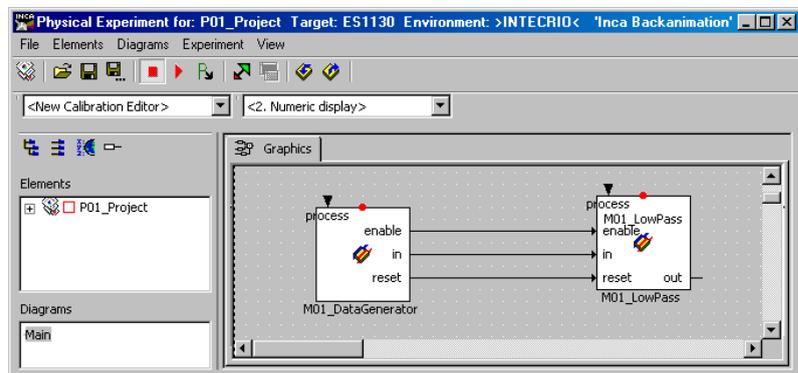
oder

- wählen Sie **Component** → **Reconnect to Experiment**.

Wenn Sie in den Hardware-Optionen (s. Kapitel 2.1) die Option **Use ETAS Network Manager (enables 'Select Hardware')** aktiviert haben, öffnet sich an dieser Stelle unter bestimmten Umständen das Hardware-Auswahlfenster (s. Kapitel 2.2.1).

Was Sie diesem Fenster tun müssen, ist im Abschnitt „Hardware auswählen (mit ETAS Network Manager):“ auf Seite 38 beschrieben.

Die Verbindung zu dem bereits laufenden INCA-Experiment wird hergestellt. Das Fenster „Physical Experiment ...“ öffnet sich. Der Zusatz „INCA Backanimation“ kennzeichnet die spezielle Experimentierumgebung.



Anders als beim Online- oder Offline-Experiment enthält dieses Fenster nur das Register „Graphics“.

- Erzeugen Sie die erforderlichen Messfenster (siehe ASCET-Benutzerhandbuch, Abschnitt „Das Mess-System“) und richten Sie diese ein.
- Erzeugen Sie die erforderlichen Verstellfenster (siehe ASCET-Benutzerhandbuch, Abschnitt „Das Verstellsystem“) und richten Sie diese ein.
- Wählen Sie **Experiment** → **Start Measurement**

oder



- klicken Sie auf die Schaltfläche **Start Measurement**, um die Messung zu starten.

Die Anzeigen der Mess- und Verstellfenster werden zyklisch aktualisiert.

Sie können Umgebungen laden, speichern und exportieren wie im ASCET-Benutzerhandbuch, Abschnitt „Laden und Speichern von Umgebungen“ beschrieben. Wenn Sie eine Umgebung laden, die nicht verfügbare Elemente (z.B. ein Oszilloskop) enthält, werden diese ignoriert.

Die Monitorfunktion (s. ASCET-Benutzerhandbuch, Abschnitt „Monitor“) für die Überwachung numerischer und logischer Größen steht zur Verfügung. Sie können die Funktion für einzelne oder alle Größen einer Komponente aktivieren. Die Einstellung der Monitorfunktion wird in der Umgebung gespeichert.

Sie können zwischen den Komponenten Ihres Projekts navigieren (siehe ASCET-Benutzerhandbuch, Abschnitt „Navigieren in Blockdiagrammen“).

Wenn Ihr Projekt Zustandsautomaten enthält, können Sie die Animationsfunktion für Zustandsautomaten verwenden (siehe ASCET-Benutzerhandbuch, Abschnitt „Experimentieren mit Zustandsautomaten“).

Sie können Daten aus dem Experiment in das ASCET-Modell oder auf die Festplatte schreiben; ebenso können Sie Daten von der Festplatte einlesen. Wie das geschieht, ist im ASCET-Benutzerhandbuch, Abschnitt „Manipulieren von Daten“, beschrieben.

Back-Animation beenden:

- Wählen Sie **File** → **Exit**

oder



- klicken Sie auf die Schaltfläche **Exit to Component**.

Die Back-Animation wird beendet und die Experimentierumgebung geschlossen. Das INCA-Experiment läuft jedoch weiter.

4.3 Experimentieren mit INTECRIO

Wenn Sie neben ASCET-RP auch INTECRIO installiert haben, können Sie mit Ihrem Rapid Prototyping-Projekt auch in INTECRIO experimentieren. Hierfür gibt es im Projekteditor eine Funktionalität, die einen bequemen Transfer des Experiments ermöglicht.

Dieses Kapitel enthält eine allgemeine Anleitung zum Experimentieren mit INTECRIO. Eine konkrete Beispielaufgabe finden Sie in Kapitel 11.1 „Tutorial – Experimentieren mit INTECRIO“.

Sie erstellen zunächst wie gewohnt das ASCET-Projekt. Dabei stehen Ihnen alle Möglichkeiten von ASCET zur Verfügung. Beachten Sie jedoch folgende Punkte:

- Standardmäßig bilden Messages, die in ASCET nur gelesen werden (d.h. Receive-Messages ohne zugehörige Send-Message) in INTECRIO die Signalsenken. Messages, die nur geschrieben werden (d.h. Send-Messages ohne zugehörige Receive-Message), bilden in INTECRIO die Signalquellen. Messages, die in ASCET sowohl gelesen als auch geschrieben werden, werden bei der Integration nicht berücksichtigt.
Bei Bedarf können Sie die letztgenannten Messages als Signalquellen in INTECRIO erscheinen lassen, siehe Seite 57.
- Wenn Ihr Projekt nicht aufgelöste Messages (importierte Messages ohne zugehörigen Export) enthält, wird bei der Codegenerierung für INTECRIO eine Fehlermeldung angezeigt.
Sie können die Messages automatisch auflösen oder die Codegenerierung abbrechen und die Messages manuell auflösen.
- Die Verwendung von globalen Variablen und Parametern ist möglich, wird aber ausdrücklich *nicht* empfohlen.
- In den Codegenerierungsoptionen des Projekts müssen Sie das Target ES1130 oder ES1135 wählen. Nur mit einem dieser beiden Targets ist im Kombikästchen „Experiment Target“ der Eintrag INTECRIO verfügbar.

Nachdem Sie das Projekt komplett spezifiziert haben, rufen Sie als ersten Schritt der Codegenerierung den Transfer des Projekts nach INTECRIO auf.

Transfer aufrufen:



- Öffnen Sie das Projekt, mit dem Sie experimentieren wollen.
- Wählen Sie im Projekteditor in den Codegenerierungsoptionen das Target ES1130 oder ES1135 und den Compiler GNU-C (PowerPC).
- Wählen Sie aus dem Kombikästchen „Experiment Target“ den Eintrag INTECRIO.

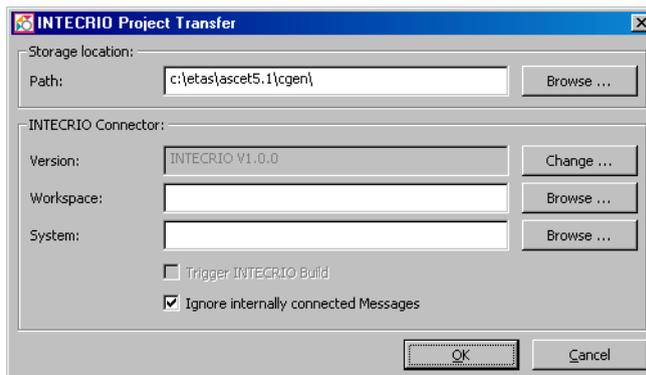
Die Schaltflächen **Transfer Project to selected Experiment Target** und **Reconnect to Experiment of selected Experiment Target** sind nun verfügbar.

- Klicken Sie auf die Schaltfläche **Transfer Project to selected Experiment Target**

oder

- wählen Sie **Component** → **Transfer Project**.

Das Fenster „INTECRIO Project Transfer“ öffnet sich. Im Feld „Path“ ist das Codegenerierungsverzeichnis aus dem ASCET-Optionsfenster (Register „Code Generation“) eingetragen.



Als zweiten Schritt stellen Sie im Fenster „INTECRIO Project Transfer“ den Pfad ein, unter dem die generierten Dateien abgelegt werden. Danach haben Sie vier Möglichkeiten:

- Wenn Sie lediglich den für INTECRIO erforderlichen *Code generieren* wollen, bleiben die übrigen Felder leer.

Das kann z.B. dann der Fall sein, wenn der generierte Code für die Weitergabe bestimmt ist.

Hinweis

Codegenerierung für INTECRIO ist auch dann möglich, wenn INTECRIO auf Ihrem Rechner nicht installiert ist.

- Wenn Sie *Code generieren und in INTECRIO importieren* wollen, wählen Sie zusätzlich die INTECRIO-Version und die INTECRIO-Arbeitsumgebung aus. Das Feld „Systems“ bleibt leer.

Standardmäßig ist im Feld „Version“ die *zuletzt* installierte Version von INTECRIO eingetragen. Wenn nur eine INTECRIO-Version installiert ist, ist diese eingetragen; die Schaltfläche **Change** ist dann deaktiviert.

Wenn die angegebene Arbeitsumgebung nicht existiert, wird sie automatisch angelegt.

- Wenn Sie *Code generieren, in INTECRIO importieren und integrieren* (d.h. in ein INTECRIO-Systemprojekt einfügen) wollen, geben Sie auch noch das INTECRIO-Systemprojekt an, mit dem Sie arbeiten wollen.

In diesem Fall müssen sowohl die Arbeitsumgebung als auch das Systemprojekt bereits existieren.

- Wenn Sie *Code generieren, in INTECRIO importieren und integrieren und den Build-Prozess in INTECRIO starten* wollen, füllen Sie alle Felder aus und aktivieren Sie die Option **Trigger INTECRIO Build**.

Damit der Build-Prozess durchlaufen kann und ein brauchbarer Prototyp generiert wird, müssen in INTECRIO ein Hardwaresystem und die Betriebssystemkonfiguration komplett spezifiziert sein.

Wenn Sie Messages konvertieren wollen, die innerhalb des ASCET-Projekts gelesen und geschrieben werden, deaktivieren Sie die Option **Ignore internally connects messages**. Diese Option kann mit allen vier Möglichkeiten verwendet werden. Tab. 4-1 fasst die Konvertierung von Messages in Schnittstellen für die aktivierte und deaktivierte Option zusammen.

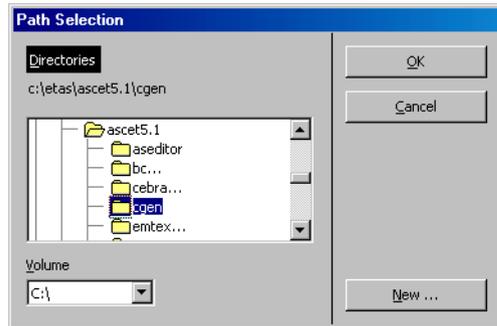
Message-Zugriff im			INTECRIO-Schnittstellen	
Projekt	Modul A	Modul B	Option aktiviert	Option deaktiviert
S/R			—	—
S/R	S		Signalquelle	Signalquelle
S/R	R		Signalsenke	Signalsenke
S/R	S/R		Signalquelle	Signalquelle
	S		Signalquelle	Signalquelle
	S	S	Signalquelle	Signalquelle
	S	R	—	Signalquelle
	S	S/R	—	Signalquelle
	R		Signalsenke	Signalsenke
	R	R	Signalsenke	Signalsenke
	R	S/R	—	Signalquelle

Tab. 4-1 Zusammenfassung der Message-Konvertierung. *S* bezeichnet Messages, die von der jeweiligen Komponente *gesendet* werden, *R* bezeichnet Messages, die von der jeweiligen Komponente *empfangen* werden

Pfad für die generierten Dateien einstellen:

- Klicken Sie auf die Schaltfläche **Browse** neben dem Feld „Path“.

Das Fenster „Path Selection“ öffnet sich.



- Wählen Sie bei Bedarf im Kombikästchen „Volume“ ein Laufwerk.
- Wählen Sie in der Liste „Directories“ ein vorhandenes Verzeichnis

oder

- erzeugen Sie mit der Schaltfläche **New** ein neues Verzeichnis.
- Klicken Sie auf **OK**.

Das Verzeichnis wird im Feld „Path“ angezeigt. Ihre Auswahl wird mit dem Projekt gespeichert und ist beim nächsten Transfer voreingestellt.

INTECRIO-Version auswählen:

Wenn auf Ihrem Rechner nur eine INTECRIO-Version installiert ist, ist diese Version automatisch ausgewählt. Sie brauchen dann nicht weiter tun.

- Klicken Sie auf die Schaltfläche **Change** neben dem Feld „Version“.

Ein Auswahlfenster öffnet sich. Es enthält alle INTECRIO-Versionen, die auf Ihrem Rechner installiert sind.

- Wählen Sie eine INTECRIO-Version.

- Klicken Sie auf **OK**.

Die Version wird im Feld „Versions“ angezeigt. Ihre Auswahl wird mit dem Projekt gespeichert.

INTECRIO-Arbeitsumgebung auswählen:

- Geben Sie im Feld „Workspace“ Pfad und Namen der INTECRIO-Arbeitsumgebung ein, die Sie verwenden wollen.

Oder

- Klicken Sie auf die Schaltfläche **Browse** neben dem Feld „Workspace“.

Das Windows-Dateiauswahlfenster öffnet sich.

- Wählen Sie das Verzeichnis, in dem sich die Arbeitsumgebung befindet.
- Wählen Sie die Arbeitsumgebung (*.iow).
- Klicken Sie auf **Öffnen**.

Die Arbeitsumgebung wird im Feld „Workspace“ angezeigt.

INTECRIO-Systemprojekt auswählen:

Für die erfolgreiche Integration des ASCET-Projekts in ein INTECRIO-Systemprojekt muss eine Arbeitsumgebung ausgewählt sein und INTECRIO laufen.

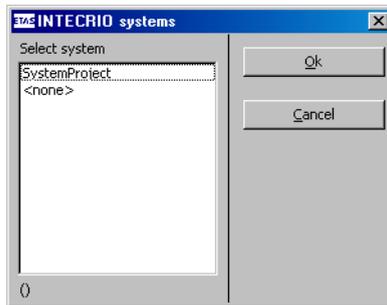
- Geben Sie im Feld „System“ den Namen des INTECRIO-Systemprojekts ein.

Oder

- Klicken Sie auf die Schaltfläche **Browse** neben dem Feld „System“.

Wenn INTECRIO noch nicht läuft, wird es gestartet.

Das Fenster „INTECRIO systems“ öffnet sich. Es zeigt im Feld „Select system“ alle in der Arbeitsumgebung enthaltenen Systemprojekte.



- Wählen Sie das Systemprojekt, in das Sie das ASCET-Projekt einfügen wollen.
- Klicken Sie auf **OK**.

Das Systemprojekt wird im Feld „System“ angezeigt.

Build-Prozess in INTECRIO auswählen:

Wenn das INTECRIO-Systemprojekt nur ein ASCET-Projekt enthält, und wenn ein Hardwaresystem und die OS-Konfiguration in INTECRIO bereits erstellt sind, kann der Build-Prozess beim Transfer automatisch gestartet werden.

- Aktivieren Sie die Option **Trigger INTECRIO Build**.
Diese Option ist nur sinnvoll, wenn sowohl eine INTECRIO-Arbeitsumgebung als auch ein Systemprojekt ausgewählt wurden.

Als letzten Schritt führen Sie den Transfer nach INTECRIO durch.

Transfer durchführen:

- Nachdem Sie im Fenster „INTECRIO Project Transfer“ alle Angaben gemacht haben, die Sie benötigen, klicken Sie auf **OK**.

Der Transfer des ASCET-Projekts nach INTECRIO wird gestartet.

1. Problem: nicht aufgelöste Messages

Wenn Ihr Projekt nicht aufgelöste Messages enthält, erscheint die folgende Meldung.



- Klicken Sie **OK**, um die Messages automatisch auflösen.

Funktioniert die automatische Auflösung, läuft der Transfer nach INTECRIO weiter.

Oder

- Klicken Sie **Cancel**, um die Codegenerierung abbrechen und die Messages manuell auflösen.

Sie müssen den Transfer erneut starten.

2. Problem: eine andere Arbeitsumgebung in INTECRIO

Wenn INTECRIO beim Start des Transfers läuft und eine andere Arbeitsumgebung geöffnet ist als die, die Sie ausgewählt haben, erscheint die folgende Meldung.



- Klicken Sie **OK**, um die bereits geöffnete Arbeitsumgebung zu schließen und den Transfer des ASCET-Projekts nach INTECRIO fortzusetzen.

Oder

- Klicken Sie **Cancel**, um den Transfer abzubrechen.

Bei dem Transfer werden sämtliche für die Arbeit mit INTECRIO erforderlichen Dateien generiert und in dem angegebenen Verzeichnis abgelegt.

Wenn Sie die entsprechenden Angaben gemacht haben, wird das Projekt in INTECRIO importiert und in das Systemprojekt integriert sowie der Build-Prozess gestartet.

Folgende generierte Dateien sind für die Arbeit mit INTECRIO von Bedeutung:

- `<project name>.six`

Diese Datei enthält die Beschreibung der Projektschnittstellen in der XML-basierten Sprache SCOOP-IX.
Die Schnittstellen des HWC-Moduls sind in der SCOOP-IX-Datei *nicht* enthalten, da die Hardwarekonfiguration in INTECRIO erstellt wird.
Eine SCOOP-IX-Schnittstellenbeschreibung besteht im wesentlichen aus den folgenden Informationen:

 - Name, Typ und Größe von C-Variablen
 - Name, Rückgabewert und Signatur von C-Funktionen
 - Dateiursprung der C-Elemente

Details finden Sie im Kapitel „SCOOP und SCOOP-IX“ im INTECRIO-Benutzerhandbuch.
- `<project name>.a21`

Die für die Arbeit mit INTECRIO generierte ASAM-MCD-2MC-Datei.

- `<project name>.oil`

Diese Datei enthält die in INTECRIO verwendbare Beschreibung des Betriebssystems.

Auch hier bleibt das HWC-Modul unberücksichtigt, da Hardware- und Betriebssystemkonfiguration in INTECRIO vorgenommen werden.

Hinweis

*Diese Datei wird in **keinem** Fall automatisch in INTECRIO importiert. Sie müssen das Betriebssystem entweder in INTECRIO manuell konfigurieren oder die *.oil-Datei manuell importieren.*

*Das Format dieser *.oil-Date entspricht nicht dem OSEK-Standard; es ist eine XML-basierte Beschreibung der Betriebssystemkonfiguration.*

- `*.c` und `*.h`

Die C-Code- und Header-Dateien für das Projekt und seine verschiedenen Komponenten. Welche `*.c`- und `*.h`-Dateien genau von INTECRIO verwendet werden, ist in folgendem Block der `*.six`-Datei enthalten:

```
<fileContainer complete="false">
  <pathBase path="{{codeDir}}" />
  <!-- model specific C files -->
  ... *.c- und *.h-Dateien ...
</fileContainer>
```

Darüber hinaus werden bei der Codegenerierung weitere Dateien erzeugt und unabhängig vom ausgewählten Verzeichnis in `ETAS\ASCET5.1\CGen` abgelegt. Für die Arbeit mit INTECRIO sind diese Dateien jedoch irrelevant.

Nach Beendigung des Transfers können Sie mit dem Projekt in INTECRIO experimentieren. Je nachdem, welche Angaben für den Transfer Sie gemacht haben, müssen Sie dazu verschiedene Schritte ausführen.

Experiment starten:

Die Ausführung der einzelnen Schritte ist in der INTECRIO-Dokumentation beschrieben.

- [Importieren Sie den Code manuell in INTECRIO.](#)

[Dieser Schritt entfällt, wenn Sie den Code automatisch importiert haben.](#)

- Fügen Sie das Modell in das INTECRIO-Systemprojekt ein.
Dieser Schritt entfällt, wenn Sie den Code automatisch integriert haben.
- Vervollständigen Sie das Systemprojekt.
Dazu gehören das Erstellen eines Hardware-systems, die Betriebssystemkonfiguration, die Verbindungen zwischen Hard- und Software
- Konfigurieren Sie das Betriebssystem entweder manuell oder durch Import der *.oil-Datei.
- Generieren Sie die ausführbare Datei.
- Starten Sie das Experiment.

Die Arbeit mit der Experimentierumgebung von INTECRIO ist im INTECRIO-Benutzerhandbuch beschrieben.

Zusätzlich steht Ihnen über die Back-Animation von INTECRIO in ASCET eine spezielle Experimentierumgebung zur Verfügung, in der Sie wie gewohnt Werte verstellen können. Das Mess-System dieser Experimentierumgebung funktioniert ebenfalls wie gehabt, hat aber verglichen mit Offline- und Online-Experimenten in ASCET einen reduzierten Umfang: Oszilloskop, Recorder und Data-Logger stehen nicht zur Verfügung. Diese brauchen eine synchrone Messung, die beim Experimentieren mit INTECRIO für die Back-Animation nicht gegeben ist. Verwenden Sie stattdessen die entsprechenden Instrumente von INTECRIO.

Back-Animation verwenden:

- Starten Sie das INTECRIO-Experiment mit Ihrem Projekt.
- Vergewissern Sie sich, dass im Projekteditor von ASCET im Kombikästchen „Experiment Target“ der Eintrag **INTECRIO** gewählt ist.
- Klicken Sie auf die Schaltfläche **Reconnect to Experiment of Selected Experiment Target**

oder



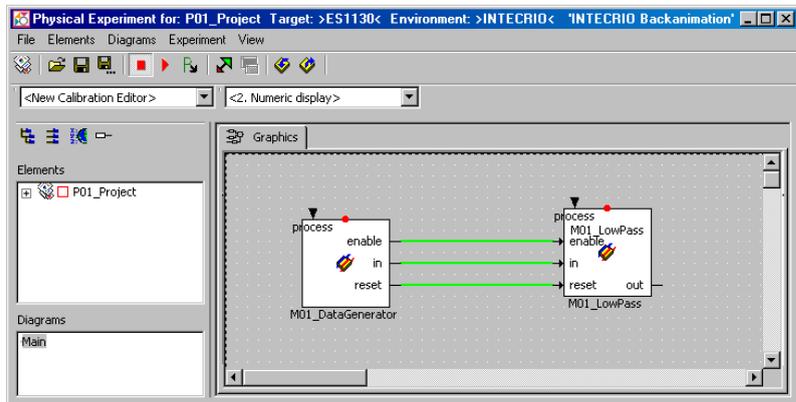
Reconnect to Experiment of selected Experiment Target

- wählen Sie **Component** → **Reconnect to Experiment**.

Wenn Sie in den Hardware-Optionen (s. Kapitel 2.1) die Option **Use ETAS Network Manager (enables 'Select Hardware')** aktiviert haben, öffnet sich an dieser Stelle unter bestimmten Umständen das Hardware-Auswahlfenster (s. Kapitel 2.2.1).

Was Sie diesem Fenster tun müssen, ist im Abschnitt „Hardware auswählen (mit ETAS Network Manager):“ auf Seite 38 beschrieben.

Die Verbindung zu dem bereits laufenden INTECRIO-Experiment wird hergestellt. Das Fenster „Physical Experiment ...“ öffnet sich. Der Zusatz „INTECRIO Backanimation“ kennzeichnet die spezielle Experimentierumgebung.



Anders als beim Online- oder Offline-Experiment enthält dieses Fenster nur das Register „Graphics“.

- Erzeugen Sie die erforderlichen Messfenster (siehe ASCET-Benutzerhandbuch, Abschnitt „Das Mess-System“) und richten Sie diese ein.

- Erzeugen Sie die erforderlichen Verstellfenster (siehe ASCET-Benutzerhandbuch, Abschnitt „Das Verstellsystem“) und richten Sie diese ein.
- Wählen Sie **Experiment** → **Start Measurement**

oder



- klicken Sie auf die Schaltfläche **Start Measurement**, um die Messung zu starten.

Die Anzeigen der Mess- und Verstellfenster werden zyklisch aktualisiert.

Sie können Umgebungen laden, speichern und exportieren wie im ASCET-Benutzerhandbuch, Abschnitt „Laden und Speichern von Umgebungen“ beschrieben. Wenn Sie eine Umgebung laden, die nicht verfügbare Elemente (z.B. ein Oszilloskop) enthält, werden diese ignoriert.

Die Monitorfunktion (s. ASCET-Benutzerhandbuch, Abschnitt „Monitor“) für die Überwachung numerischer und logischer Größen steht zur Verfügung. Sie können die Funktion für einzelne oder alle Größen einer Komponente aktivieren. Die Einstellung der Monitorfunktion wird in der Umgebung gespeichert.

Sie können zwischen den Komponenten Ihres Projekts navigieren (siehe ASCET-Benutzerhandbuch, Abschnitt „Navigieren in Blockdiagrammen“).

Wenn Ihr Projekt Zustandsautomaten enthält, können Sie die Animationsfunktion für Zustandsautomaten verwenden (siehe ASCET-Benutzerhandbuch, Abschnitt „Experimentieren mit Zustandsautomaten“).

Sie können Daten aus dem Experiment in das ASCET-Modell oder auf die Festplatte schreiben; ebenso können Sie Daten von der Festplatte einlesen. Wie das geschieht, ist im ASCET-Benutzerhandbuch, Abschnitt „Manipulieren von Daten“, beschrieben.

Back-Animation beenden:

- Wählen Sie **File** → **Exit**

oder



- klicken Sie auf die Schaltfläche **Exit to Component**.

Die Back-Animation wird beendet und die Experimentierumgebung geschlossen. Das INTECRIO-Experiment läuft jedoch weiter.

5 Realtime-Input-Output-Package

5.1 Einleitung

Das **RealTime-Input-Output-Package** (RTIO-Package) vereinfacht das Zusammenspiel zwischen ASCET und einer ETAS Experimentalsystem-Hardware. Die ETAS Experimentalsystem-Hardware ist eine universell einsetzbare Entwicklungs- und Experimentalplattform auf VMEbus-Basis. Die Ein- und Ausgabe von analogen und digitalen Signalen wird im ES System durch VMEbus Schnittstellenkarten realisiert.

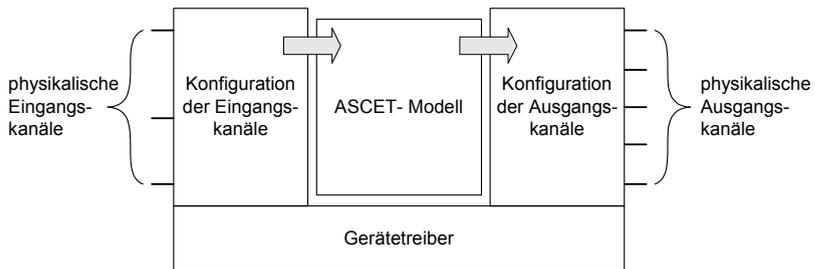
Das RTIO-Package führt zu einer wesentlich einfacheren Einbindung von zusätzlicher I/O-Hardware in ein ASCET Projekt. Eine komfortable Benutzeroberfläche ermöglicht die Konfiguration von mehreren auch interruptgesteuerten Hardware-Komponenten.

Die Einbindung erfolgt durch automatische Generierung von C-Code für das Targetsystem.

5.2 Architektur des RTIO-Package

ASCET-RP erlaubt die einfache Entwicklung eines Prototypen für Ihr ASCET Modell in einer Echtzeitumgebung auf der ES1000 Hardware.

Dafür muss das Modell in die ES1000 Umgebung, bestehend aus dem Simulationsknoten sowie verschiedenen I/O-Karten, eingebettet werden. Auf der logischen Ebene muss das ASCET Modell mit den I/O-Kanälen verbunden werden.



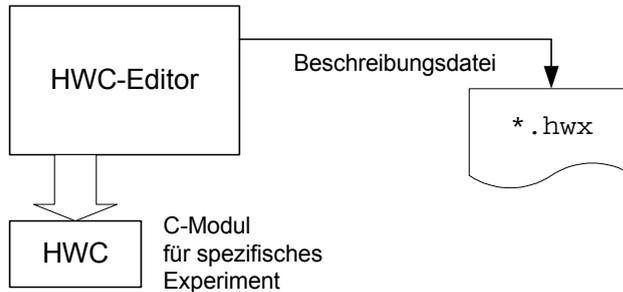
Diese Verbindung geschieht in zwei Schritten. Der HWC-Editor (Kapitel 7) wird verwendet, um die I/O-Karten zu beschreiben und zu konfigurieren, um also die physikalischen Eingangs- und Ausgangskanäle zu konfigurieren. Jeder dieser Kanäle wird dann auf eine ASCET-Message abgebildet, die als Ein- oder Ausgang für das ASCET-Modell dient.

Die Konfiguration wird in einer XML-Beschreibungsdatei mit der Endung * .hwx abgelegt, zusammen mit dem ASCET-Projekt.

Hinweis

*Die Hardwarekonfiguration wird standardmäßig im XML-Format * .hwx abgelegt.
Das bisherige Format * .hwc ist weiterhin gültig und kann als alternatives Format ausgewählt werden. Es wird jedoch empfohlen, das XML-Format zu verwenden, da das Format * .hwc in zukünftigen ASCET-RP-Versionen nicht mehr unterstützt wird.*

Um den Prototypen zu erzeugen, erlaubt der HWC-Editor auch die Generierung von C-Code (Kapitel 8) für das Targetsystem für den ausgewählten Experimenttyp (`physical experiment`, `quantized experiment` oder `implementation experiment`). Dieser C-Code wird in einem C-Modul mit Namen `HWC` (s. Kapitel 5.2.1) gespeichert, das zum Projekt gehört und in der ASCET-Datenbank abgelegt wird.



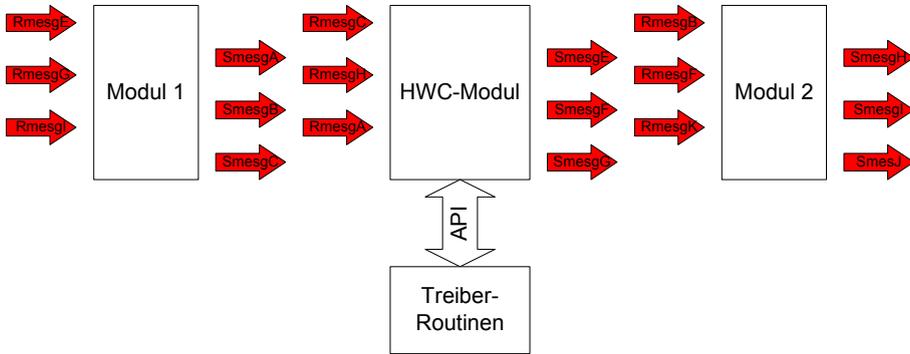
Wenn der Code für das gesamte ASCET-Modell generiert und kompiliert wird, wird die Hardware-Konfiguration ebenfalls kompiliert und mit den nötigen I/O-Treibern verlinkt, um einen ausführbaren Prototypen zu erhalten.

5.2.1 Hardware Configuration Modul

Das besondere Merkmal eines ASCET-Projekts, welches das RTIO-Package nutzt, ist das Hardware-Configuration-Modul (HWC-Modul). Dabei handelt es sich um ein normales C-Code-Modul, das als Container zur Aufnahme des für die HW-Ankopplung notwendigen C-Codes dient. Das RTIO Framework identifiziert das C-Code-Modul durch den Instanznamen `HWC`.

Das HWC-Modul kommuniziert ausschließlich über Messages mit dem übrigen Modell, so dass eine eindeutige Trennung zwischen HW-Ansteuerung und Restmodell gewährleistet ist.

Das nachfolgende Schaubild zeigt die wesentlichen Zusammenhänge zwischen den Modulen in einem RTIO-Projekt.



Die Codegenerierung für das HWC-Modul ist genau auf bestimmte, definierte API¹-Schnittstellen zu den system- und kartenspezifischen Treiber-Routinen abgestimmt. Die Treiber-Routinen werden dem System in Form von Bibliotheken zur Verfügung gestellt. Außer diesen API-Funktionsaufrufen erfolgt auch die Definition der Treiberdaten entsprechend einer einheitlichen Beschreibungsstruktur, die vom Grundsatz her treiberunabhängig ist.

Somit ergibt sich ein Baukastensystem, welches es erlaubt, relativ einfach neue Karten in das RTIO-Package zu integrieren.

5.2.2 Hardware Configuration Editor

Der Hardware-Configuration-Editor (HWC-Editor) bildet den Kern des RTIO-Frameworks.

Der HWC-Editor wird im Edit-Modus aus einem Projekt gestartet und dient zur Spezifikation der gewünschten Hardware Configuration. Eine anschließende Generierungssequenz generiert alle notwendigen Elemente sowie den passenden C-Code damit die Hardware angesteuert werden kann.

¹. API = Application Programming Interface

6 Vorbereitende Maßnahmen

In diesem Abschnitt werden die grundsätzlichen Voraussetzungen beschrieben, die für die Installation und den Betrieb des RTIO-Package notwendig sind.

6.1 Hardware – Experimentalsystem ES1000.x

Voraussetzung für den Betrieb des RTIO-Package ist ein ETAS-Experimentalsystem, welches in unterschiedlichen Baureihen zur Verfügung steht. Üblicherweise kommen die folgenden Varianten in Verbindung mit ASCET zum Einsatz, wobei Mischformen jederzeit zulässig sind:

- ES1000.2
- ES1000.3.

Das RTIO-Framework unterstützt alle ETAS-System-Controllerkarten die sich im Einsatz befinden. Die nachfolgende Abbildung zeigt die üblichen Konfigurationen – davon abweichend sind auch Sonderkonfigurationen möglich.

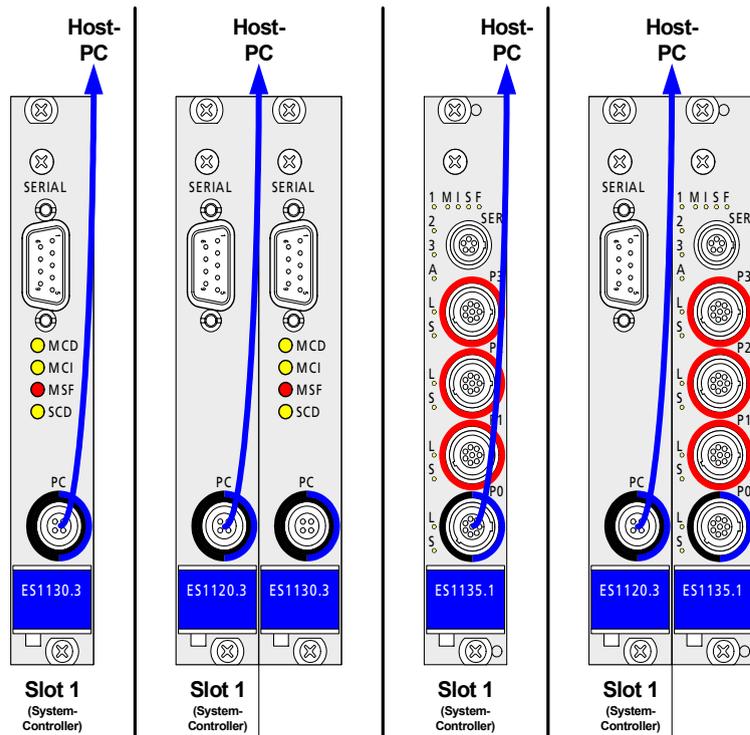


Abb. 6-1 Erlaubte und unterstützte Systemcontroller-Konfigurationen

Steuereinheit ES1120 und Simulationsrechner ES1130/ES1135

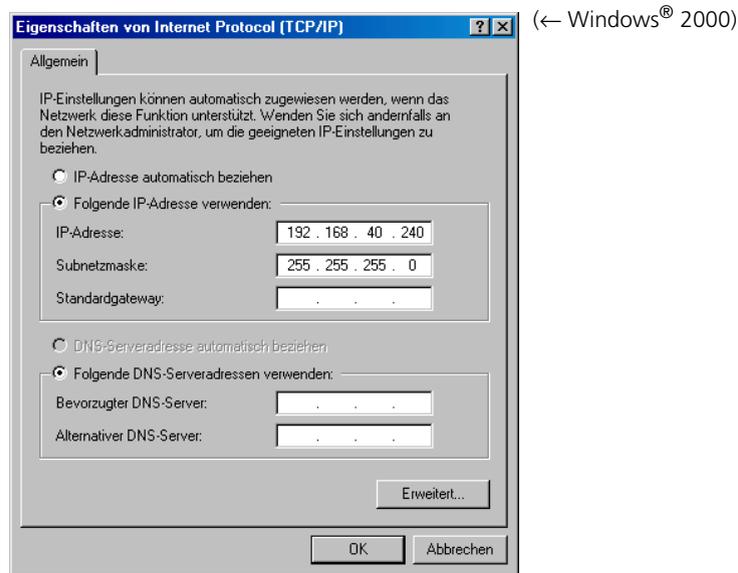
Wird die ES1000.x simultan in der Applikation und zum Rapid Prototyping eingesetzt, wird der Host-PC über ein Ethernet-Kabel mit der Steuereinheit ES1120 verbunden. Die mit ASCET entwickelten Funktionen werden über die Steuereinheit ES1120 auf das PPC-Modul ES1130 oder ES1135 heruntergeladen und dort ausgeführt. Während das Experiment läuft, können Daten mit ASCET gemessen und verstellt werden.

TCP/IP-Protokolleinstellungen

Für den Betrieb der ES1000.x muss das TCP/IP-Protokoll konfiguriert werden. Um Konflikte mit einer ggf. vorhandenen zweiten Netzwerkkarte für das lokale Netzwerk zu vermeiden, sind folgende TCP/IP-Einstellungen vorzunehmen:

Konfiguration der TCP/IP-Protokolleinstellungen:

- Deaktivieren Sie den DHCP-Dienst.
- Geben Sie als IP-Adresse 192 . 168 . 40 . 240 ein.
- Geben Sie als Subnetzmaske 255 . 255 . 255 . 0 ein.



- Verwenden Sie für den DNS-Dienst die lokalen Einstellungen Ihres internen Netzwerks.

- Deaktivieren Sie den WINS-Dienst.
- Stellen Sie sicher, dass die Option „IP-Forwarding“ **nicht** aktiviert ist.

6.2 Besonderheiten der ES1135

Die ES1135 ist eine Weiterentwicklung der ES1130, die dem Anwender einige zusätzliche Funktionen bietet. Diese Funktionen sind in den Abschnitten dieses Kapitels beschrieben.

6.2.1 Non-volatile RAM (NVRAM)

Grundlagen

Eine nicht-flüchtige (non-volatile oder kurz NV) Variable kann wie jede andere ASCET-Variable verwendet werden. Insbesondere kann sie vom Modell gelesen und geschrieben werden, in einem Verstellfenster verstellt und von der Datenerfassung gemessen und aufgezeichnet werden. Das besondere an einer NV-Variablen ist, dass im Fall einer Unterbrechung der Simulation der aktuelle Wert der NV-Variablen verfügbar ist, wenn die Simulation mit demselben Modell erneut gestartet wird. Dies ist besonders nützlich für adaptive Kenngrößen, die im Steuergerätecode oft für selbstlernende Algorithmen verwendet werden, oder für die Speicherung von diagnostischen Ergebnissen.

Das optionale Attribut *non-volatile* (NV) wird für allen primitiven Datentypen von ASCET (Skalare, Arrays, Matrizen, Kennlinien/-felder) unterstützt. Nur ASCET-Variablen können als NV-Variablen konfiguriert werden, C-Code-Variablen werden nicht unterstützt.

Eine NV-Variable kann im Editor einer Klasse oder eines Moduls erzeugt werden. Das geschieht durch Aktivierung der Option **Non-Volatile** im Elementeditor der Variablen.



Achtung

*In Projekten, die den NVRAM verwenden, wird ein **benutzerdefinierter** Init-Prozess erwartet, der prüft, ob alle NV-Variablen sowohl einzeln als auch in Kombination mit anderen NV-Variablen für das aktuelle Projekt gültig sind. Wenn das nicht der Fall ist, müssen alle NV-Variablen mit ihren (sinnvollen) Standardwerten initialisiert werden.*

*Wegen des NVRAM-Sicherungskonzepts ist dies **absolut notwendig**, wenn Projekte in einem Umfeld verwendet werden, wo es durch die Initialisierung mit ungeeigneten Werten zu Sach- oder Personenschäden kommen kann (z.B. bei der Verwendung in Fahrzeugen oder an einem Prüfstand).*

Im Adressraum des Hauptprozessors der ES1135 (IBM750GX) stehen 64 KByte nichtflüchtiges RAM (NVRAM) zur Verfügung. In diesem Speicherbereich können Daten gespeichert werden, die über einen Spannungsausfall oder einen Einschaltzyklus hinweg erhalten bleiben sollen.

Aus Gründen der Performanz (Zugriffe auf den NVRAM sind signifikant langsamer als Zugriffe auf den normalen, flüchtigen RAM und können darüber hinaus auch nicht im Cache-Speicher gepuffert werden), werden NV-Variablen nicht direkt dem NVRAM zugeordnet. Stattdessen werden sie wie normale, flüchtige Variablen dem normalen RAM zugeordnet und periodisch im NVRAM gesichert (Auto-Update-Modus). Die Periode ist standardmäßig auf 10 Sekunden gesetzt; sie kann mit der API-Methode

```
uint32 nvramSetUpdateInterval(uint32 interval_sec)
```

(siehe Kapitel 13.2) im Bereich von 1 bis 30 Sekunden eingestellt werden. Die Sicherung erfolgt in der Idle-Task und hat keine Rückwirkung auf das Echtzeitverhalten des Modells.

Aus Gründen der Datenkonsistenz ist das NVRAM als Wechsellpuffer organisiert, was die vorhandene Kapazität halbiert. Darüber hinaus gibt es einen gewissen Overhead, der die NVRAM-Kapazität, die dem ASCET-Modell zur Verfügung steht, auf etwas unter 32 KByte begrenzt.

Die Firmware der ES1135 stellt sicher, dass das Kapazitätslimit eingehalten wird. Wenn die kumulative Größe der NV-Variablen die NVRAM-Kapazität übersteigt, erscheint beim Start des Experiments die Fehlermeldung „NVRAM Overflow“ im ASCET-Monitorfenster. In diesem Fall wird das NVRAM nicht verwendet, das heißt, es werden keine Werte dorthin geschrieben. Um das NVRAM zu verwenden, muss der Anwender Anzahl und/oder Größe der NV-Variablen im Modell reduzieren.

Die Daten im NVRAM enthalten weder Adressinformationen noch die Namen der Variablen. Sie passen daher nur dann zu dem Modell, wenn die Struktur der NV-Variablen im Modell sich nach dem letzten Speichern nicht geändert hat. Um das zu überprüfen, wird bei der Codegenerierung ein spezieller *NV-Identifizier* erzeugt.

Dieser NV-Identifizier ändert sich bei folgenden Aktionen:

- Änderung des Instanznamens eines beliebigen NV-Elements auf Projekt-, Modul-, Klassen- oder Unterklassen-Ebene.
- Änderung der Implementierung eines beliebigen NV-Elements auf Projekt-, Modul-, Klassen- oder Unterklassen-Ebene, und zwar:
 - Codegenerierungsoption `Physical Experiment` – Wechsel des Implementierungsdatentyps zwischen `cont` und `sdisc/udisc`

- Codegenerierungsoption `Implementation Experiment` –
Änderungen am Implementierungsintervall
- Änderung der Formel-Parameter eines beliebigen NV-Elements.
Wenn der Name oder der Kommentar der Formel geändert wird, bleibt NV-Identifizier gleich.
- Bei einer NV-Enumeration:
 - Änderung der Reihenfolge der Enumeratoren
 - Änderung der Namen (Werte) der Enumeratoren
 - Entfernen/Hinzufügen von Enumeratoren
 - Auswahl einer anderen Enumeration, auch wenn diese dieselben Enumeratoren hat
- Änderung der maximalen Größe von mehrdimensionalen NV-Elementen (Array, Matrix, Kennlinie/-feld).
- Änderung der Einstellungen für die Limitierung der Implementierung beim Implementierungsexperiment (`Codegenerierungsoption Implementation Experiment`).
- Entfernen und Hinzufügen von NV-Elementen auf Projekt-, Modul-Klassen- oder Unterklassen-Ebene.
- Entfernen und Hinzufügen von Zuständen in einem Zustandsautomaten.

Bei folgenden Aktionen ändert sich der NV-Identifizier *nicht*:

- Umbenennung des Projekts.
- Umbenennung der im Projekt enthaltenen Module oder Klassen.
- Umbenennung der Instanzen beliebiger normaler (flüchtiger) Elemente auf Projekt-, Modul-, Klassen- und Unterklassen-Ebene.
- Änderung des Namens oder Kommentars der Formel eines beliebigen NV-Elements.
Wenn die Formel-Parameter geändert werden, ändert sich der NV-Identifizier.
- Umbenennung einer Enumeration.
- Änderungen der Daten von NV-Elementen und normalen Elementen.
- Änderung der tatsächlichen Größe von mehrdimensionalen Elementen (Array, Matrix, Kennlinie/-feld).
- Änderung des Speicherbereichs von NV-Elementen und normalen Elementen.

Simulation starten: Nachdem der Modellcode auf das Target geladen wurde, werden die NV-Variablen mit ihren Standardwerten initialisiert, wenn es im NVRAM keine passenden Daten gibt. Keine passenden Daten bedeutet, dass der NV-Speicher leer oder inkonsistent (verifiziert über eine Prüfsumme) ist, oder dass die NV-Daten nicht zum geladenen Modell passen (verifiziert über den NV-Identifizier).

In dem Fall, dass passende Daten im NV-Speicher vorhanden sind, werden die Variablen mit diesen Daten initialisiert, bevor das Experiment gestartet werden kann (Start ERCOS).

Simulation beenden: Wenn die Simulation beendet wird (Stop ERCOS), werden die Werte der letzten Sicherung der NV-Variablen dauerhaft im NVRAM gespeichert. Selbst wenn die Stromversorgung des Targets unterbrochen wird, oder wenn der Code neu heruntergeladen wird, kann die Simulation mit der zuletzt gesicherten Version der NV-Variablen weitergeführt werden.

Wie bereits erwähnt, werden im Auto-Update-Modus die NV-Variablen periodisch in das NVRAM gesichert. Um sicherzustellen, dass sich nach dem Ende der Simulation auch wirklich die aktuellen Werte der NV-Variablen – und nicht die Werte der letzten, periodischen Sicherung – im NVRAM befinden, sollte die API-Funktion

```
void nvramUpdateMemoryExit(void)
```

am Ende der Exit-Task (Task mit Betriebsmodus `inactive`) aufgerufen werden.

Wenn im aktuellen Modell keine NV-Variablen vorkommen, bleibt der Inhalt des NVRAM unverändert.

Modell mit NV-Variablen im FLASH-Speicher: Ein Simulationsmodell mit NV-Variablen im FLASH-Speicher des Simulationscontrollers wird hochgefahren, sobald das Gerät eingeschaltet wird. Die potentiell passenden NV-Daten werden für die Initialisierung der NV-Variablen vor dem Start der Simulation verwendet.

Anzeige, ob das Modell mit den Standardwerten der NV-Variablen

arbeitet: Ob das Modell mit den Standardwerten der NV-Variablen (spezifiziert in einem ASCET-Dateneditor) startet, oder ob die Variablen aus dem NVRAM initialisiert werden, kann auf zwei Arten festgestellt werden. In der Experimentierumgebung wird eine entsprechende Meldung im Target Debugger-Fenster angezeigt. Aus dem Modell heraus liefert die API-Funktion

```
uint8 nvramCheckForInitializedVars(void)
```

dieselbe Information.

Den Inhalt des NVRAM löschen: Um die Initialisierung eines Modells mit dem Inhalt des NVRAM trotz passendem Programm-Identifizierer zu verhindern, ist es möglich, den Speicherinhalt des NVRAM zu löschen. Dies erzwingt die Initialisierung der NV-Variablen mit ihren Standardwerten. Die grafische Oberfläche unterstützt diese Möglichkeit derzeit nicht, aber die API-Funktion

```
uint32 nvramClear(void)
```

erlaubt es, das NVRAM aus dem Modell heraus zurückzusetzen (siehe Kapitel 13.2 „API-Funktionen (NVRAM)“).

Datenkonsistenz

Wenn die Simulation durch einen Stromausfall oder einen Systemabsturz unterbrochen wird, enthält das NVRAM Werte der NV-Variablen. Die Aktualität der Werte hängt vom Zeitpunkt der letzten automatischen oder manuell aus dem Modell heraus ausgelösten Sicherung ab.

Um die Datenkonsistenz des NVRAM-Inhalts bei einem unerwarteten Ende der Simulation zu garantieren, können verschiedene Strategien verwendet werden, die im folgenden vorgestellt werden.

Der Grad an Konsistenz kann über folgende API-Funktion eingestellt werden:

```
uint32 nvramSetConsistencyLevel(T_consistencyLevel  
level)
```

Keine Konsistenz: Die Aktualisierung des NVRAM geschieht ohne Rücksicht auf Konsistenz innerhalb einer NV-Variablen oder zwischen einzelnen NV-Variablen.

Niedrige Konsistenz (einzelne Variablen): Niedrige Konsistenz (low-level consistency) bedeutet, dass Datenkonsistenz innerhalb der NV-Variablen (Skalare, Arrays, Matrizen, jedoch nicht Kennlinien/-felder) garantiert wird.

Hierzu muss erwähnt werden, dass die Aktualisierung von Kennlinien/-feldern nicht atomar geschehen kann (im Sinne der niedrigen Konsistenz).

Hohe Konsistenz (zwischen Variablen, nach Abschluss einer Task):

Hohe Konsistenz (high-level consistency) bedeutet, dass alle NV-Variablen innerhalb der Idle-Task ohne Unterbrechung durch das Modell aktualisiert werden.

Die Aktualisierung einer Gruppe von NV-Variablen sollte atomar sein. Beispielsweise verwendet ein selbstlernender Algorithmus verschiedene Variablen, hierbei muss sichergestellt sein, dass die Daten der verschiedenen Variablen im NVRAM aus dem selben Berechnungszyklus kommen.

Modell-kontrollierte Konsistenz (zwischen Variablen und über mehrere Taskzyklen): Der Mechanismus der hohen Konsistenz garantiert Konsistenz nur dann, wenn alle Manipulationen von NV-Variablen innerhalb eines Taskzyklus vorgenommen werden. Es kann aber Fälle geben, in denen diese Manipulationen über mehrere Taskzyklen hinweg erfolgen (z.B. die Aktualisierung einer adaptiven Kennlinie verteilt über mehrere Tasks). Die Firmware der ES1135 kann dies nicht feststellen, daher muss die Kontrolle der NVRAM-Aktualisierung durch das Modell erfolgen.

Zu diesem Zweck kann die automatische Aktualisierung mit folgender API-Funktion unterbunden werden:

```
uint32 nvramDisableAutoUpdate(void)
```

Die manuelle Aktualisierung des kompletten Satzes von NV-Variablen wird mit folgendem Befehl gestartet:

```
uint32 nvramManualUpdateBackground(void)
```

Selbst mit der manuellen Aktualisierung ist es nicht erlaubt, das komplette System zu blockieren und somit das Echtzeitverhalten zu beeinflussen, bis die Aktualisierung beendet ist. Daher springt die Funktion für die manuelle Aktualisierung sofort zurück ins Programm, und die Aktualisierung läuft im Hintergrund ab. Das Modell kann den Status der manuellen Aktualisierung mit folgendem Befehl abfragen:

```
uint8 nvramCheckRunningUpdate(void)
```

Diese Funktion gibt `true` zurück, wenn die Aktualisierung läuft. Der Anwender bzw. das Modell ist dafür verantwortlich, dass die NV-Variablen während der Aktualisierung nicht verändert werden.

Das Umstellen von manueller auf automatische Aktualisierung kann mit folgender API-Funktion realisiert werden:

```
uint32 nvramEnableAutoUpdate(void)
```

Eine selektive Aktualisierung von NV-Variablen wird nicht unterstützt.

Wenn im aktuellen Modell keine NV-Variablen verwendet werden, bleibt der NVRAM-Inhalt unverändert.

Defekter NVRAM-Inhalt: Bei defektem NVRAM-Inhalt, z.B. wenn der Prüfsummenvergleich fehlschlägt, wird der Anwender gewarnt. Dies geschieht über eine textuelle Warnung in der Experimentierumgebung (oder dem ASCET-Monitorfenster).

NVRAM-Cockpit

Die NVRAM-API (siehe Kapitel 13.2) bietet verschiedene Funktionen, mit denen Sie die Aktualisierung des NVRAM-Inhalts steuern können. Diese Funktionen sind im laufenden Experiment in einem speziellen Fenster, dem *NVRAM-Cockpit*, zugänglich. Auch Änderungen, die Sie mit Hilfe der API-Funktionen vornehmen, werden in das NVRAM-Cockpit übernommen.

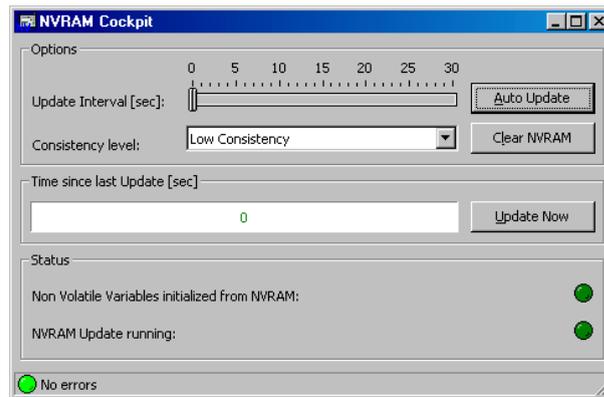
Arbeiten mit dem NVRAM-Cockpit:

- Wählen Sie im Experiment **Experiment** → **NVRAM Cockpit**

oder

- klicken Sie auf die Schaltfläche **Open NVRAM Cockpit**.

Das NVRAM-Cockpit öffnet sich.



- Verwenden Sie die Schaltelemente gemäß Ihren Bedürfnissen.
- Schließen Sie das NVRAM-Cockpit mit **Close**.

Das NVRAM-Cockpit enthält folgende Schaltelemente:

- **Update Interval [sec]**

Mit diesem Schieberegler stellen Sie das Intervall (in Sekunden) für die automatische Aktualisierung des NVRAM-Inhalts ein. Sie können bis zu 30 Sekunden einstellen, voreingestellt ist ein Intervall von 10 Sekunden.

Der Schieberegler ist *nur* aktiviert, wenn die automatische Aktualisierung eingeschaltet ist.

- **Auto Update**

Diese Schaltfläche schaltet die automatische Aktualisierung ein und aus. Die automatische Aktualisierung ist eingeschaltet, wenn die Schaltfläche eingedrückt erscheint (a), und abgeschaltet, wenn die Schaltfläche erhoben erscheint (b).



Hinweis

Die automatische Aktualisierung des NVRAM-Inhalts funktioniert nur, solange das Experiment läuft. Wenn Sie das Experiment mit **Experiment** → **Stop ERCOS** oder mit der Schaltfläche **Stop ERCOS** angehalten haben, wird der NVRAM-Inhalt nicht mehr automatisch aktualisiert.

- **Consistency level**

In diesem Kombikästchen wählen Sie den Konsistenzgrad der Aktualisierung; siehe auch „Datenkonsistenz“ auf Seite 77.

- **Clear NVRAM**

Mit dieser Schaltfläche löschen Sie den NVRAM-Inhalt.

Wenn Sie auf **Clear NVRAM** klicken, während die automatische Aktualisierung läuft, wird der NVRAM-Inhalt zwar gelöscht, aber spätestens nach Ablauf des Intervalls wieder beschrieben.

- **Update Now**

Mit dieser Schaltfläche starten Sie die Aktualisierung des NVRAM-Inhalts manuell.

Die Schaltfläche ist nur verfügbar, wenn die automatische Aktualisierung abgeschaltet ist.

Hinweis

Die manuelle Aktualisierung des NVRAM-Inhalts funktioniert auch, wenn Sie das Experiment mit **Experiment** → **Stop ERCOS** oder mit der Schaltfläche **Stop ERCOS** angehalten haben.

Das NVRAM-Cockpit enthält folgende Anzeigen:

- **Time since last Update [sec]**

Diese Balkenanzeige zeigt an, wieviel Zeit seit der letzten Aktualisierung vergangen ist. Der komplette Balken entspricht 30 Sekunden; wenn diese Zeit bei deaktivierter automatischer Aktualisierung überschritten wird, wird nur noch die Zahl hochgezählt.

Die Zählung der Sekunden läuft auch dann weiter, wenn das Experiment gestoppt wird, da die Zeit auch dann weiterläuft. Nur bei einer Aktualisierung nach dem Stoppen des Experiments wird der Wert zurückgesetzt und neu gezählt.

Der Balken ist grün, solange die Zeit seit der letzten Aktualisierung kleiner als 30 s ist, und rot, wenn diese Zeit überschritten wird. Ausnahme: Vor dem Überlauf wurde das Experiment angehalten (**Stop ERCOS**); dann wird der Balken bei Überlauf gelb.

- **Non Volatile Variables initialized from NVRAM**

Diese Anzeige ist hellgrün, wenn die NV-Variablen mit dem NVRAM-Inhalt initialisiert werden (a), und dunkelgrün, wenn die NV-Variablen mit ihren Standardwerten initialisiert werden (b).

(a) Non Volatile Variables initialized from NVRAM: 

(b) Non Volatile Variables initialized from NVRAM: 

- **NVRAM Update running**

Diese Anzeige ist hellgrün, wenn gerade eine Aktualisierung des NVRAM-Inhalts läuft..

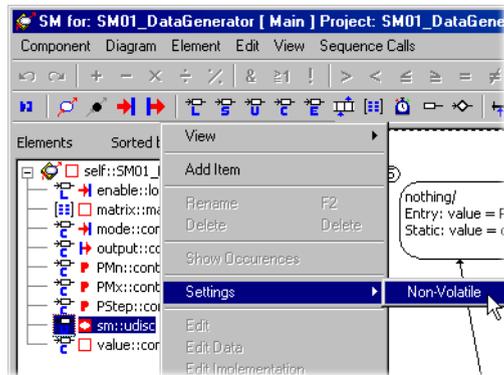
NVRAM Update running: 

Tipps

Die folgenden Tipps sind für die Arbeit mit dem NVRAM von Vorteil.

- **Zustandsvariable als *non-volatile***

Die erweiterte NVRAM-Unterstützung bietet die Möglichkeit, der Zustandsvariable `sm` eines Zustandsautomaten das Attribut *non-volatile* zuzuweisen. Klicken Sie dazu in der Liste „Elements“ des Zustandsautomateneditors mit der rechten Maustaste auf die Variable `sm` und wählen Sie **Settings** → **Non-Volatile** aus dem Kontextmenü.



- **Tatsächliche Größe von mehrdimensionalen Elementen**

Die tatsächliche Größe (Attribut „Current Size“) von mehrdimensionalen Elementen (Array, Matrix, Kennlinie/-feld) wird im NVRAM gesichert.

Änderungen der tatsächlichen Größe im Modell (offline) werden im heruntergeladenen Programm nur wirksam, nachdem der NVRAM-Inhalt gelöscht wurde (z.B. über das NVRAM-Cockpit), oder wenn der NVRAM-Inhalt inkonsistent mit dem Programm ist.

- **Flash-Projekt mit NVRAM auf der ES1000**

Denken Sie daran, dass bei jedem Start der ES1000 das Programm aus dem Flash-Speicher hochgefahren wird. Wenn dieses Projekt NV-Variablen enthält, verwendet es den NVRAM. Ein nachfolgendes Herunterladen irgendeines anderen Programms mit NV-Variablen führt zu einem (unerwarteten) Reset des NVRAM-Inhalts.

6.2.2 Watchdog

Zur Integration in ein Sicherheitskonzept für das Rapid Prototyping-System bietet die ES1135 eine Hardware Watchdog-Funktion. Der Watchdog ist eine unabhängige Steuereinheit, die den Hauptprozessor auf der ES1135 überwacht. Dabei wird eine festgelegte Datensequenz in bestimmten Zeitabständen in eine Speicherzelle (Watchdog Service-Register) geschrieben. Bei Ablauf einer maximalen Zeitdauer (Watchdog-Periode) ohne einen korrekten Schreibzugriff (Watchdog-Service) auf das Watchdog Service-Register wird im Prozessor eine Ausnahmebehandlung (Event) ausgelöst.

Der ES1135 HW Watchdog kann in zwei Modi betrieben werden:

1. Sicherheitsorientierte Betriebsart (Safety-Modus)
2. Flexible Betriebsart mit mehr Funktionen (Reduced Safety, Enhanced Function- oder RSEF-Modus)

Im RSEF-Modus können folgende Watchdog-Einstellungen zur Laufzeit umkonfiguriert werden:

- Event-Konfiguration
Definiert die Ausnahmebehandlung bei Ablauf des Watchdogs. Über die Event-Konfiguration kann der Watchdog auch deaktiviert werden.
- Watchdog-Periode
Definiert die Zeitdauer bis zum Ablauf des Watchdogs, wenn kein erneuter Watchdog-Service erfolgt.
- Modus-Umschaltung
Der Safety-Modus wird mit einer frei wählbaren Watchdog-Perioden- und Eventkonfiguration eingeschaltet und kann danach nicht mehr umkonfiguriert oder verlassen werden. Zuvor sollte der Watchdog-Service so programmiert werden, dass es nicht zu einem ungewollten Watchdog-Event kommt.

Nach dem Einschalten der Versorgungsspannung wird der Watchdog in den RSEF-Modus versetzt und ausgeschaltet.

Watchdog-Service

Der Watchdog muss bedient, d.h. zurückgesetzt werden, bevor die Watchdog-Periode abläuft. Andernfalls tritt das ausgewählte Watchdog-Ereignis ein. Es ist die Aufgabe des Modellentwicklers, den Aufruf der Servicefunktion an einer Stelle zu platzieren, an der eine Fehlfunktion des Modells entdeckt werden kann.

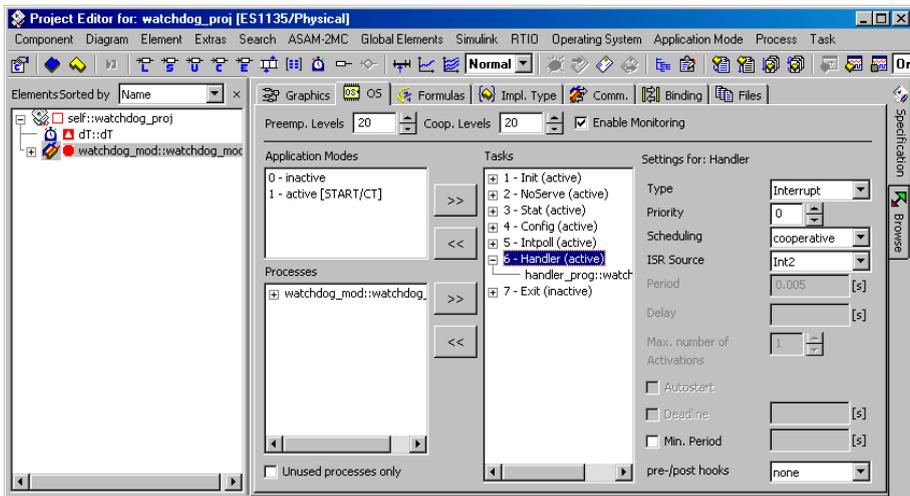
Die Firmware des Simulationscontrollers stellt einen automatischen Watchdog Service-Mechanismus bereit, der den Watchdog alle 30 ms bedient, wenn die Interrupts nicht vom Modell deaktiviert werden. Auf diese Weise wird, unter der Annahme, dass das Betriebssystem korrekt läuft, der Watchdog regelmäßig bedient. Dies ist für viele Anwendungsfälle ausreichend.

Interruptsteuerung

Speziell für Debugging- und Überwachungszwecke kann der Watchdog so eingerichtet werden, dass er einen Interrupt des Simulationsprozessors auslöst, wenn die Periode abgelaufen ist.

Dieser Interrupt kann entweder angefragt oder an den internen Interruptcontroller geleitet werden. Ein Watchdog-Interrupt ist gesperrt und erfordert eine explizite Bestätigung. Funktionen für schnelles Ein- und Ausschalten der Interruptquelle sind verfügbar. Diese Funktionen haben nur Einfluss auf die Weitergabe des Interrupts.

Der Watchdog-Interrupt wird in ASCET auf eine Hardwaretask, den Watchdog-Handler, abgebildet. Im OS-Editor von ASCET müssen dafür der Tasktyp `Interrupt` und die ISR-Quelle `Int2` ausgewählt werden.



Der Watchdog-Handler läuft unterhalb des `ERCOSEK`-Levels, aber oberhalb anderer Hardware-Interrupts (z.B. vom VMEbus), das heißt, der Watchdog-Interrupt wird auch dann bearbeitet, wenn gerade ein anderer Hardware-Interrupt bearbeitet wird. Die Interruptbestätigung erfolgt in der Firmware der

ES1135, sie sollte daher nicht innerhalb des Watchdog-Handlers erfolgen. Der zur Verfügung stehende Satz von ERCOS^{EK}-Aufrufen ist innerhalb des Watchdog-Handlers nicht beschränkt.

Wenn ein Watchdog-Interrupt auftritt, wird der Watchdog automatisch 250 μ s nach der Überlaufsituation wieder hergestellt und beginnt einen neuen Zyklus mit der zuvor ausgewählten Periode. Diese Wiederherstellungszeit kann verkürzt werden, indem mit `wdService()` ein normaler Watchdog-Service durchgeführt wird.

Eine detaillierte Beschreibung der Programmierschnittstelle (API) des Watchdogs finden Sie in Kapitel 13.3 „API-Funktionen (Watchdog)“ auf Seite 420.

6.2.3 LEDs

Die auf der Frontplatte der ES1135 angebrachten LEDs unterscheiden sich in System-LEDs (M, I, S, F, A, L, S) und frei programmierbaren LEDs (1, 2, 3).

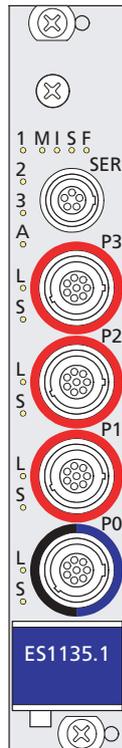


Abb. 6-2 ES1135 – Frontplatte

Die Bedeutung der System-LEDs entnehmen Sie bitte dem Hardware-Handbuch.

Die frei programmierbaren LEDs können entweder über die RTIO (Device ES1135-LED, s. Kapitel 10.2) oder über die Programmierschnittstelle (API, s. Kapitel 13.4 „API-Funktionen (ES1135-LEDs)“) angesteuert werden.

6.3 System-Software

Das im vorliegenden Handbuch beschriebene RTIO-Package ist in den nachfolgenden Software-Produkten enthalten oder verfügbar:

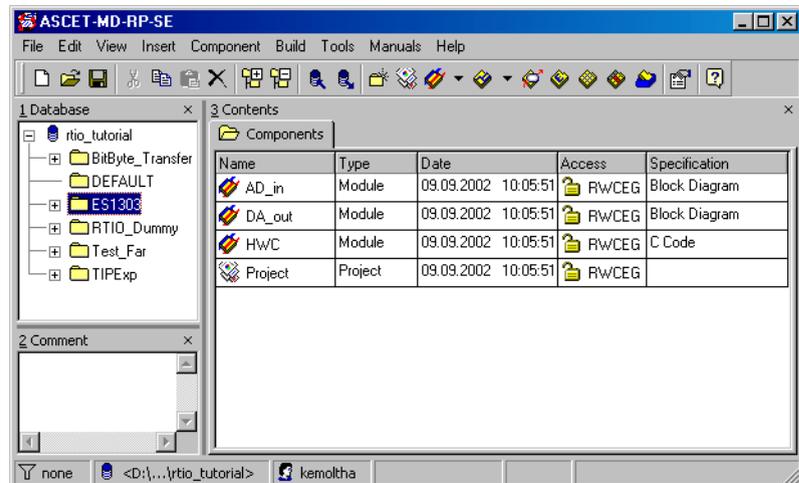
- ASCET ab Version 5.1.1 mit installiertem Zusatzprodukt ASCET-RP ab Version V5.4.0.

6.3.1 System Root Path

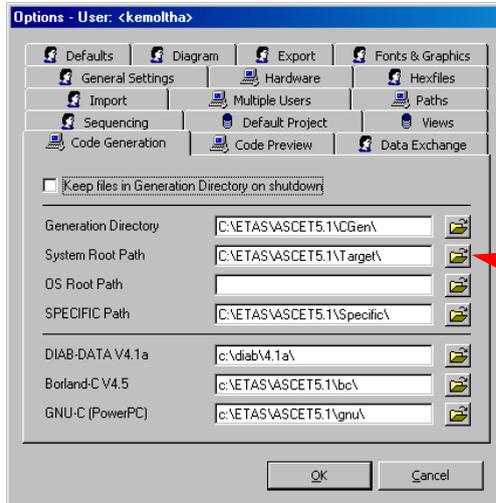
Mit ASCET-RP wird im Unterverzeichnis Ihrer ASCET Installation das neue Target-Verzeichnis `ASCET5.1\target` installiert. Der Systempfad muss auf das Target-Unterverzeichnis eingestellt sein.

Festlegen des System-Hauptpfades:

- [Der System-Hauptpfad \(System Root Path\) wird im Fenster „Options“ von ASCET gesetzt.](#)



- Wählen Sie im ASCET-Komponentenmanager **Tools** → **Options**.
Das Fenster „Options“ erscheint.



- Im Fenster „Options“ muss \TARGET als „System Root Path“ ausgewählt sein.

6.3.2 C-Code-Modul

Jedes ASCET-Projekt, für das eine RTIO Hardware-Ankopplung durchgeführt werden soll, benötigt ein separates C-Code-Modul, welches in der Datenbank angelegt werden muss.

Hinweis

Im Gegensatz zu vorhergehenden RTIO-Versionen sind die Einschränkungen bezüglich des Modul-Namens („HWC“) in der Datenbank nun aufgehoben, d.h. es sind beliebige Namen zulässig.

6.3.3 Projekt

Einige wenige Belange müssen beachtet werden, damit eine RTIO Hardware-Ankopplung durchgeführt werden kann:

- Ein zuvor angelegtes C-Code-Modul muss zum Projekt hinzugefügt werden, welches exklusiv für die RTIO-Codegenerierung verwendet werden kann (siehe auch voriger Abschnitt). Das Modul muss den Instanznamen „HWC“ erhalten.

- Abhängig vom eingesetzten Systemcontroller-Typ (ES1130 oder ES1135) muss der gewünschte Target-Typ gewählt werden.
- Die Task-Liste im Register „OS“ des Projekteditors sollte folgende Tasks beinhalten, damit die RTIO Ankopplung möglichst einfach realisiert werden kann:

Task-Name	Task-Typ	Application Mode	Aufgabe
Init	Init	active	Wird vom RTIO Framework zur Initialisierung der Hardware-Treiber benötigt
Config	Software	active	nur in speziellen Fällen für die Rekonfigurierung der Hardware benötigt
Exit	Init	inactive	Wird zum Freigeben von Treiber-Ressourcen benötigt

Darüber hinaus benötigen einige HW-Komponenten noch zusätzliche Tasks – darauf wird dann bei der Beschreibung der entsprechenden Komponente eingegangen.

- Alle Messages, mit denen das RTIO HWC-Modul mit den übrigen Komponenten im Projekt (weitere Module, globale Messages) kommuniziert, müssen von der Gegenseite als „EXPORTED“ angelegt werden. Üblicherweise ist das für Send- oder Send&Receive-Messages standardmäßig der Fall; bei Receive-Messages muss darauf geachtet werden, dass dies explizit durchgeführt wird.

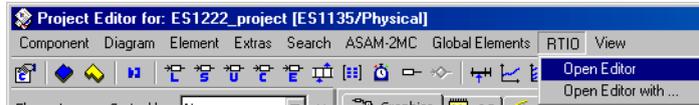
Abweichend von den vorangegangenen RTIO Versionen ist nun auch der Zugriff auf Messages möglich, die innerhalb hierarchischer Module angelegt sind.

7 HWC-Editor

Der HWC-Editor bildet den Kern des RTIO-Frameworks und dient zur Definition und Beschreibung von Hardware-Konfigurationen.

7.1 Öffnen des HWC-Editors

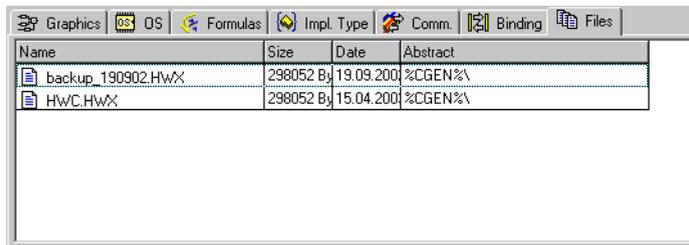
Zunächst muss der Projekteditor für das gewünschte ASCET-Projekt geöffnet werden. Bei ordnungsgemäß installiertem RTIO-Package findet sich in der Hauptmenüzeile das Menü **RTIO**. Über die Menüfunktionen **Open Editor** oder **Open Editor with** kann nun der HWC-Editor geöffnet werden.



Das erstmalige Öffnen des HWC-Editors innerhalb einer ASCET-Session dauert etwas länger, da zunächst einige Systembestandteile nachgeladen werden müssen. (Um den Systemstart zu beschleunigen und die Ressourcen zu schonen, werden einige Systemerweiterungen erst dann nachgeladen, wenn sie tatsächlich benötigt werden.)

Wird der HWC Editor mit **RTIO** → **Open Editor** geöffnet, wird anschließend das Register „Files“ des ASCET-Projekts nach einer vorhandenen Datei `HWC.HWX` (oder `HWC.HWC`) durchsucht.

Ist diese vorhanden, so wird sie automatisch in den Editor geladen. Die Datei `HWC.HWX` (oder `HWC.HWC`) ist immer diejenige, mit der die letzte Codegenerierung des HWC-Moduls durchgeführt wurde.

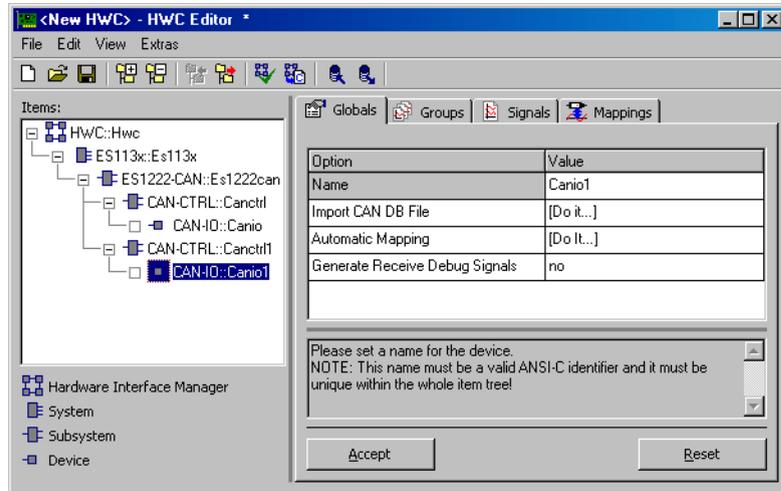


Die Menüfunktion **Open Editor with** ermöglicht die Auswahl einer bestimmten Hardware-Konfigurationsdatei, die anstelle der Datei `HWC.HWX` (oder `HWC.HWC`) geladen wird.

Welches der beiden Formate `*.hwx` und `*.hwc` als Standard verwendet wird, legen Sie im Fenster „HWC Editor Options“, Register „Files“, fest. Das Fenster können Sie mit der Menüfunktion **Extras** → **Options** öffnen.

7.2 Bedienelemente

In diesem Abschnitt werden die Bedienelemente des HWC-Editors vorgestellt.



Die Benutzeroberfläche besteht aus einer Schaltflächenleiste, der Liste „Items“ und einem Satz Registerkarten, über die die meisten Modifikationen und Eingaben vorgenommen werden können.

Unterhalb der Registerkarten ist ein Textfeld, das Kurzinformationen zur ausgewählten Zelle enthält. Mit den Schaltflächen **Accept** und **Reset** übernehmen bzw. verwerfen Sie vorgenommene Änderungen.

7.2.1 Schaltflächen

In diesem Abschnitt finden Sie eine Beschreibung der über die Schaltflächen der Schaltflächenleiste erreichbaren Funktionen.

Hinweis

Die genaue Beschreibung der einzelnen Funktionen finden Sie im Abschnitt „Hauptmenü“ auf Seite 95.

Erstellen einer neuen Hardware-Konfiguration



- Zum Erstellen einer neuen Hardware-Konfiguration klicken Sie auf die Schaltfläche **New** oder
- wählen Sie **File** → **New**.

Öffnen einer Hardware-Konfiguration



- Zum Öffnen einer Hardware-Konfiguration klicken Sie auf die Schaltfläche **Open**

oder

- wählen Sie **File** → **Open**.

Speichern einer Hardware-Konfiguration



- Zum Speichern einer Hardware-Konfiguration klicken Sie auf die Schaltfläche **Save**

oder

- wählen Sie **File** → **Save**.
- Wählen Sie **File** → **Save As**, wenn Sie die Hardware-Konfiguration unter einem beliebigen Namen abspeichern wollen.

Expand all



- Um die Liste „Items“ vollständig aufzuklappen, klicken Sie auf die Schaltfläche **Expand all**

oder

- Wählen Sie **View** → **Expand all**.

Collapse all



- Um die Liste „Items“ vollständig zuzuklappen, klicken Sie auf die Schaltfläche **Collapse all**

oder

- Wählen Sie **View** → **Collapse all**.

Ein neues Item hinzufügen



- Um ein neues Element in die Liste „Items“ einzufügen, klicken Sie auf die Schaltfläche **Add Item**

oder

- wählen Sie **Edit** → **Add Item**.

Ein Item entfernen



- Um ein Element zu entfernen, selektieren Sie es in der Liste und klicken Sie auf die Schaltfläche **Delete Item**

oder

- Wählen Sie **Edit** → **Delete Item**.

Überprüfen der Hardware-Konfiguration



- Zum Überprüfen einer Hardware-Konfiguration klicken Sie auf die Schaltfläche **Check Hardware Configuration**

oder

- wählen Sie **Extras** → **Check Hardware Configuration**.

Starten der Codegenerierung



- Zum Starten der Codegenerierung klicken Sie auf die Schaltfläche **Generate Code For Current Experiment**

oder

- wählen Sie **Extras** → **Generate Code** → **For Current Experiment**.

Importieren einer Hardware-Konfiguration



- Zum Importieren einer Hardware-Konfiguration klicken Sie auf die Schaltfläche **Import**

oder

- wählen Sie **File** → **Import**.

Exportieren einer Hardware-Konfiguration



- Zum Exportieren einer Hardware-Konfiguration klicken Sie auf die Schaltfläche **Export**

oder

- wählen Sie **File** → **Export**.

7.2.2 Liste „Items“

Die baumartige Liste „Items“ dient zur Definition bzw. Anzeige der Hardware-Struktur.

Der Ausdruck „Item“ bezeichnet ganz allgemein ein Hardware-Konfigurationselement. Das kann entweder eine gesamte Einschubkarte sein oder nur eine funktionell in sich abgeschlossene Einheit.

Hinweis

Eine Hardware ist dann vollständig spezifiziert, wenn jeder Endknoten des HW-Baums vom Typ „Device“ ist.

Die Register rechts neben der Liste „Items“ zeigen immer die zum ausgewählten Item zugehörigen Einstellungen an.

7.2.3 Konfigurationsregister

Je nach dem ausgewählten Item in der Liste „Items“ werden unterschiedliche Registerkarten angezeigt, über die die individuellen Einstellungen für die Items vorgenommen werden können.

Ist das gewählte Item vom Typ „Device“, so werden 4 Register eingeblendet; bei allen anderen Elementtypen gibt es nur das eine Register „Globals“.

Die Aufgabenverteilung auf die Register ist wie folgt:

- **Globals**

Hier finden sich alle globalen Einstellungen, die das gesamte Element betreffen (z.B. VMEbus Adresse der I/O-Karte).

- **Groups**

Hier befinden sich alle Einstellungen, die eine Signalgruppe betreffen (z.B. Verstärkungsfaktor für einen mehrkanaligen AD-Wandler oder den Identifier einer CAN-Botschaft).

Hinweis

Eine Signalgruppe zeichnet sich dadurch aus, dass alle in ihr enthaltenen Signale die gleiche Übertragungsrichtung (send oder receive) aufweisen und zum gleichen Zeitpunkt oder im selben Raster erfasst werden.

- **Signals**

In diesem Register werden alle signalspezifischen Einstellungen vorgenommen (z.B. Umrechnungsformel für ein Signal).

- **Mappings**

In diesem Register erfolgt die Zuordnung zwischen einem Signal und einer ASCET Message. Darüber hinaus kann hier der RTIO-Datenfluss zwischen HWC-Modul und Rest-Modell kontrolliert und manipuliert werden.

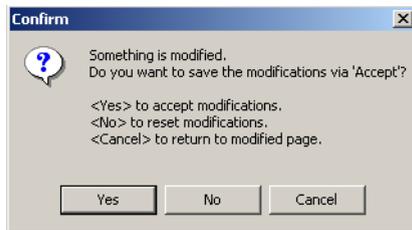
Genauere Informationen bezüglich der Registerkarten-Einstellungen finden Sie im Abschnitt „Konfigurationsregister“ auf Seite 111.

Darüber hinaus ist jedes Register noch mit den zwei Schaltflächen **Accept** und **Reset** ausgerüstet, die folgende Aufgabe haben.

Accept:

Sobald ein Register erstmalig oder erneut aufgebaut wird, wird von dem darin enthaltenen Datensatz eine Kopie erstellt, die dann in der Daten-Tabelle angezeigt wird. Jede Veränderung der Daten, die durch Editieren der gewünschten Zelle(n) erreicht wird, verändert nur die Kopie der Originaldaten.

Ist das Editieren beendet, so muss der Anwender durch Drücken der Schaltfläche **Accept** diese veränderten Daten auf die Originaldaten explizit zurückschreiben. Er wird dazu aufgefordert, falls er versehentlich vorzeitig das Register verlassen möchte:



Dieser Mechanismus ist notwendig, um unter allen Umständen einen konsistenten Datensatz zu gewährleisten, da zum Teil innerhalb der „Items“-Struktur vielfache Abhängigkeiten zu berücksichtigen sind.

Reset:

Diese Schaltfläche macht alle nach dem letzten Klicken von **Accept** vorgenommenen Änderungen rückgängig.

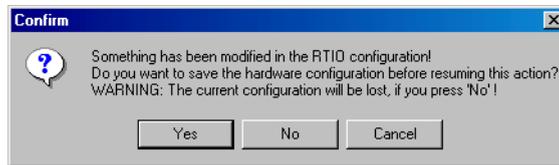
7.2.4 Hauptmenü

In diesem Abschnitt finden Sie eine Beschreibung der Einträge in den einzelnen Menüs.

Menü „File“

- **File → New**

Legt eine neue Hardware-Konfiguration an. Wurden zuvor an einer bestehenden Hardware-Konfiguration bereits Änderungen vorgenommen, die aber noch nicht gesichert wurden, so erscheint folgende Sicherheitsabfrage:

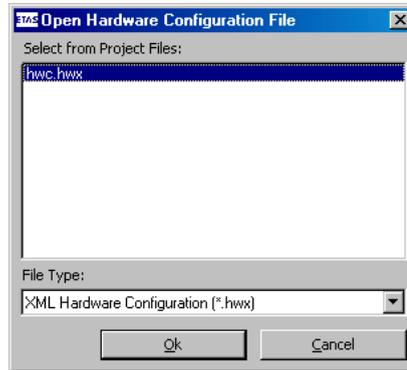


Nach Bestätigen mit **Yes** kann die Konfiguration im File-Container des Projekts gesichert werden. Anklicken von **No** löscht die erstellte Konfiguration und legt eine neue Hardwarekonfiguration an. **Cancel** bricht den Vorgang ab.

- **File → Open**

Über diese Menüfunktion wird das Fenster „Open Hardware Configuration File“ geöffnet. Im Kombikästchen „File Type“ wählen Sie das Format (* .hwx oder * .hwc); die Liste „Select from Project Files“ ent-

hält alle im File-Container des Projekts enthaltenen Dateien des gewählten Formats. Sie wählen hier die Datei, die mit **OK** in den HWC-Editor geladen wird.



Hinweis

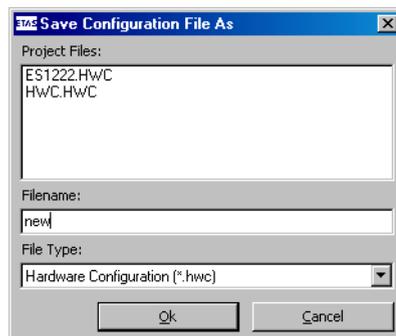
Der Datei `HWC.HWX` (oder `HWC.HWC`) kommt eine besondere Bedeutung zu: Diese Datei wird immer dann implizit angelegt bzw. überschrieben, wenn im HWC-Editor die Codegenerierung gestartet wird.

- **File → Save**

Sichert die erstellte Hardware-Konfiguration unter gleichem Namen in den File-Container im ASCET-Projekt.

- **File → Save As**

Sichert die erstellte Hardware-Konfiguration unter neuem Namen in den File-Container im ASCET-Projekt. Dazu wählen Sie im Fenster „Save Configuration File As“ das Format (* .hwx oder * .hwc) und geben in „Filename“ den neuen Dateinamen ein.



Die Liste „Project Files“ enthält die bereits vorhandenen Dateien des gewählten Formats. Wenn Sie eine davon überschreiben wollen, klicken Sie darauf. Der Dateiname wird in das Feld „Filename“ eingetragen. Wenn Sie auf **OK** klicken, wird die neue Hardware-Konfiguration unter dem existierenden Namen gesichert.

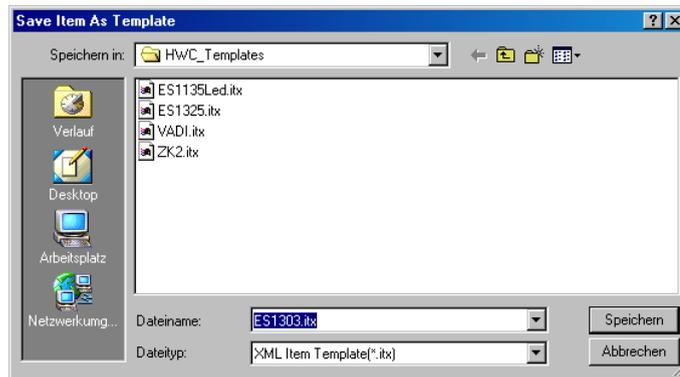
- **File → Save Item as Template**

Diese Funktion speichert die Einstellungen des gerade selektierten Items als Vorlage für alle Items dieses Typs (siehe hierzu auch **Use Item Templates** im Menü Extras → Options auf Seite 107).

Hinweis

*Die Hardwarekonfiguration wird standardmäßig im XML-Format *.itx abgelegt.
Das bisherige Format *.itp ist weiterhin gültig und kann als alternatives Format ausgewählt werden. Es wird jedoch empfohlen, das XML-Format zu verwenden, da das Format *.itp in zukünftigen ASCET-RP-Versionen nicht mehr unterstützt wird.*

Sie öffnet das Dialogfenster „Save Item As Template“, über das das Item als Vorlage in ein gewünschtes Verzeichnis kopiert werden kann. Sie wählen dazu im Kombikästchen „Dateityp“ das Format (*.itx oder *.itp) und stellen den gewünschten Pfad ein.



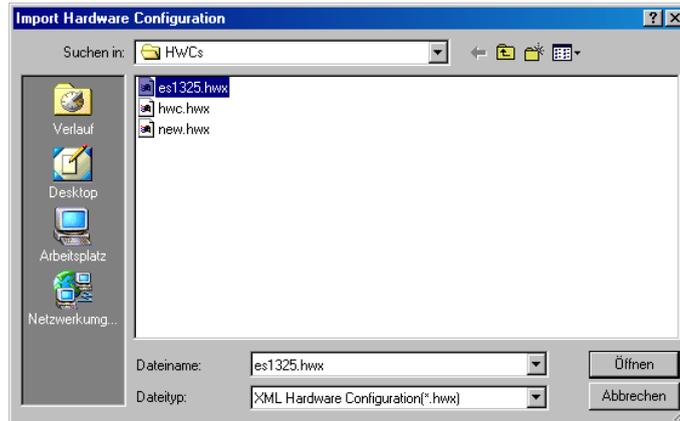
Hinweis

*Beim Öffnen des Dialogs wird immer der Dateipfad verwendet, der in den „HWC Editor Options“, Register „Templates“, unter „Template Path“ voreingestellt ist. Sie können jedoch einen beliebigen Pfad eingeben.
Der vorgeschlagene Dateiname darf jedoch **nicht** geändert werden.*

- **File → Import**

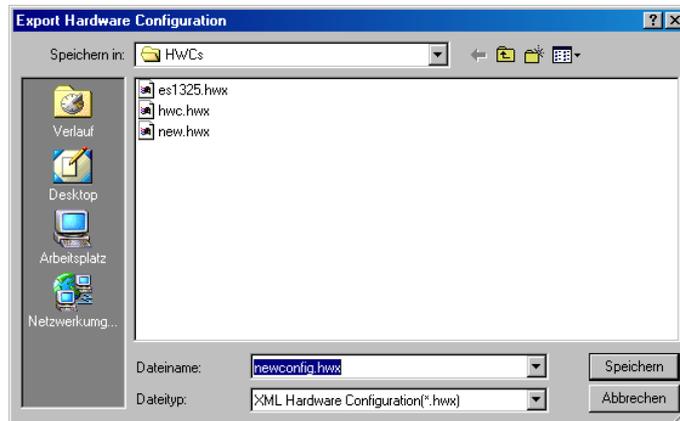
Über die Import-Schnittstelle kann eine Hardware-Konfiguration vom Dateisystem eingelesen werden. Sie wählen dazu im Fenster „Import Hardware Configuration“ den Dateityp (* .hwx oder * .hwc), den geeigneten Pfad und schließlich die Datei, die Sie importieren wollen.

Bestehende Hardware-Konfigurationen können eingelesen werden, wenn sie mit ASCET TIP Exp V4.0.0 oder später erzeugt wurden.



- **File → Export**

Über die Export-Schnittstelle kann eine Hardware-Konfiguration unter Angabe eines neuen Dateinamens und eines Formats auf das Dateisystem exportiert werden. Es stehen drei Dateiformate zur Auswahl.



Die Formate *.hwx und *.hwc werden für den Austausch und das Archivieren von Hardware-Konfigurationen verwendet, während das Format *.csv (kommagetrennte Werte) Dokumentationszwecken dient und in Kalkulationsprogrammen angezeigt werden kann.

Hinweis

*Die Hardwarekonfiguration wird standardmäßig im XML-Format *.hwx abgelegt.*

*Das bisherige Format *.hwc ist weiterhin gültig und kann als alternatives Format ausgewählt werden. Es wird jedoch empfohlen, das XML-Format zu verwenden, da das Format *.hwc in zukünftigen ASCET-RP-Versionen nicht mehr unterstützt wird.*

- **File → Exit**

Diese Menüfunktion schließt den HWC-Editor. Wird dabei festgestellt, dass eine geänderte Hardware-Konfiguration noch nicht gesichert wurde, besteht noch die Möglichkeit diese zu sichern, bevor der Editor geschlossen wird.

Menü „Edit“

- **Edit → Copy → Item(s)** (<CTRL> + <C>)

Erstellt eine Kopie vom selektierten Item mitsamt dem ggf. vorhandenen Teilbaum und überträgt sie in einen internen Puffer des HWC-Editors.

Der Puffer ist verfügbar, solange der HWC-Editor geöffnet ist.

- **Edit → Copy → Item(s) For Export**

Funktioniert wie **Copy → Item(s)**, nur wird hier die Kopie in eine externe Datei (*.hsx oder *.hws) geschrieben. Diese Funktion eignet sich dazu, Teil-Konfigurationen auszutauschen.

Hinweis

*Der Teilbaum wird standardmäßig im XML-Format *.hsx abgelegt. Das bisherige Format *.hws ist weiterhin gültig und kann als alternatives Format ausgewählt werden. Es wird jedoch empfohlen, das XML-Format zu verwenden, da das Format *.hws in zukünftigen ASCET-RP-Versionen nicht mehr unterstützt wird.*

- **Edit → Copy → Item Data**

Kopiert nur die Daten eines Items in einen internen Puffer (diese Funktion bezieht sich **nicht** auf ganze Teilbäume). Diese Funktion kann dazu verwendet werden, die Einstellungen zwischen zwei Items auszutauschen.

- **Edit → Paste → To Selected Item** (<CTRL> + <V>)

Fügt ein über **Copy → Item(s)** kopiertes Item (ggf. mit Teilbaum) unter das gerade selektierte Item ein. Folgenden Voraussetzungen sind notwendig, damit die Funktion gewährleistet ist.

1. Die Item-Typen müssen übereinstimmen; davon ausgenommen ist die Möglichkeit, Item-Teilbäume zwischen Systemkontroller-Elementen auszutauschen.
2. Es muss möglich sein, das Item bzw. den Teilbaum auch unter das selektierte Item einzufügen (hier gilt das Gleiche wie bei **Add Hardware Item**)

- **Edit → Paste → To Selected Item From Import**

Funktioniert gleich wie **Paste → To Selected Item**, nur erfolgt hier das Einlesen des kopierten Teils von der gewünschten Datei. Es gelten die selben Beschränkungen wie beim vorigen Menüpunkt.

- **Edit → Paste → Data to selected Item**

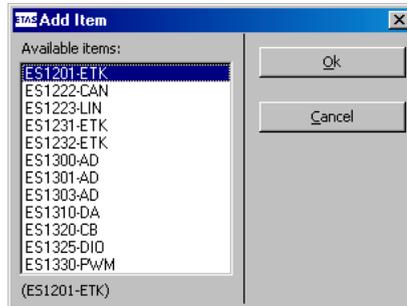
Überschreibt die Daten des vorhandenen Items mit denen, die über **Copy → Item Data** in den Puffer geschrieben wurden.

Im Gegensatz zu den vorhergehenden **Paste...**-Menüpunkten gibt es hier auch die Möglichkeit, Daten zwischen ähnlichen bzw. kompatiblen Items auszutauschen. Diese Funktion ist dann möglich, wenn der Typ des selektierten Items gleich oder kompatibel zu dem Item ist, von dem die Daten kopiert wurden. Darüber hinaus muss auch die Struktur nahezu identisch sein; d.h. die Anzahl der Reihen in den Tabellen in den Registern „Groups“, „Signals“ und „Mappings“ muss übereinstimmen.

- **Edit → Add Item** (<INS>)

Über diese Menüfunktion kann ein neues Hardware-Item in die Hardware-Konfiguration eingefügt werden.

Ein Auswahlfenster öffnet sich, das alle verfügbaren Items der nächst tieferliegenden Hierarchiestufe anzeigt.



Das gewählte Element wird dann in die Hardware-Konfiguration entsprechend eingefügt.

Hinweis

Abhängig von der bereits erstellten Konfiguration kann auch nach erfolgreicher Auswahl eines Items das Hinzufügen mit einer Fehlermeldung abgebrochen werden. Dies ist z. B. dann der Fall, wenn die Anzahl zur Verfügung stehender Ressourcen (Ports, Steckplätze) bereits aufgebraucht ist oder wenn eine bestimmte Einfügereihenfolge notwendig ist.

- **Edit → Delete Item** ()

Über diese Menüfunktion kann das markierte Element aus der Hardware-Konfiguration gelöscht werden.

Das Entfernen des obersten Items der Hierarchie vom Typ „Hardware Interface Manager“ ist nicht möglich.

- **Edit → Move Item Up** (<CTRL> + <U>)

Dient zum Ändern der Item-Reihenfolge innerhalb einer Hierarchiestufe in der Hardware-Konfiguration. Die Menüfunktion bewegt das ausgewählte Item um ein Element nach oben.

- **Edit → Move Item Down** (<CTRL> + <D>)

Dient zum Ändern der Item-Reihenfolge innerhalb einer Hierarchiestufe in der Hardware-Konfiguration. Die Menüfunktion bewegt das ausgewählte Item um ein Element nach unten.

- **Edit → Add Row Before**

Diese Menüfunktion ist nur für Items des Typs „Device“ verfügbar, sofern sie die Möglichkeit bieten, ihre Signalgruppen- bzw. Signal-Anzahl dynamisch zu ändern.

Hinweis

Systembedingt ist die Größe der Tabelle oft beschränkt. In diesem Fall sind die Menüfunktionen zum Hinzufügen und Löschen von Tabellenzeilen deaktiviert.

Unter folgenden Voraussetzungen ist die Menüfunktion verfügbar und wählbar:

1. Item vom Typ „Device“ ausgewählt, das über „dynamische“ Eigenschaften verfügt (z.B. CAN-IO Device);
2. Register „Groups“ oder „Signals“ ist ausgewählt;
3. Ein Eintrag der Spalte „No“ ist ausgewählt (1. Spalte der Tabelle).

Diese Menüfunktion fügt eine Zeile vor der selektierten Zeile bzw. dem selektierten Bereich ein.

Hinweis

*Da es sich hier um eine strukturelle Änderung der Tabelle handelt, lassen sich Änderungen über **Edit → Add *** und **Edit → Delete *** **nicht** mehr durch die Schaltfläche **Reset** wieder rückgängig machen.*

Beispiel:

Im Register „Signals“ soll vor dem gewählten Zellenbereich (No 3-4) eine Zeile eingefügt werden; d.h. die Zeile 3 (Signal „Signal6“) entsteht neu.

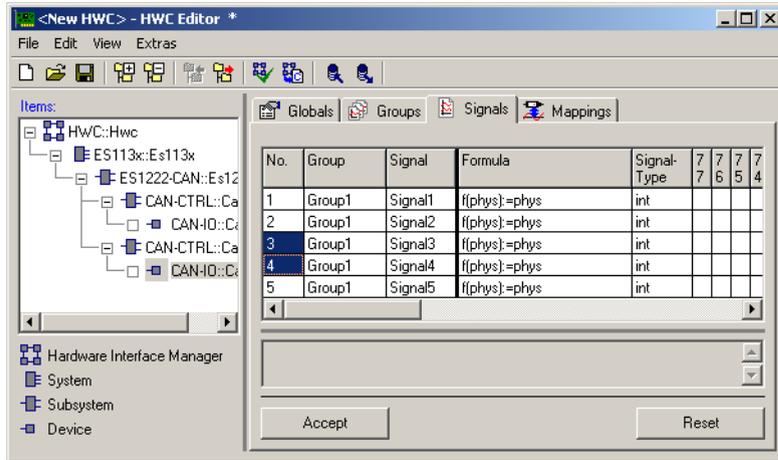


Abb. 7-1 HWC-Editor mit markiertem Bereich, vor dem eine Zeile eingefügt werden soll

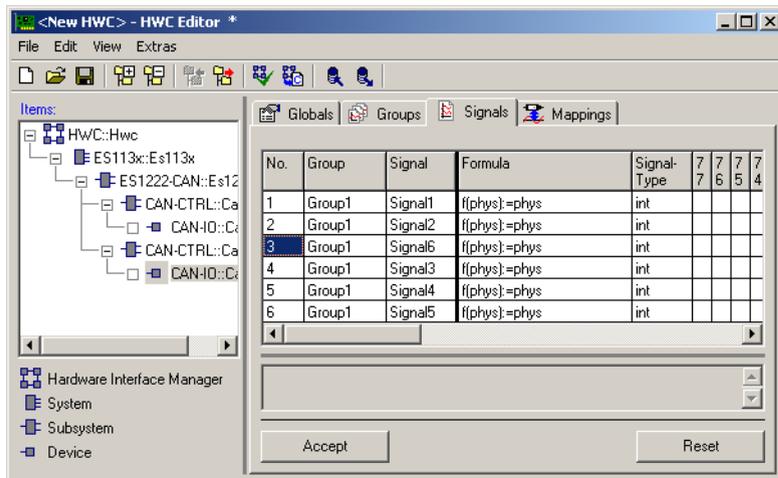


Abb. 7-2 HWC-Editor nach Einfügen der neuen Zeile (Signal signal6, markiert)

- **Edit → Add Row After**

Funktioniert wie **Add Row Before**, nur wird hier eine neue Zeile *nach* der selektierten Zelle bzw. dem selektierten Bereich eingefügt.

- **Edit → Add Multiple Rows Before**

Funktioniert wie **Add Row Before**, nur können hier gleich mehrere Zeilen eingefügt werden. Die Anzahl der neu einzufügenden Zeilen kann in einem Dialogfenster eingegeben werden:



- **Edit → Add Multiple Rows After**

Funktioniert wie unter **Add Multiple Rows Before** beschrieben, nur werden hier neue Zeilen *nach* der selektierten Zelle, bzw. dem selektierten Bereich eingefügt.

- **Edit → Delete Row(s)**

Löscht die Zeile bzw. den Bereich, der in der Spalte „No“ markiert ist.

Hinweis

Aus Systemgründen muss mindestens eine Signalgruppe bzw. ein Signal in der Tabelle verbleiben. Beim Versuch, die letzte Reihe zu löschen, erscheint eine Fehlermeldung.

Menü „View“

- **View → Expand all**

Diese Menüfunktion klappt die Baumstruktur im Feld „Items“ vollständig auf.

- **View → Collapse all**

Diese Menüfunktion klappt die Baumstruktur im Feld „Items“ vollständig zu.

- **View → Show All**

Diese Menüfunktion blendet alle darstellbaren Optionen/Spalten ein.

Die Tabelle einer Registerkarte kann so gestaltet sein, dass einige der möglichen Optionen bzw. Spalten standardmäßig nicht eingeblendet sind. Dies hat den Vorteil, dass wenig benutzte oder selten gebrauchte Optionen ausgeblendet und so die Tabelle übersichtlicher gestaltet werden kann.

Hinweis

*Wird anschließend die Tabelle mit der Schaltfläche **Accept** bestätigt und die Hardware-Konfiguration gesichert, steht nach dem erneuten Laden die Tabelle wieder so zur Verfügung, wie sie gesichert wurde, d.h. neu eingeblendete Optionen/Spalten bleiben eingeblendet.*

- **View → Hide All Hidable**

Bildet das Gegenstück zur vorigen Menüfunktion, d.h. alle ausblendbaren Optionen/Spalten werden ausgeblendet.

- **View → Hide Selected**

Blendet die aktuell gewählte Option bzw. Spalte aus, sofern sie ausblendbar ist.

Menü „Extras“

- **Extras → Check Item**

Diese Funktion ermöglicht eine explizite Überprüfung aller Einstellungen für das ausgewählte Item einerseits und der Überprüfung gegenüber dem aktuellen ASCET Projekt andererseits. Diese Überprüfung erfolgt auch implizit vor jeder RTIO-Codegenerierung.

Erst wenn diese Überprüfung ohne Fehler abgeschlossen werden konnte, ist eine RTIO-Codegenerierung überhaupt möglich. Warnungen sind zulässig, sollten aber im Einzelfall immer beachtet werden.

- **Extras → Check Hardware Configuration**

Dieser Menüpunkt ermöglicht eine explizite Überprüfung aller Einstellungen hinsichtlich Konsistenz innerhalb der Hardware-Konfiguration einerseits und der Überprüfung gegenüber dem aktuellen ASCET Projekt andererseits. Diese Überprüfung erfolgt auch implizit vor jeder RTIO-Codegenerierung.

Erst wenn diese Überprüfung ohne Fehler abgeschlossen werden konnte, ist eine RTIO-Codegenerierung überhaupt möglich. Warnungen sind zulässig, sollten aber im Einzelfall immer beachtet werden.

- **Extras → Generate Code → For Current Experiment**

Dieser Menüpunkt startet die RTIO-Codegenerierung für das aktuell im ASCET Projekteditor gewählte Experiment. Im Einzelnen werden folgende Schritte durchgeführt:

1. Der Inhalt des gesamten HWC-Moduls wird gelöscht (damit werden alle vom HWC-Modul generierten Prozesse aus der Task-Liste ausgetragen).
2. Alle benötigten Items werden im HWC-Modul angelegt (Messages, Parameter, Variablen...).
3. Der C-Code für den Header und für die benötigten Prozesse des HWC-Moduls wird generiert.
4. Alle Prozesse des HWC-Moduls werden in die entsprechenden Tasks einsortiert.

- **Extras → Generate Code → For Phys. and Quant. Experiment**

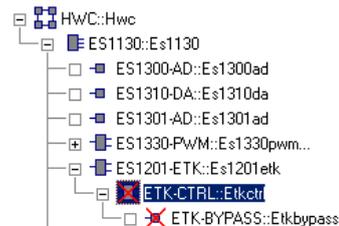
Die Funktion ist identisch zur vorher beschriebenen mit dem Unterschied, dass hier der Code für das Physikalische und Quantisierungs-Experiment im HWC-Modul angelegt wird.

- **Extras → Generate Code → For Phys., Quant. and Impl. Experiment**

Diese Funktion ist identisch zur vorher beschriebenen, nur wird hier Code für das Physikalische, Quantisierungs- und Implementierungs-Experiment generiert.

- **Extras → Item Code Generation On/Off**

Diese Funktion erlaubt es, für ein ausgewähltes Item (und die darunter hängende Struktur) die Codegenerierung an- oder abzuschalten. Standardmäßig ist die Codegenerierung eingeschaltet. Wenn Sie diese Funktion einmal wählen, wird die Codegenerierung abgeschaltet; das Symbol für das ausgewählte Item (und die darunter liegende Struktur) wird durchgestrichen.



Wenn Sie jetzt Code generieren, wird das markierte Item nicht berücksichtigt.

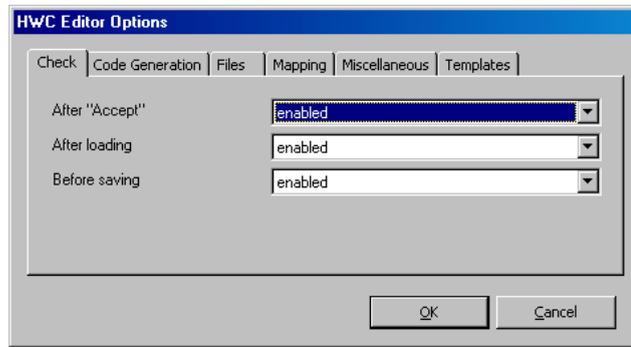
Wenn Sie erneut **Extras → Item Code Generation On/Off** wählen, wird die Codegenerierung wieder eingeschaltet, das Symbol erscheint nicht mehr durchgestrichen.

- **Extras → Clear HWC Module**

Löscht den gesamten Inhalt (Items und Prozesse) des HWC-Moduls.

- **Extras → Options**

Öffnet das Dialogfenster, in dem die Einstellungen des HWC-Editors verändert werden können.



Nachfolgend werden die Register und deren Optionen erläutert.

Register „Check“

After „Accept“: enabled / disabled

Aktiviert bzw. deaktiviert die automatische Überprüfung, die beim Drücken der Schaltfläche **Accept** ausgelöst wird. Aus Performancegründen wird hier nur die Datenkonsistenz innerhalb der gerade ausgewählten Registerkarte überprüft.

After loading: enabled / disabled

Aktiviert bzw. deaktiviert die implizite Überprüfung, die durchgeführt wird, wenn eine neue Hardware-Konfiguration geladen wird.

Before saving: enabled / disabled

Aktiviert bzw. deaktiviert die automatisch durchgeführte Überprüfung der Hardware-Konfiguration, bevor eine Konfiguration gesichert wird.

Register „Code Generation“

Automatic repair: no / yes

Aktiviert bzw. deaktiviert die Möglichkeit, einige festgestellte Fehler in der Hardware-Konfiguration automatisch zu beheben. Diese Möglichkeit wird aber zur Zeit nur sehr eingeschränkt unterstützt.

Log Level: silent / brief / verbose

Legt den Umfang des Protokolls fest, das bei der HWC-Codegenerierung bei Bedarf erzeugt werden kann.

Register „Files“:

Default storage format: XML (.hwx and *.itx) / Legacy (*.hwc and *.itp)*

Diese Option legt das Standardformat für Hardware-Konfigurationsdateien und Item-Templates fest.

Validate XML files: yes / no

Diese Option legt fest, ob der XML-Code in *.hwx- und *.itx-Dateien beim Einlesen auf syntaktische Richtigkeit geprüft wird. Wenn die Option eingeschaltet ist (yes), wird das Einlesen abgebrochen, wenn ein Fehler im XML-Code gefunden wird.

Verify XML Checksum: yes / no

Diese Option legt fest, ob beim Einlesen von *.hwx- und *.itx-Dateien deren Prüfsumme verifiziert wird. Mit der Prüfsummenverifikation wird geprüft, ob die fragliche Datei eine von ASCET-RP generierte XML-Datei ist, die nicht verändert wurde.

Wenn die Option eingeschaltet ist (yes), wird eine Warnung ausgegeben, wenn die Prüfsumme nicht verifiziert werden kann.

Register „Mapping“

Message Creation Target: Project / Module

Diese Option legt fest, ob beim Mapping generierte ASCET-Messages im Projekt selbst abgelegt werden (Auswahl `Project`, Standard) oder in einem der zum Projekt gehörenden Module.

Module Instance Name

Diese Option ist nur von Bedeutung, wenn unter „Message Creation Target“ `Module` gewählt ist.

Im Eingabefeld wird der Name des Moduls eingegeben, in dem die Messages angelegt werden sollen. Wenn das angegebene Modul nicht direkt im Projekt eingebunden ist, erscheint folgende Fehlermeldung:

```
Can't add sendReceiveMessage '<msg name>',  
because module '<module name>' doesn't exist in  
project, see option 'Module Instance Name' in  
tab 'Automatic Mapping'!
```

Register „Miscellaneous“

Data digits after decimal point: 1...6

Data digits before decimal point: 1...9

Die beiden Optionen legen das Format fest, in dem die Werte in der Spalte „Data“ in den „Mappings“-Registerkarten der Devices angezeigt werden.

Darstellung: <Data digits before...>.<Data digits after...>

Register „Templates“

Template Path

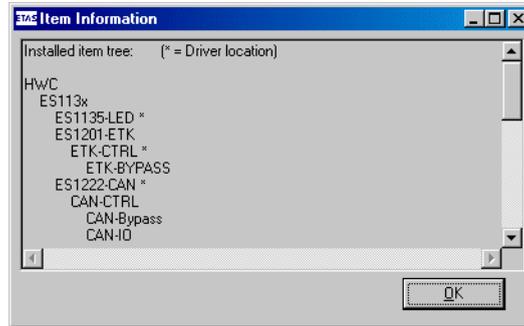
Hier kann das Verzeichnis gewählt werden, das als Ablage für die Item-Templates verwendet wird.

Use Item Templates: yes / no

Aktiviert bzw. deaktiviert die Verwendung von Item-Templates. Item-Templates dienen dazu, die vom System vorgegebenen Default-Werte mit eigenen Einstellungen zu überschreiben. Dazu werden bei aktivierter Template-Option jedesmal beim Einfügen von Items (Menü **Edit** → **Add Item**) die vom System vorgegebenen Standardwerte durch diejenigen überschrieben, die in dem Template-Verzeichnis gesichert wurden (Menü **File** → **Save Item as Template**).

- **Extras → Installed Items**

Dieser Menüpunkt listet alle aktuell vom System unterstützten Items sowie deren hierarchische Abhängigkeiten auf.

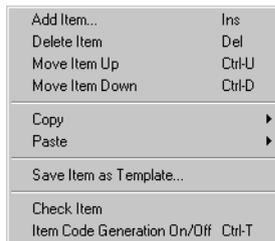


Hinweis

*Bei den mit einem * markierten Items erfolgt eine tatsächliche Ankopplung an die entsprechenden Low-Level-Hardware-Treiber über API-Funktionen.*

7.2.5 Kontextmenü in der Liste „Items“

In der Liste „Items“ steht Ihnen das nachstehend abgebildete Kontextmenü zur Verfügung.



Die Funktionen des Kontextmenüs sind auch über das Hauptmenü verfügbar. Sie sind in den vorhergehenden Kapiteln beschrieben.

7.3 Konfigurationsregister

In den Konfigurationsregistern können alle Einstellungen der Items vorgenommen werden. Das Register „Globals“ steht für alle Item-Typen zur Verfügung, die Register „Groups“, „Signals“ und „Mappings“ gibt es nur für Items vom Typ „Device“.

7.3.1 Allgemeine Hinweise

Option / Zelle editieren

In der Tabelle eines Registers kann durch Anklicken mit der Maus die betreffende Zelle ausgewählt und editiert werden. Die Navigation innerhalb der Tabelle kann auch mit den Pfeiltasten erfolgen.

Manche Spalten sind auch multi-selektionsfähig, d.h. es können gleichzeitig mehrere Zellen innerhalb der Spalte selektiert werden (z.B. „Formula“-Spalte).

Multiselektionsmöglichkeiten zusammenhängender Felder einer Spalte:

- Mit Klicken und gleichzeitigem „Ziehen“ der Maus über die gewünschten Felder der Spalte

oder

- mit Klicken in das erste und in das letzte gewünschte Feld bei gleichzeitig gedrückter <Shift>-Taste werden die gewünschten Felder markiert.

Mit dem Drücken von <F2> oder einem weiteren Mausklick bei gedrückter <SHIFT>-Taste (geht *nur* mit Kombikästchen oder Eingabefeldern) wird der Editiermodus erreicht.

- Eine Mehrfachselektion ist auch über die Tastatur mit gleichzeitig gedrückter <Shift>-Taste möglich.

Mit dem Drücken von <F2> wird der Editiermodus erreicht.

Multiselektionsmöglichkeiten nicht zusammenhängender Felder einer Spalte:

- Mit Klicken der Maus bei gleichzeitig gedrückter <STRG>-Taste werden die gewünschten Felder markiert. Das zuletzt zu selektierende Feld muss bei gleichzeitig gedrückter <STRG>-Taste mit einem Doppelklick markiert werden.

Gleichzeitig wird der Editiermodus erreicht.

- Mit linken Mausklicks bei gleichzeitig gedrückter <STRG>-Taste werden alle gewünschten Felder markiert.

Mit dem Drücken von <F2> wird der Editiermodus erreicht.

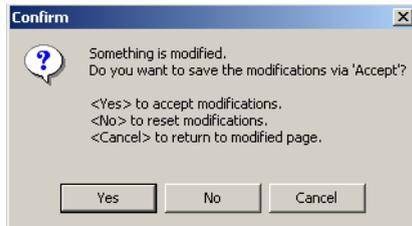
Die Bestätigung der Eingaben erfolgt mit <ENTER>, mit <ESC> wird der Editiermodus abgebrochen.

Editierbarkeit

Nicht alle Optionen können vom Anwender verändert werden. Manche Optionen sind ständig gesperrt und dienen nur zur Anzeige von bestimmten Statuswerten. Andere Optionen können zeitweise gesperrt oder eingabebereit sein, was unter Umständen von anderen Optionseinstellungen in diesem oder einem anderen Item in der Items-Hierarchie abhängig sein kann.

Modifiziert-Zustand

Der HWC-Editor registriert, ob ein Wert der Tabelle bereits geändert wurde oder sich gerade im Editions-Zustand befindet. Wird in diesem Fall ein anderes Item bzw. Register angewählt, ohne dass die Änderung zuvor über die Schaltfläche **Accept** gültig gemacht wurde, kommt es zu folgender Meldung:



Sichern Sie die Änderung mit **OK** oder brechen Sie den Wechsel mit **Abbrechen** ab.

Scrollen der Tabelle

Die Tabelle kann unter Umständen zu groß für die Darstellung in einem Register sein. In diesem Fall werden „Scrollbars“ eingeblendet.

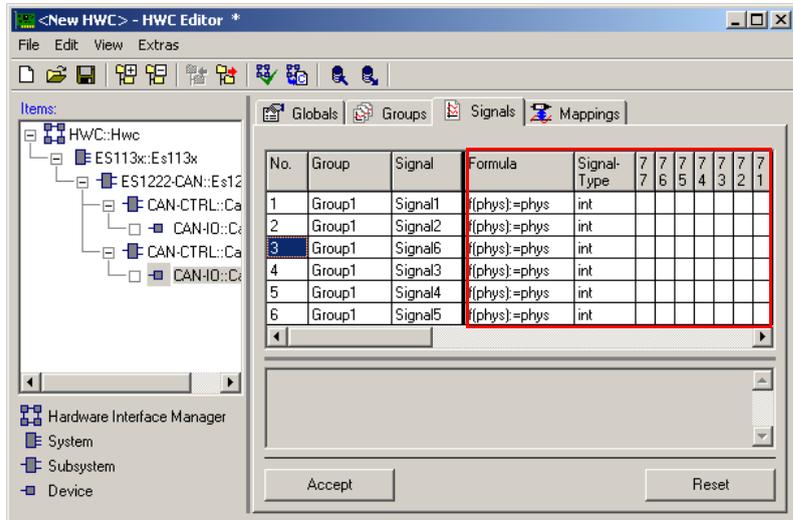


Abb. 7-3 Scrollbarer Bereich eines Konfigurationsregisters

Mit ihnen wird dann aber **nicht** die gesamte Tabelle verschoben, sondern nur der variable Teil (markierter Bereich in Abb. 7-3); der übrige Teil bleibt unverändert.

Hinweis

Unter Umständen kann es vorkommen, dass im linken (statischen) Teil der Tabelle so viele Spalten eingeblendet sind, dass der rechte scrollbare Teil nicht mehr in das Register passt, so dass der Anwender vermeintlich keine Reaktion auf ein Scrollen sieht. Dieses Problem kann entweder durch ein Vergrößern des HWC-Editors oder durch ein Ausblenden nicht benötigter statischer Spalten behoben werden.

Spaltenbreite ändern

Die Spaltenbreite kann durch linken Mausklick auf die Spaltenbegrenzung mit anschließendem Festhalten nach Belieben verändert werden. Die neu festgelegten Spaltenbreiten werden dann mit der Hardware-Konfiguration mit gesichert, wenn sie zuvor über die Schaltfläche **Accept** bestätigt wurden.

Sie werden dann nach erneutem Laden wieder wie gewünscht eingestellt.

Hilfe-Text

Im unteren Teil des Registers ist ein Textfeld angebracht, das die Möglichkeit bietet, einen kurzen Hilfetext für die gerade ausgewählte Option anzuzeigen.

7.3.2 Standardoptionen im Register „Globals“

In diesem Abschnitt werden die im Register „Globals“ geführten Standard-Optionen beschrieben. Weitere Item-spezifische Optionen finden Sie in den entsprechenden Abschnitten in Kapitel 10 „HWC-Items“.

Name

Hier kann eine individueller Name für das Item vorgegeben werden. Der Name muss ein gültiger ANSI-C kompatibler Name sein und im gesamten Hardware-Baum eindeutig sein.

Diese Option ist in jedem Register „Globals“ vorhanden.

Init Task

Hier muss die Task ausgewählt werden, in der der Hardware-Treiber initialisiert werden soll. Das ist in der Regel eine „echte“ Init-Task. In Ausnahmefällen kann es auch eine „Software“-Task sein, das ist jedoch ungewöhnlich und zieht bei der Überprüfung eine Warnung nach sich (siehe hierzu auch Kapitel „Vorbereitende Maßnahmen“ auf Seite 71).

Diese Option ist immer dann vorhanden, wenn das Item mit einem Low-Level-Treiber in Verbindung steht.

Hinweis

In jedem Fall muss bei der Abarbeitung der Tasks sichergestellt sein, dass diese Task erstmalig ausgeführt wird, bevor irgendeine andere Task, die mit diesem Treiber in Verbindung steht (Start, Config...), ausgeführt werden kann. Desweiteren muss darauf geachtet werden, dass der „Init“-Aufruf erst wieder erfolgen darf, wenn zuvor ein „Exit“-Aufruf erfolgt ist. Falls nicht, so kommt es zu einem Laufzeitfehler im Experiment was eine „Sperrung“ des Treibers nach sich zieht (Driver Lock)!

Eine in sich konsistente Task-Reihenfolge muss in jedem Fall eingehalten werden:

1. Init Task
2. Start Task
3. Stop Task
4. Exit Task

Exit Task

Hier muss die Task ausgewählt werden, in der der Hardware-Treiber deinitialisiert werden soll. Das ist in der Regel eine „echte“ Exit-Task (= inactive Init-Task), es könnte aber auch eine „Software“-Task sein, was aber eher ungewöhnlich ist und bei der Überprüfung eine Warnung nach sich zieht (siehe hierzu auch Kapitel „Vorbereitende Maßnahmen“ auf Seite 71).

Eine „Exit Task“ ist immer dann vorhanden, wenn das Item mit einem Low-Level-Treiber in Verbindung steht.

Hinweis

Task-Reihenfolge unbedingt beachten (siehe „Init Task“ auf Seite 114)!

Config Task

Die Auswahl einer Config Task zur Änderung bestimmter Konfigurationsparameter zur Laufzeit ist nur aus Gründen der Kompatibilität mit anderen ETAS-Produkten (LabCar) enthalten. Für die Verwendung von ASCET-RP ist diese Einstellung ohne Bedeutung.

IRQ Handler Task

Die hier ausgewählte Task muss immer eine Software-Task sein, d.h. sie muss im ASCET-Projekteditor in der Task-Liste als „Software“-Task angelegt werden. Bei „Max. No. of Activations“ ist mindestens die Anzahl 2 (maximal 50) einzutragen.

Eine „IRQ Handler Task“ kann von Treibern benötigt werden, die im Interrupt-Betrieb arbeiten.

Hinweis

*Diese Task muss **ausschließlich** für das Element bzw. den betreffenden Hardware-Treiber zur Verfügung stehen und darf von keinem anderen Element bzw. von Anwender-Prozessen mitbenutzt werden. Auch zwei gleichartige Elemente (z.B. zwei ES1222) dürfen sich **nicht** die gleiche Task teilen!*

Device Manager Task

Diese Task wird von einigen Treibern benötigt, um unabhängig vom eigentlichen Datenaustausch eine gewisse Grundversorgung des Treibers sicherzustellen (z.B. Fehlerbehandlung, Bus-Überwachung).

Hier muss in der Regel eine Timer-Task vorgegeben werden. Die Zykluszeit ist stark unterschiedlich und wird im Hilfetext des Registers bzw. in der entsprechenden Item-Dokumentation genauer spezifiziert.

Version

Anzeige der Versionsbezeichnung des Items, welche zur Versions-Überprüfung mit dem Low-Level-Hardware-Treibern herangezogen werden kann.

„Version“ ist standardmäßig ausgeblendet.

Format

Anzeige des Formats des Items, welches zur Kompatibilitäts-Überprüfung mit dem Low-Level-Hardware-Treibern herangezogen werden kann.

„Format“ ist standardmäßig ausgeblendet.

7.3.3 Standardoptionen im Register „Groups“

In diesem Abschnitt werden die im Register „Groups“ geführten Standard-Optionen beschrieben. Weitere Item-spezifische Optionen finden Sie in den entsprechenden Abschnitten in Kapitel 10 „HWC-Items“.

No

Ist für die Nummerierung der Zeilen zuständig, damit z.B. Fehlermeldungen zugeordnet werden können.

Device

Hier wird der zugehörige Item-Name angezeigt.

„Device“ ist standardmäßig ausgeblendet.

Group

Hier wird die Bezeichnung der Signalgruppe angegeben. Dieser Name ist in der Regel editierbar, wobei der Name aber in diesem Fall ein gültiger ANSI-C Name sein muss.

Hinweis

Alle Signalgruppen innerhalb eines Devices müssen unterschiedliche Namen besitzen!

Direction

Gibt die Übertragungsrichtung der betreffenden Signalgruppe an. Falls die Einstellung gesperrt ist, so ist die Übertragungsrichtung hardwaremäßig fest vorgegeben und kann nicht geändert werden (z.B. AD-Karte immer „receive“, DA-Karte immer „send“). Bei einigen Devices (z.B. CAN-IO) ist die Übertragungsrichtung frei wählbar.

Task

Hier wird die Task angegeben, in der die Datenübertragung erfolgen soll. Auch die Vorgabe mehrerer Tasks ist möglich - dann wird in jeder der Tasks die Datenübertragung durchgeführt. Hier sind in der Regel Timer-Tasks oder Soft-

ware-Tasks zulässig. Bei manchen Devices ist hier auch keine Angabe möglich; dann erfolgt die Datenübertragung auf andere Weise, in der Regel interrupt-gesteuert.

Hinweis

Wenn mehrere Tasks für dieselbe Signalgruppe ausgewählt werden, kann es während der Ausführungszeit zu Konsistenzproblemen kommen, weil dann der zugehörige RTIO-Prozess mehreren Tasks zugeordnet wird.

Die vom HWC-Editor generierten RTIO-Prozesse sind jedoch nicht wiedereintrittsfähig (non-reentrant). Deshalb muss bei der Erstellung der OS-Konfiguration darauf geachtet werden, dass es nicht zu Überschneidungen bei der Ausführung des RTIO-Prozesses kommt.

Dies kann am einfachsten dadurch erreicht werden, dass man bei der Zuordnung von mehreren Tasks zu derselben Signalgruppe ausschließlich kooperative Tasks verwendet. Sollen trotzdem preemptive Tasks verwendet werden, liegt es in der Verantwortung des Anwenders, eine korrekte OS-Konfiguration zu gewährleisten.

7.3.4 Standardoptionen im Register „Signals“

In diesem Abschnitt werden die im Register „Signals“ geführten Standard-Optionen beschrieben. Weitere Item-spezifische Optionen finden Sie in den entsprechenden Abschnitten in Kapitel 10 „HWC-Items“.

No

Ist für die Nummerierung der Zeilen zuständig, damit z.B. Fehlermeldungen zugeordnet werden können.

Device

Hier wird der zugehörige Elementname angezeigt.

„Device“ ist standardmäßig ausgeblendet.

Group

Hier wird der zugehörige Signalgruppen-Name angezeigt. Bei manchen Devices wie z.B. „CAN-IO“ ist hier die Zuordnung des Signals zu einer Signalgruppe auch frei wählbar.

„Group“ ist standardmäßig ausgeblendet.

Direction

Hier wird die Übertragungsrichtung der entsprechenden Signalgruppe (und damit auch des Signals) angezeigt.

„Direction“ ist standardmäßig ausgeblendet.

Task

Hier wird die zugeordnete Task angezeigt, in der die betreffende Signalgruppe übertragen wird.

„Task“ ist standardmäßig ausgeblendet.

Signal

Hier kann ein Signal-Name festgelegt werden. Bei einigen Devices ist die Bezeichnung der Signale auch fest und kann vom Anwender nicht verändert werden. Falls eine Eingabe möglich ist, so muss der Name den ANSI-C Richtlinien entsprechen.

Hinweis

Alle Namen von Signalen, die einer Signalgruppe zugeordnet sind, müssen unterschiedlich sein!

Formula

In dieser Spalte kann bei Bedarf ein Signal-Wert mit einer linearen Umrechnung versehen werden. Standardmäßig ist hier eine 1:1-Umrechnung gewählt (Formula $f(\text{phys}) := \text{phys}$). Bei einigen Devices ist diese Formel fest vorgegeben und die Spalte gesperrt. Dies ist z.B. beim ETK-BYPASS Device der Fall, bei dem das Signal aus einer ASAM-MCD-2MC-Information generiert wird und die Formel daher fest vorgegeben ist.

Diese Formel wird in der Signalaufbereitung verwendet. Die Signalaufbereitung umfasst drei Schichten: Umsetzung, Quantisierung und Mapping. Eine vom RTIO-Treiber empfangene Signalgruppe wird direkt an die Signalaufbereitung weitergereicht.

In der *Umsetzungsschicht* wird die Signalgruppe in die einzelnen I/O-Signale zerlegt und mit Hilfe der Formel im Register „Signals“ in die physikalische Darstellung der korrespondierenden ASCET Message umgerechnet.

Durch die *Quantisierungsschicht* werden diese physikalischen Signale für quantisierte oder Implementierungsexperimente an die Quantisierung der ASCET Message angepasst. Dabei wird die in der ASCET Implementierung für die entsprechende gemappte Message festgelegte Formel verwendet.

In der *Mapping-Schicht* erfolgt letztendlich die Umkopierung der quantisierten Signale auf die zugehörige ASCET Messages.

7.3.5 Standardoptionen im Register „Mappings“

In diesem Abschnitt werden die im Register „Mappings“ geführten Standard-Optionen beschrieben. Weitere Item-spezifische Optionen finden Sie in den entsprechenden Abschnitten in Kapitel 10 „HWC-Items“.

No

Ist für die Nummerierung der Zeilen zuständig, damit z.B. Fehlermeldungen zugeordnet werden können.

Device

Hier wird der zugehörige Item-Name angezeigt.
„Device“ ist standardmäßig ausgeblendet.

Group

Hier wird der zugehörige Signalgruppen-Name angezeigt.
„Group“ ist standardmäßig ausgeblendet.

Direction

Hier wird die Übertragungsrichtung der entsprechenden Signalgruppe (und damit auch des Signals) angezeigt.
„Direction“ ist standardmäßig ausgeblendet.

Task

Hier wird die zugeordnete Task angezeigt, in der die betreffende Signalgruppe übertragen wird.
„Task“ ist standardmäßig ausgeblendet.

Signal

Hier wird der Signalname angezeigt.

ASCET Message

Hier kann bei Bedarf eine Zuordnung zu einer ASCET-Message vorgenommen werden („Mapping“). In einem Auswahldialog erhält der Anwender alle im Projekt vorhandenen EXPORTED Messages, die passend zum aktuellen Signal sind. Das sind für ein „receive“-Signal alle exportierten Receive-Messages und für ein „send“-Signal alle exportierten Send-Messages.

Im Dialogfenster „Message selection“ kann durch Eingabe einer Zeichenfolge nach der gewünschten Message gesucht werden, wobei die Zeichenfolge unten im Dialogfenster eingeblendet wird.

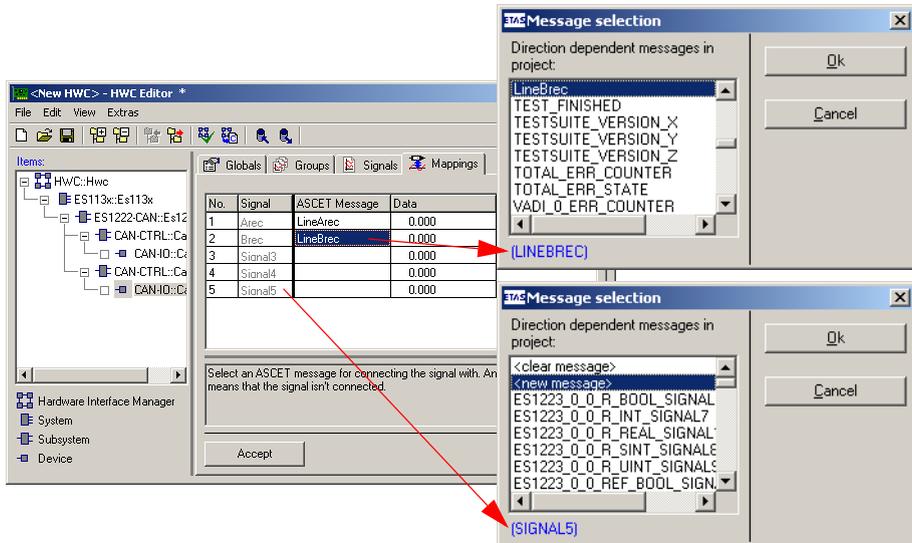


Abb. 7-4 Dialogfenster „Message selection“

Beim Öffnen des Auswahlfensters wird so vorgegangen (siehe Abb. 7-4):

- Ist eine Message in der Spalte „ASCET Message“ eingetragen, so wird die Message als Suchzeichenfolge verwendet
- Ist keine Message in der Spalte „ASCET Message“ eingetragen, so wird der zugehörige Signalname als Suchzeichenfolge verwendet

Eine eingetragene Message in der Spalte „ASCET Message“ kann im Dialogfenster „Message selection“ durch `<clear message>` gelöscht werden.

Eine neue Message kann durch Auswahl von `<new message>` erzeugt werden. Hierfür öffnet sich folgendes Dialogfenster:



In das Dialogfenster wird der neue Message-Name eingegeben, unter dem die Message als globale Send&Receive-Message im Projekt eingetragen wird.

Data

In dieser Spalte werden die Ersatzwerte für Send-Messages eingegeben. Die RTIO-Codegenerierung setzt die ASCET Parameter auf die angegebenen Werte.

Explanation

Diese Spalte enthält Erläuterungen zu den Signalen. Sie ist nicht bei allen Items vorhanden.

8 Codegenerierung

Die RTIO-Codegenerierung kann erst dann gestartet werden, wenn die implizite Überprüfung der Hardware-Konfiguration fehlerlos war (Warnungen sind zulässig, sollten aber in jedem Fall beachtet werden).

Die Codegenerierung löscht zunächst den gesamten Inhalt des HWC-Moduls (Elemente, Code und Prozesse), bevor der Inhalt neu generiert wird.

Hinweis

Der Anwender sollte unter keinen Umständen den generierten Code von Hand modifizieren, da sonst eine ordnungsgemäße Funktion nicht mehr gewährleistet werden kann. Im Extremfall kann sogar die Hardware zerstört werden!

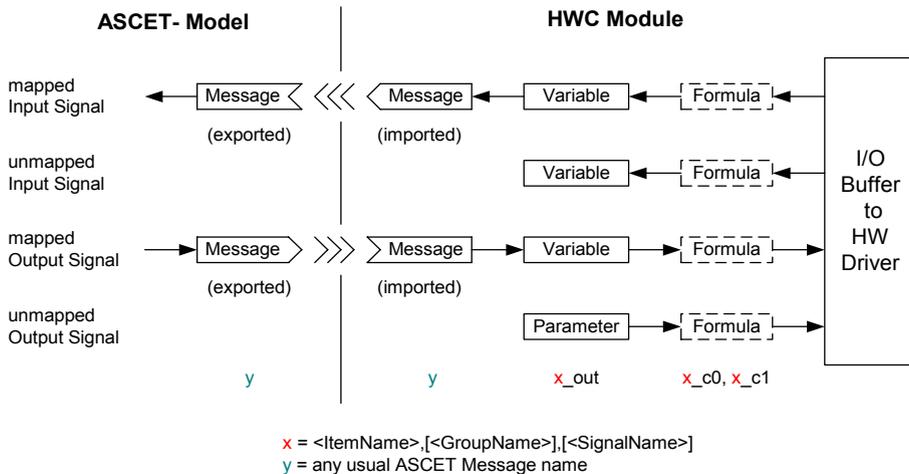
8.1 HWC-Modul

In diesem Abschnitt soll kurz ein Überblick über den generierten Code gegeben werden. Dies soll dem versierten Anwender ein Gefühl von der Arbeitsweise der RTIO-Ankopplung vermitteln und die Fehlersuche erleichtern.

Für den einfachen Anwender ist dieses Kapitel aber nicht von Belang.

8.1.1 Elemente

Das nachfolgende Schaubild soll einen Überblick über den Datenfluss zwischen Projekt und HWC-Modul geben:



Messages

Für jeden Eintrag, der in der Spalte „ASCET Message“ (Register „Mappings“) im HWC-Editor eingetragen ist, wird ein gleichlautendes Message-Element angelegt, das immer „Imported“ ist (mit „y“ bezeichnet).

I/O-Schnittstelle

Jedes Signal, das mit dem Hardware-Treiber kommuniziert, wird über eine I/O-Schnittstelle innerhalb des HWC-Modules geführt (mit x_{out} bezeichnet). Je nachdem, ob es sich dabei um ein „send“ oder „receive“-Signal handelt, bzw. ob das Signal auf eine Message abgebildet ist, wird entweder eine Variable oder ein Parameter mit dem Namen x_{out} angelegt.

Formula

Ist für das Signal eine lineare Umrechnungsformel gewählt, so gibt es für die Koeffizienten c_0 und c_1 noch die entsprechenden Parameter.

Verstellparameter

Für jeden Konfigurations-Wert wird noch ein zusätzlicher Verstellparameter angelegt. Die Bezeichnung des Parameters setzt sich dabei wieder aus „Item“- und / oder „Group“- Prefix und einer Parameterbezeichnung zusammen, welche aber individuell für jedes Item ist.

Externer Code

Im externen Code des HWC-Moduls wird der Code für eine Interrupt-Vektor-Tabelle und für die benötigten Interrupt-Vektoren generiert.

Header Code

Im Header-Code werden die Datenpuffer für den Datenaustausch mit den Low-Level-Treibern und die Datenstrukturen zur Initialisierung der Treiber definiert.

Prozesse

Die Prozesse, die vom RTIO-Framework generiert werden, gliedern sich in Prozesse für die Treiber-Steuerung und Prozesse für den Datenaustausch.

Es kann folgende Prozesse für die Treiber-Steuerung geben:

- `<Item-Name>_InitCode_<Task-Name>_HWCF`
- `<Item-Name>_StartCode_<Task-Name>_HWCF`
- `<Item-Name>_ConfigCode_<Task-Name>_HWCF`
- `<Item-Name>_DeviceManagerCode_<Task-Name>_HWCF`
- `<Item-Name>_AcknowledgeCode_<Task-Name>_HWCF`
- `<Item-Name>_AnalyzeCode_<Task-Name>_HWCF`
- `<Item-Name>_StopCode_<Task-Name>_HWCL`
- `<Item-Name>_ExitCode_<Task-Name>_HWCL`

Für jede definierte Signalgruppe wird ein eigener Prozess für den Datenaustausch mit dem Hardware-Treiber generiert.

Es kann folgende Prozesse für den Datenaustausch geben:

- `<Item-Name>, <Signalgruppen-Name>_<TaskName>_HWCF`
(„receive“-Prozess)
- `<Item-Name>, <Signalgruppen-Name>_<TaskName>_HWCL`
(„send“-Prozess)

HWCF bedeutet, dass der Prozess in die mit `<Task-Name>` angegebene Task **vor** die vorhandenen Prozesse eingefügt werden soll, HWCL bedeutet, dass der Prozess **nach** den vorhandenen Prozessen eingefügt werden soll.

8.2 Prozessreihenfolge

Das RTIO-Framework sorgt mit der automatischen Einsortierung der Prozesse in die Task-Liste für einen in sich stimmigen Ablauf der Hardware-Kommunikation.

Hinweis

Diese Zuordnung wird bei jeder RTIO-Codegenerierung erneut nach dem festgelegten Schema durchgeführt. Der Anwender sollte unter keinen Umständen die Zuordnung der „HWC“-Prozesse ändern. Dies kann zu unvorhersagbaren Resultaten führen, was auch eine Beschädigung der Hardware nach sich ziehen kann!

Würden alle vom RTIO-Framework generierten Prozesse einer Task zugeordnet, so würde sich folgende Prozessreihenfolge ergeben:

- Init
- Analyze
- Acknowledge
- DeviceManager
- Start
- Config
- erste "receive"-Signalgruppe
- ...
- letzte "receive"-Signalgruppe
- ANWENDER-PROZESSE
- erste "send"-Signalgruppe
- ...
- letzte "send"-Signalgruppe
- *Stop
- *Exit

Die Prozesse werden für die „Items“ gemäß der Reihenfolge, die durch die hierarchische Struktur vorgegeben ist von oben nach unten abgearbeitet, d.h. der Systemcontroller ES113x muss zuerst initialisiert werden, bevor ein untergeordneter Hardware-Treiber initialisiert wird.

Für die mit "*" markierten Prozesse ("Stop" und "Exit") gilt die umgekehrte Reihenfolge; d.h. es muss zuerst ein Hardware-Treiber beendet werden, bevor der Systemcontroller beendet wird.

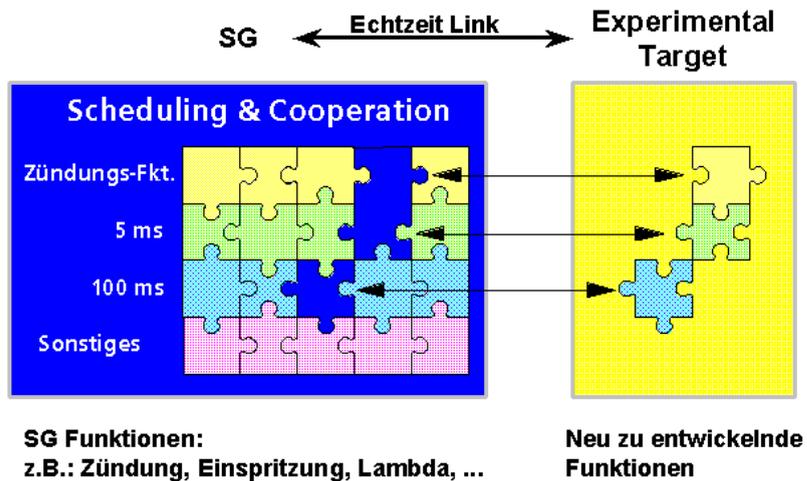
9 Der ETK-BYPASS (ES1200/ES1201/ES1231/ES1232)

In diesem Abschnitt wird das RTIO-Package für ETK-Bypass beschrieben. Dabei wird vorausgesetzt, dass der Anwender die nötigen Techniken in ASCET beherrscht, auf die Bedienung von ASCET wird nicht eingegangen.

9.1 Definition des Begriffs ETK-Bypass

Bei einem ETK-Bypass werden bestimmte Funktionen des Steuergeräts (SG) auf den Simulationsrechner, d.h. den PowerPC-Rechenknoten des ETAS Experimentalsystems, ausgelagert. Dabei werden Daten aus dem SG zur Experimental-Hardware übertragen. Auf Grundlage dieser Daten werden einzelne Funktionen aus dem SG in der Experimental-Hardware gerechnet. So können die ausgelagerten Daten auf einfache Weise modifiziert und getestet werden. Die Ergebnisse werden zum SG zurück übertragen.

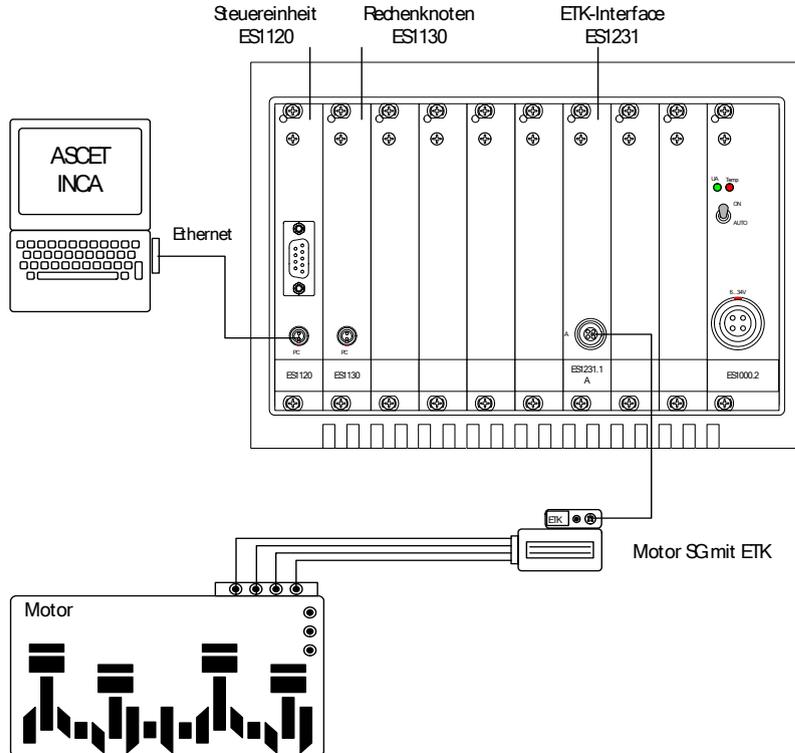
Welche Funktionen ausgelagert werden, wird durch Modifikationen in der Steuergeräte-Software, den sogenannten Freischnitt, bestimmt.



Die Verbindung zum Steuergerät erfolgt über einen ETK (Emulator-Tastkopf). Die Kommunikation zwischen dem SG und dem ETK erfolgt über ein Dual Ported RAM.

9.2 Hardwarekonfiguration eines ETK-Bypass

Die folgende Abbildung zeigt eine Beispielkonfiguration für eine ETK-Bypass Anwendung mit dem ES1000.2 System:



Bei einer ETK-Bypass Anwendung werden die ausgelagerten Funktionen in ASCET spezifiziert, aus der Spezifikation wird dann Code generiert, der auf dem PPC-Modul des ETAS Experimentalsystems lauffähig ist. Der generierte Code wird vom Host-PC in das ETAS Experimentalsystem heruntergeladen. Das ETAS Experimentalsystem ist mit dem ETK des Steuergeräts über das ETK-Interface ES1200, ES1201, ES1231 oder ES1232 verbunden.

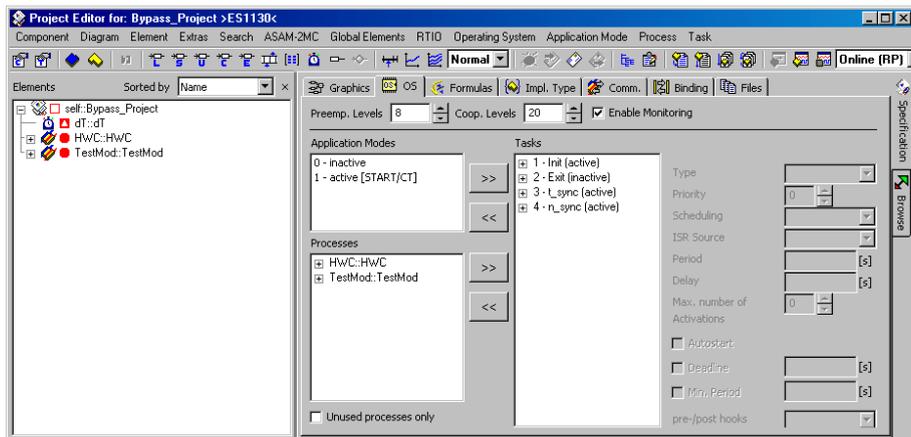
Zusätzlich können im Steuergeräteprogramm Schnittstellenparameter verstellt werden, die für die Konfiguration der Softwareschnittstelle zwischen dem Simulationsrechner und dem Steuergerät notwendig sind. Ferner können neue Softwarestände in das Steuergerät heruntergeladen werden. Auch Mess- und Verstellaufgaben im Steuergerät können während des Bypassbetriebs durchgeführt werden.

9.3 ASCET Projekt für den ETK-Bypass

In jedem ASCET RTIO-Projekt für ETK-Bypass müssen die folgenden Besonderheiten berücksichtigt werden:

- Das HWC-Modul (siehe Kapitel 5.2.1) muss zum ASCET RTIO-Projekt hinzugefügt werden. Der Name der Instanz muss `HWC` lauten.
- Die Task-Liste im OS-Editor muss die Tasks `Init` (Type: Init / Application Mode: active) und `Exit` (Type: Init / Application Mode: inactive) enthalten. Die Tasknamen „Init“ und „Exit“ sind Standardbezeichnungen, die vom RTIO-Package genutzt werden. Bei anderen Tasknamen muss die Task-Zuordnung jeder neuen Komponente manuell im HWC-Editor erfolgen.
- Die Task-Liste im OS-Editor benötigt zwei spezielle, individuelle Tasks. Diese Tasks werden vom RTIO-Package für die Zuordnung der Signalgruppen, in denen die vom Treiber empfangenen Bypass-Daten bereitgestellt werden, bzw. in denen die Bypass-Daten zum Treiber übertragen werden, benötigt. Diese Tasks müssen in jedem Fall den Typ *Software* besitzen. Die Task-Zuordnung muss manuell im HWC-Editor erfolgen.
- Alle Messages, auch Receive-Messages, die für die RTIO-Kommunikation zur Verfügung stehen sollen, müssen als „Exported“ deklariert werden.
- Globale Messages stehen für die RTIO-Kommunikation zur Verfügung.

Das nachfolgende Schaubild zeigt die fertige Konfiguration im Register „OS“:



Die nachfolgende Tabelle zeigt die Task-Einstellungen:

Taskname	Application Mode	Type	Prio.	Scheduling	Max. No of Act.	Period
Init	active	Init	-	-	-	-
Exit	inactive	Init	-	-	-	-
t_sync	active	Software	16	preemptive	1	-
n_sync	active	Software	17	preemptive	1	-

9.4 Funktionsweise des ETK-Bypass

Zur Durchführung eines Bypass-Projekts wird ein Steuergerät mit freigeschnittener, d.h. für den Bypass modifizierter Software, benötigt. Der Freischnitt ermöglicht es, zwischen dem Steuergerät und dem im Simulationsrechner ablaufenden Funktionen umzuschalten. Der Freischnitt enthält alle benötigten Informationen, um die Bypass-Eingangsdaten an den Simulationsrechner zu übermitteln, sowie die Bypass-Ausgangsdaten im Steuergerät weiterzuverarbeiten.

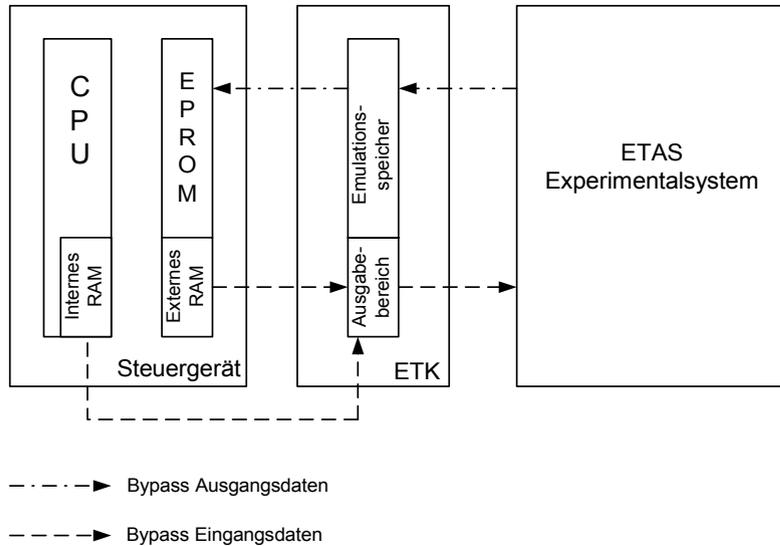
Die im Steuergerät freigeschnittenen Funktionen werden in der Regel auch bei aktiviertem Bypass gerechnet. Statt ihrer Ergebnisse werden aber die im Simulationsrechner ermittelten Ergebnisse, d.h. die Bypass-Ausgangsdaten, verwendet.

Die Bypass-Eingangsdaten sind die Daten, die aus dem Steuergerät ausgelesen werden. Aufgrund dieser Daten werden dann die ausgelagerten Funktionen gerechnet. Die Ergebnisse der Berechnungen werden als Bypass-Ausgangsdaten an das Steuergeräteprogramm zurückgegeben.

Der Speicher des ETK ist in einen Emulations Speicherbereich und einen Ausgabebereich aufgeteilt. Im Emulationsbereich steht das Steuergeräteprogramm, dieser Bereich ersetzt das ROM des Steuergeräts. Ferner werden hier die Bypass-Ausgangsdaten gespeichert, da das Steuergerät nicht direkt aus dem Ausgabebereich lesen kann.

Der Ausgabebereich wird nur für das Übermitteln der Bypass-Eingangsdaten vom Steuergerät an den Simulationsrechner verwendet. In diesem Bereich befindet sich eine Kopie des Steuergeräte-RAM. Je nach ETK-Modell und Steuergerät kann auch das CPU-interne RAM des Steuergeräts in diesem Bereich abgelegt sein.

Die folgende Abbildung veranschaulicht den Datenfluss innerhalb eines ETK-Bypass Projekts:



Der Datenfluss innerhalb des Bypass-Projekts läuft zyklisch ab, d.h. die Bypass-Eingangsdaten werden zuerst vom ETAS Experimentalsystem gelesen, und dann dort verarbeitet. Anschließend werden die in dem ETAS Experimentalsystem berechneten Bypass-Ausgangsdaten ins Steuergerät zurückschrieben und dort weiterverarbeitet. Dabei wird das Lesen der Bypass-Eingangsdaten durch ein zeit- oder winkelsynchrones Raster im Steuergerät synchronisiert. Das Zurückschreiben der Daten erfolgt unsynchronisiert.

9.5 Datenaustausch zwischen Steuergerät und ETAS Experimentalsystem

Für den Datenaustausch zwischen ETAS Experimentalsystem und Steuergerät kommt das DISTAB-Datenaustauschverfahren zum Einsatz.

Seit einigen Jahren wird im Zusammenspiel zwischen Steuergerät und ETK das DISTAB 12 Verfahren eingesetzt. DISTAB 12 unterstützt Messdaten bis zu zwei Byte Länge.

Vier Byte Integer-Werte oder vier bzw. acht Byte Real- oder Float-Werte machen den Einsatz von DISTAB 13 erforderlich. DISTAB 13 unterstützt die Erfassung von 1, 2, 4 und 8 Byte langen Messdaten aus dem Steuergerät unabhängig vom Format derselben (signed / unsigned Integer oder Float).

Für die Handhabung von DISTAB benötigt das ETK-Bypass Projekt verschiedene Parameter, die über eine vollständige und zum aktuellen Softwarestand des Steuergeräts passende ASAM-MCD-2MC-Beschreibungsdatei an ASCET übergeben werden können. Die DISTAB-Definition geschieht im Abschnitt DISTAB_CFG des TP_BLOBs:

```

/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

```

Hinweis

Die Kommentare sind in der ASAM-MCD-2MC-Datei üblicherweise nicht enthalten. Sie wurden hier lediglich der Übersicht halber eingefügt.

ASAM-MCD-2MC bezeichnet einen in der Automobilindustrie etablierten Standard zur Beschreibung eines Steuergeräte-Projektes (Verstellparameter, Messgrößen, Umrechnungsregeln, Adresslagen, etc.). Die ASAM-MCD-2MC-Datei muss für jeden neuen Programmstand neu erzeugt werden und sollte deshalb Ergebnis des Software-Entwicklungsprozesses sein. Detaillierte Kenntnisse des DISTAB-Verfahrens sind für die Einrichtung des ETK-Bypass nicht erforderlich, da alle nötigen Einstellungen Teil des Freischnitts sind. Dennoch wird an dieser Stelle kurz auf die Funktionsweise eingegangen, um das Untersuchen des ETK-Speichers zu Diagnosezwecken zu erleichtern.

Die Kommunikationsmechanismen für die Datenübergabe zwischen Steuergerät und Simulationsrechner werden als Bypass-Kanäle bezeichnet.

Bei ES1201 und ES1231 gibt es im Steuergerät zwei Bypass-Kanäle, von denen einer (Kanal A, höhere Priorität) im winkelsynchronen Raster läuft und einer (Kanal B, niedrigere Priorität) im zeitsynchronen Raster. Jeder der beiden Bypass-Kanäle ist durch verschiedene Adressinformationen und die Größe von

Datenpuffern definiert. Bei der ES1232 gibt es bis zu 32 Kanäle (je 16 Bypass- und Messkanäle), deren Namen und Prioritäten automatisch bestimmt werden (siehe Kapitel 10.8.7 auf Seite 213).

Bypass-Kommunikation (AML V1.1)

Die Bypass-Kommunikation wird hier anhand des Bypass-Kanals A der ES1231 erläutert. Der Ablauf ist für beide Kanäle gleich. Tab. 9-1 enthält die Namen und die Bedeutung der Parameter für beide Kanäle, die in der ASAM-MCD-2MC-Datei (*.a21, AML V1.1) für die Definition verwendet werden.

Hinweis

*Die Namen der Parameter sind nicht Teil des ASAM-MCD-2MC-Standards und nicht zwingend Bestandteil der *.a21-Datei. In den Beispielen dieses Handbuchs sind sie der besseren Übersicht halber manuell eingefügt.*

Parameter	Beschreibung
BYPASS_S	Startadresse der Pointerliste für die Bypass-Eingangsdaten (Bypass-Kanal A)
BYPASS_X	Startadresse der Pointerliste für die Bypass-Eingangsdaten (Bypass-Kanal B)
CHNL_S	Startadresse des Datenpuffers für die Bypass-Eingangsdaten (Bypass-Kanal A)
CHNL_X	Startadresse des Datenpuffers für die Bypass-Eingangsdaten (Bypass-Kanal B)
CHNL_T	Startadresse des Datenpuffers für die Bypass-Ausgangsdaten (Bypass-Kanal A)
CHNL_Y	Startadresse des Datenpuffers für die Bypass-Ausgangsdaten (Bypass-Kanal B)
TRGID_S	Trigger-ID-Adresse für die Bypass-Eingangsdaten (Bypass-Kanal A)
TRGID_X	Trigger-ID-Adresse für die Bypass-Eingangsdaten (Bypass-Kanal B)
BPMAX_S	Größe des Datenpuffers für die Bypass-Eingangsdaten in Byte (Bypass-Kanal A)
BPMAX_X	Größe des Datenpuffers für die Bypass-Eingangsdaten in Byte (Bypass-Kanal B)
BPMAX_T	Größe des Datenpuffers für die Bypass-Ausgangsdaten in Byte (Bypass-Kanal A)

Parameter	Beschreibung
BPMAX_Y	Größe des Datenpuffers für die Bypass-Ausgangsdaten in Byte (Bypass-Kanal B)
TRGSEG_A	Triggeradresse für Bypass-Kanal A
TRGSEG_B	Triggeradresse für Bypass-Kanal B

Tab. 9-1 Parameter für die Bypass-Kommunikation (AML V1.1)

In der ASAM-MCD-2MC-Datei sind die Parameter für Kanal A in einem QP_BLOB im Abschnitt IF_DATA ETK angeordnet:

```

/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

```

Abb. 9-1 zeigt schematisch den Ablauf eines Bypass-Zyklus mit dem DISTAB-Datenaustauschverfahren am Beispiel des Bypass-Kanals A. Die Zahlen repräsentieren die einzelnen Schritte des Bypass-Zyklus.

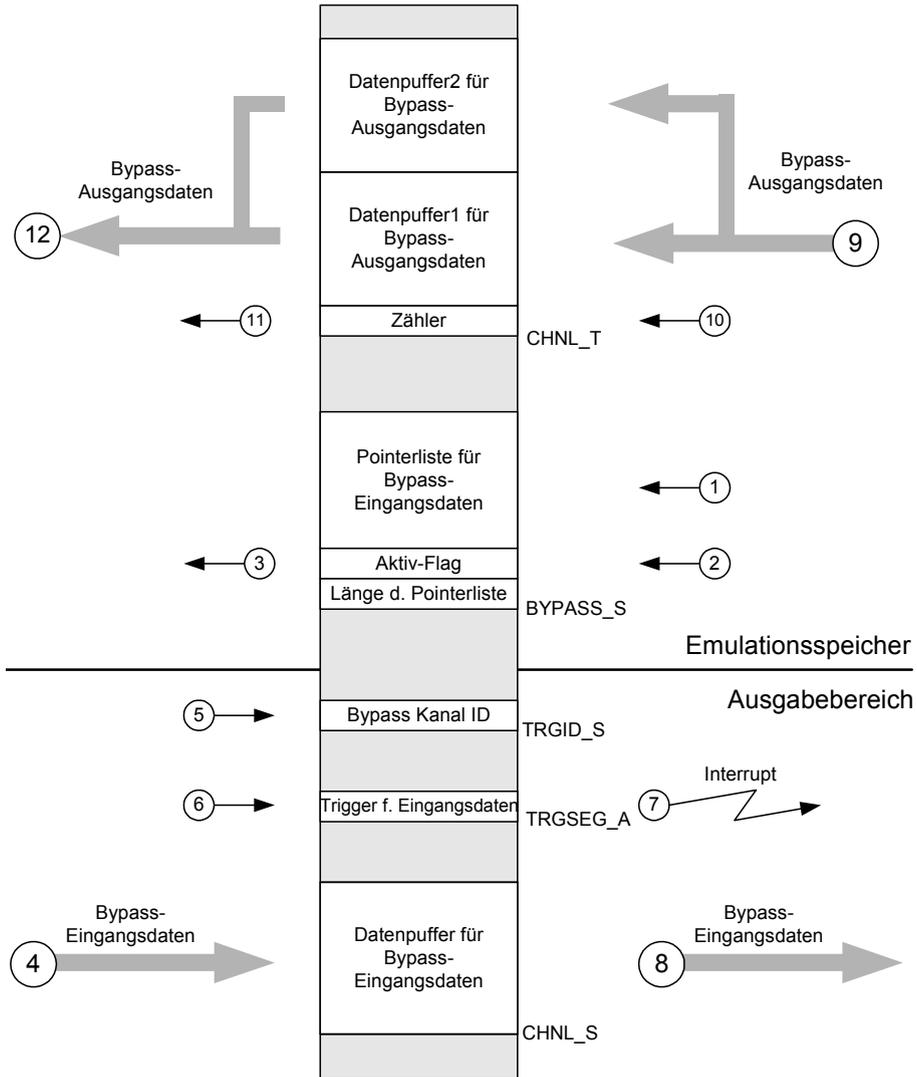


Abb. 9-1 schematischer Ablauf eines Bypass-Zyklus (DISTAB-Verfahren)

1. Beim Starten des Bypass-Experiments auf dem Simulationsrechner wird die Pointerliste für die Bypass-Eingangsdaten beschrieben.

DISTAB12	DISTAB13
Für jedes in den Simulationsrechner zu übertragende Byte gibt es einen Pointer, der auf die Adresse des Bytes im Steuergerätespeicher zeigt. Daten werden immer byteweise übertragen, d.h. zur Übertragung eines Words werden zwei Pointer benötigt. Nach Beschreiben der Pointerliste wird ihre Länge in das erste Byte der Liste eingetragen.	Die Bytes 4 bis 7 der Liste enthalten die Anzahlen der übertragenen 8 Byte-, 4 Byte-, 2 Byte- und 1 Byte-Signale. Die folgenden Bytes enthalten die Adressen der Signale, vom ersten 8 Byte-Signal bis zum letzten 1 Byte-Signal. Jede Adresse ist 4 Byte breit; in welcher Reihenfolge die Bytes einer Adresse geschrieben werden, hängt von der Einstellung für die Byte-Order (s. Seite 132) ab.

In diesem Schritt werden ebenfalls die Bypass-Offsets übertragen (s. Schritt 11, Seite 137).

2. Nachdem die Pointerliste beschrieben wurde, wird das Aktiv-Flag auf 1 gesetzt. Dadurch wird die Kommunikation zwischen Steuergerät und Simulationsrechner in Gang gesetzt. Schritt 1 und 2 werden nur einmal, zu Beginn eines Bypass-Experiments ausgeführt.

DISTAB12	DISTAB13
Aktiv-Flag: letztes Bit des zweiten Byte der Pointerliste	Aktiv-Flag: Bit 0 des ersten Byte (Byte 0) der Pointerliste

3. Das Steuergerät fragt den Flag-Byte der Pointerliste kontinuierlich ab. Sobald das Aktiv-Flag auf 1 gesetzt ist, wird der Bypass aktiviert, und die Übertragung der Daten beginnt.
4. Die Bypass-Eingangsdaten werden vom Steuergerät in den dafür vorgesehenen Datenpuffer, `CHNL_S`, geschrieben.

DISTAB12	DISTAB13
Das Wegschreiben geschieht byteweise, für jeden Eintrag in der Pointerliste wird das Byte an der entsprechenden Adresse in den Datenpuffer geschrieben. Die Übertragung wird beendet, wenn die Anzahl von Bytes übertragen wurde, die durch den Eintrag im ersten Byte der Liste festgelegt ist.	Die Daten werden signalweise in der Reihenfolge geschrieben, die in der Liste vorgegeben ist. Zunächst werden die Bytes des ersten 8 Byte-Signals geschrieben, wobei wieder die Einstellung für die Byte-Order bestimmt, welches Byte zuerst geschrieben wird. Die anderen Signale folgen.

5. Die ID des aktuellen Bypass-Kanals (für Bypass-Kanal A: 4) wird vom Steuergerät auf die Adresse `TRGID_S` geschrieben. Dies ist notwendig, weil nur zwei Adressen für die Kanal-IDs zur Verfügung stehen, es im Steuergerät aber insgesamt fünf Kanäle gibt. Davon werden zwei als Bypass-Kanäle benutzt, die anderen drei stehen für Applikationsaufgaben zur Verfügung. Die ES1232 hat bis zu 32 Kanäle, davon 16 Bypass- und 16 Messkanäle.
6. Die Triggeradresse `TRGSEG_A` wird vom Steuergerät mit einem beliebigen Wert beschrieben.
7. Durch den Schreibvorgang auf die Triggeradresse wird im Simulationsrechner ein Interrupt ausgelöst.
8. Durch Auslesen der Kanal-ID auf der Adresse `TRGID_S` stellt der Simulationsrechner fest, welcher Kanal angestoßen wurde, und liest den entsprechenden Datenpuffer aus. Anhand der Informationen aus der ASAM-MCD-2MC-Datei, z.B. der Umrechnungsformel, kann der Simulationsrechner die Rohdaten aus dem Steuergerät in physikalische Modellgrößen umrechnen, die dann vom ASCET Modell verarbeitet werden können.
9. Der Simulationsrechner rechnet die ins Bypass-Modell ausgelagerten Funktionen auf Basis der Bypass-Eingangsdaten. Anschließend werden die Ergebnisse als Bypass-Ausgangsdaten in einen der beiden dafür vorgesehenen Datenpuffer zurückgeschrieben. Für jeden Bypass-Kanal gibt es zwei Datenpuffer, in die abwechselnd geschrieben wird. Der Simulationsrechner inkrementiert zuerst intern (d.h. im Simulationsrechner) den Zähler der später an die Adresse `CHNL_T` geschrieben wird. Der Zähler wird erhöht, sobald der Interrupt von Schritt 7 erfolgt. Abhängig vom Zählerstand wird dann einer der Datenpuffer beschrieben. Ist der Zähler ungerade, wird der erste Datenpuffer beschrieben, ist er gerade, der zweite.
10. Der interne Zählerstand wird nach Beschreiben des Datenpuffers auf die Adresse `CHNL_T` geschrieben.
11. Der Wert des Zählers wird vom Steuergerät ausgelesen. Aufgrund des Werts entscheidet das Steuergerät, welcher Datenpuffer gelesen wird. Ist der Wert ungerade, wird der erste Datenpuffer gelesen, ist er gerade, der zweite. Auf diese Weise wird die Datenkonsistenz der Bypass-Ausgangsdaten sichergestellt.

Das Layout der beiden Puffer hängt von den gewählten Bypass-Ausgangssignalen ab. Dem SG muss daher mitgeteilt werden, wo im Puffer der Wert des Signals liegt. Dazu wird die Signalposition relativ zur Startadresse des Puffers – der *Bypass-Offset* – bestimmt. Für die Bypass-Signale gibt es aus Sicherheitsgründen keine Pointerliste, stattdessen wird

in der SG-Software pro Signal ein eigener Bypass-Offset-Parameter vorgehalten. In Schritt 1 (s. Seite 136) werden die Bypass-Offsets in die Bypass-Offset-Parameter eingetragen. Auf diese kann direkt zugegriffen werden, da sie sich in einem Applikationsdatenbereich des SG befinden, der im RAM des ETK allokiert ist.

12. Die Bypass-Ausgangsdaten werden vom Steuergerät ausgelesen. In der Regel ist im Steuergerät ein Sicherheitsmechanismus implementiert, der das Verhalten des Steuergeräts bei Ausfall der Bypass-Kommunikation festlegt. Details hierzu erfahren Sie vom Steuergeräteprogrammierer.

Für das Zurückschreiben der Bypass-Ausgangsdaten werden zwei Datenpuffer benötigt, da dieser Vorgang unsynchronisiert erfolgt. Die Ergebnisse der Berechnungen im Simulationsrechner werden zurückgeschrieben, wann immer diese Berechnungen abgeschlossen sind. Dadurch kann es vorkommen, dass die Ergebnisse vom Steuergerät abgefragt werden, bevor oder während die Daten zurückgeschrieben werden. Deshalb wird immer eine Kopie der Daten in einem Datenpuffer gehalten, bis das Zurückschreiben der neuen Daten beendet ist, um die Datenkonsistenz der Bypass-Ausgangsdaten sicherzustellen.

Andere AML-Versionen

Ein QP_BLOB der Version AML V1.1.1 unterscheidet sich nicht von dem obigen Beispiel aus AML V1.1 (s. Seite 134).

In AML V1.2 hat es einige Änderungen gegeben; einige Parameter sind verschwunden, dafür sind neue hinzugekommen. Die - optionalen - Namen sind ebenfalls anders. Der QP_BLOB sieht wie folgt aus:

```
/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 Adress  */
                /* 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 und AML V1.6 können ebenfalls verwendet werden. Die Änderungen gegenüber AML V1.2 sind für die Arbeit mit ASCET-RP V5.4 ohne Bedeutung.

Hinweis

AML V1.1.x beschreibt die ETK-Daten für 3+2 Messraster und ist gültig für ES1232 [ETK-CTRL-BAS], ES1231 und ES1200/01. **AML V1.2 und höher** beschreibt die ETK-Daten für Multiraster und ist gültig für ES1232 [ETK-CTRL-ADV].

9.6 Vorab benötigte Informationen und Daten

Bevor Sie beginnen können, ein ETK-Bypass Projekt zu erstellen, benötigen Sie einige Informationen und Daten über Ihr Steuergerät. Diese erhalten Sie in der Regel vom Steuergeräteprogrammierer, der den Freischnitt implementiert hat. Die in diesem Abschnitt aufgeführten Punkte können als Checkliste dienen, um einen reibungslosen Informationstransfer sicherzustellen.

Steuergeräteprogramm

Das freigeschnittene Steuergeräteprogramm muss im Steuergerät geladen sein.

ASAM-MCD-2MC-Datei

Die ASAM-MCD-2MC-Datei beschreibt u.a. die Datenstrukturen des Steuergeräts. Das ETK-Bypass Package benutzt die ASAM-MCD-2MC-Datei, um auf die Eingangs- und Ausgangsgrößen des Bypass-Projekts zuzugreifen. Eine zum aktuellen Softwarestand des Steuergeräts passende ASAM-MCD-2MC-Datei ist deshalb zwingend erforderlich. Welche AML-Version unterstützt wird, ist abhängig von der verwendeten Hardware; die Beschreibungen der verschiedenen Karten enthalten entsprechende Hinweise.

Datenformat des Steuergeräteprozessors

Verschiedene Prozessorfamilien verwenden verschiedene Word-Datenablageformate, die allgemein als *Big Endian* bzw. *Little Endian* bekannt sind. Beim Einrichten eines ETK-Bypass muss entweder bekannt sein, ob der Prozessor des Steuergeräts das *Little Endian* oder das *Big Endian* Ablageformat verwendet, oder das Word-Datenablageformat ist in der ASAM-MCD-2MC-Datei abgelegt und kann vom HWC-Editor ausgelesen werden.

Beim Format *Big Endian*, das z.B. von Motorola Prozessoren verwendet wird, steht das höherwertige Byte (*most significant byte*) zuerst. Beim Format *Little Endian* (z.B. bei Intel-Prozessoren) steht das niederwertige Byte (*least significant byte*) zuerst.

Basisadressen

Die Anfangsadressen der Übertragungskanäle und die Triggeradressen müssen dem ETK-Bypass Package angegeben werden oder die Anfangsadressen sind in der ASAM-MCD-2MC-Datei abgelegt und können vom HWC-Editor ausgelesen werden. Die Bedeutung der Basisadressen wird in Kapitel 9.5 „Datenaustausch zwischen Steuergerät und ETAS Experimentalsystem“ auf Seite 131 näher erläutert.

Parameter zur Aktivierung des Bypass

Im Steuergeräteprogramm muss der Bypass aktiviert sein. Wenn der Bypass aktiviert ist, werden statt der Funktionsergebnisse im Steuergeräteprogramm die entsprechenden Ergebnisse im Simulationsrechner verwendet. Die Aktivierung einer Bypass-Funktion erfolgt über das Verstellen eines Parameters mit INCA.

Bypass-Ausgangsgrößen

Nur bestimmte Variablen im Steuergerät können vom ETK-Bypass Projekt beschrieben werden. Diese sogenannten Bypass-Ausgangsgrößen werden vom Steuergeräteprogrammierer beim Freischnitt festgelegt. Die Bypass-Ausgangsgrößen müssen beim Anlegen des Projekts namentlich bekannt sein, oder die relevanten Informationen sind in der ASAM-MCD-2MC-Datei abgelegt und können vom HWC-Editor ausgelesen werden.

Bitmasken für die Transformation im NEAR-Bereich

Das DISTAB12-Verfahren unterstützt nur 16-Bit Speicheradressen für die Übertragung zwischen dem Steuergerät und dem Bypassrechner. Bei Steuergeräten, die Speicheradressen größer als 16 Bit verwenden, müssen diese daher mit Hilfe eines Datapage-Pointers auf 16-Bit umgesetzt werden. Hierzu dienen die Bitmasken `LongAdrANDMask` und `LongAdrORMask`. Die Bitmasken müssen im HWC-Editor eingetragen werden.

Verhalten im Fehlerfall / Sicherheitsfunktion

In der Regel ist im Steuergerät ein Sicherheitsmechanismus implementiert, der das Verhalten des Steuergeräts bei Ausfall der Bypass-Kommunikation definiert. Manchmal werden die Ergebnisse der Funktionen im Steuergerät verwendet, manchmal wird aber auch das Steuergerät in den *Reset-* oder *Notlaufmodus* geschaltet (abhängig vom Steuergerät). Besonders bei Bypass-Experimenten im Fahrzeug ist es daher wichtig, das Verhalten des Steuergeräts im Fehlerfall zu kennen.

10 HWC-Items

In diesem Kapitel werden sämtliche in ASCET verfügbaren HWC-Items beschrieben.

10.1 Implementierte Items

Um die implementierten Items des RTIO Packages anzuzeigen, wählen Sie im HWC-Editor [Options](#) → [Show Installed Items](#).

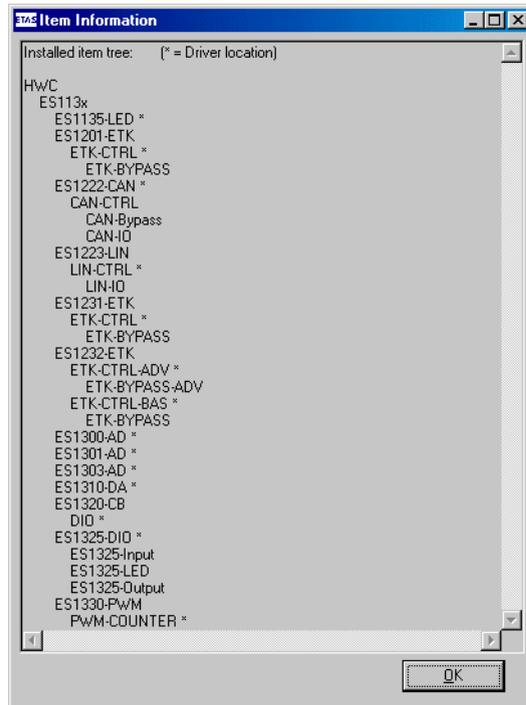


Abb. 10-1 „Item Information“ Dialog (* = Position des HW-Treibers)

In den nachfolgenden Abschnitten werden die Registerkarten des HWC-Editors für jedes Item gezeigt. Es werden nur die Item-spezifischen Einstellungen beschrieben. Einstellmöglichkeiten, die allen Items gemeinsam sind, werden im Kapitel „Konfigurationsregister“ auf Seite 111 beschrieben. Die Registerkarte „Mappings“ wird nicht gezeigt, da es keine Item-spezifischen Einstellungen in dieser Registerkarte gibt.

10.2 ES1135-LED

In diesem Kapitel werden die Einstellungen für die Steuerung der LEDs auf der Frontplatte der ES1135 beschrieben.

10.2.1 Globals (ES1135-LED Device)

In diesem Abschnitt werden die globalen Einstellungen des ES1135-LED Devices beschrieben.

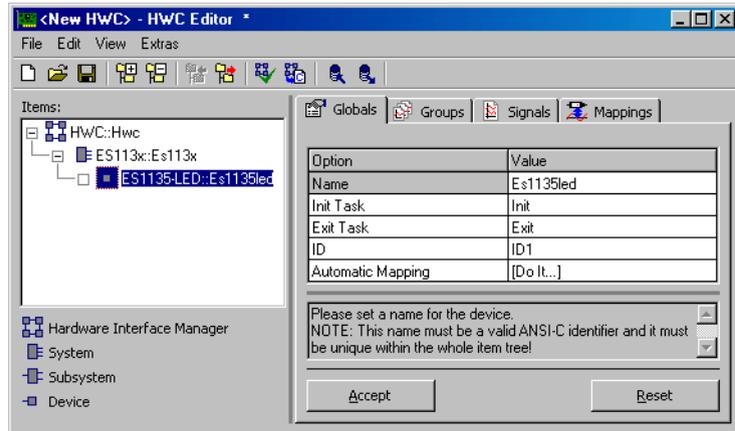


Abb. 10-2 Die Registerkarte „Globals“ des ES1135-LED Devices

Automatic Mapping

Diese Option ermöglicht die automatische Zuordnung zwischen Signalen und ASCET-Messages. Die Zuordnung erfolgt wie im Abschnitt „Automatic Mapping“ auf Seite 172 beschrieben.

10.2.2 Groups (ES1135-LED Device)

In diesem Abschnitt werden die signalgruppenspezifischen Einstellungen des ES1135-LED Devices beschrieben.

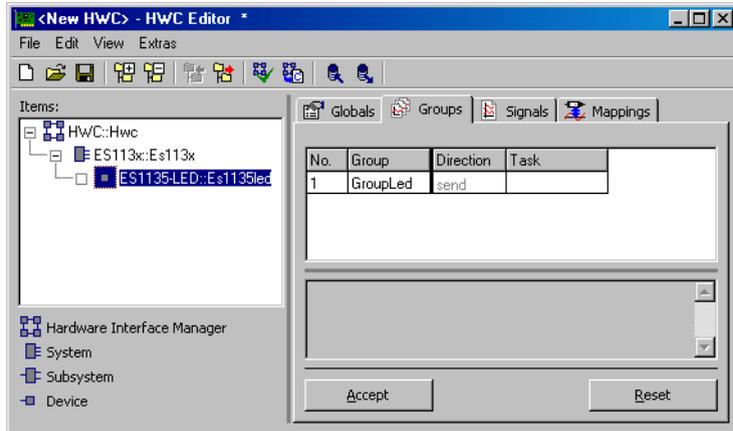


Abb. 10-3 Die Registerkarte „Groups“ des ES1135-LED Devices

Für das ES1135-LED Device sind in der Registerkarte „Groups“ keine item-spezifischen Spalten definiert. Die Einstellmöglichkeiten des Items werden in Kapitel 7.3.3 auf Seite 117 beschrieben.

10.2.3 Signals (ES1135-LED Device)

In diesem Abschnitt werden die signalspezifischen Einstellungen des ES1135-LED Devices beschrieben.

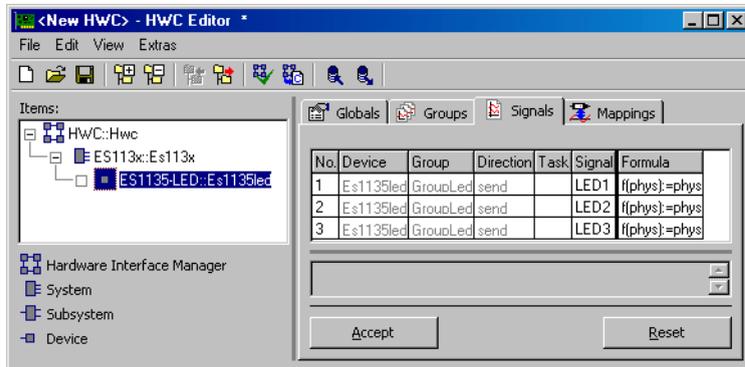


Abb. 10-4 Die Registerkarte „Signals“ des ES1135-LED Devices

Für das ES1135-LED Device sind in der Registerkarte „Signals“ keine Item-spezifischen Spalten definiert. Die Einstellmöglichkeiten der Items werden in Kapitel 7.3.4 auf Seite 118 beschrieben.

10.2.4 Mappings (ES1135-LED Device)

Für das ES1135-LED Device sind in der Registerkarte „Mappings“ keine Item-spezifischen Spalten definiert. Die Einstellmöglichkeiten der Items werden in Kapitel 7.3.5 auf Seite 120 beschrieben.

10.3 ES1201-ETK

Die ETK-Interface Karten ES1200 und ES1201 ermöglichen eine Ankopplung des Experimentalsystems an Steuergeräte mit ETK. Mittels störsicherer serieller Schnittstelle (8 MBit/s) kann der Inhalt des Steuergerätespeichers an das ETAS Experimentalsystem übergeben, bzw. von diesem verändert werden.

Die ES1200 stellt eine ETK-Schnittstelle zur Verfügung, während die ES1201 über zwei ETK-Schnittstellen verfügt. Ansonsten sind beide Karten identisch.

Hinweis

Die ES1201 Karte benötigt zur Kommunikation mit einem ETK zusätzlich noch spezielle Systemdienste, die nur von einer ES1111 (VCU) oder einer ES1120 (VCU2) Systemcontroller Karte bereitgestellt werden können (Siehe hierzu auch Abschnitt „Hardware – Experimentalsystem ES1000.x“ auf Seite 71).

10.3.1 Globals (ES1201-ETK Subsystem)

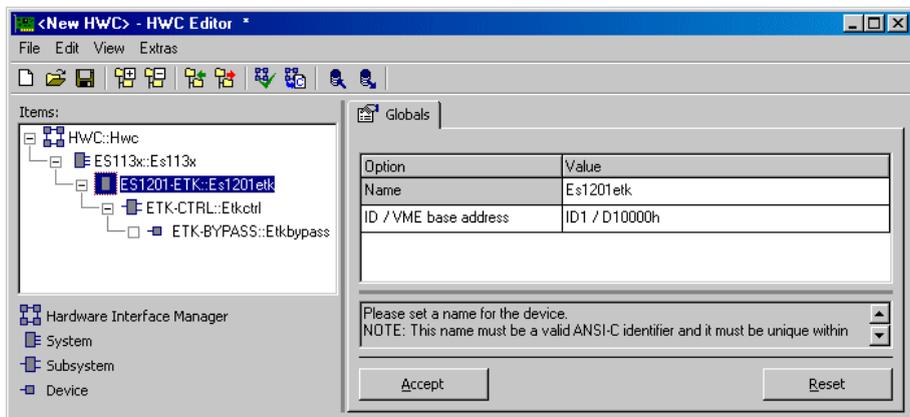


Abb. 10-5 Die Registerkarte „Globals“ des ES1201-ETK Subsystems

ID / VME base address

In dieser Zeile wird die Einstellung der VME-Basisadresse vorgenommen. Diese Einstellung muss mit den Schalterpositionen der ES1201 übereinstimmen. Für die ES1201 können zwei unterschiedliche VME-Basisadressen gewählt werden; dadurch können bis zu zwei ES1201 in einem ETAS Experimentalsystem betrieben werden.

Dem ES1201-ETK Subsystem können bis zu zwei ETK-CTRL Subsysteme zugeordnet werden. Diese ETK-CTRL Subsysteme entsprechen den beiden auf der Karte vorhandenen ETK-Schnittstellen Controllern.

Das ES1201-ETK Subsystem wird auch für die Einbindung der ES1200 verwendet. Da diese Karte nur eine ETK-Schnittstelle aufweist, darf hierfür nur ein ETK-CTRL Subsystem zugeordnet werden (mit ETK Port A).

10.3.2 Globals (ETK-CTRL Subsystem)

In der Registerkarte „Globals“ eines ETK-CTRL Subsystems wird dem ETK-CTRL Subsystem ein physikalischer ETK-Controller bzw. ein ETK-Anschluss zugeordnet.

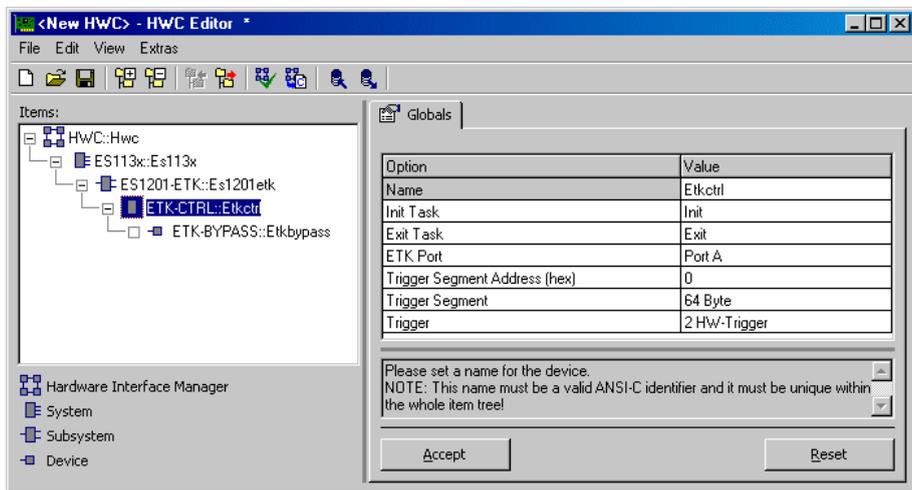


Abb. 10-6 Die Registerkarte „Globals“ des ETK-CTRL Subsystems

ETK Port

In dieser Zeile wird einer der beiden unabhängigen ETK-Kanäle (Port A, Port B) und damit einer der beiden ETK-Controller der ES1201 als ETK-Schnittstelle zugeordnet. Bei der ES1200 kann hier nur der Port A gewählt werden.

Trigger Segment Address (hex)

Hier muss die Triggersegmentadresse des DISTAB-Verfahrens angegeben werden.

Ist dem ETK-CTRL Subsystem ein ETK-Bypass Item mit passendem ASAM-MCD-2MC-Projekt zugeordnet, so ist diese Einstellung nicht editierbar und wird automatisch vorgenommen, sofern die entsprechende Größe im ASAM-MCD-2MC-Projekt ordnungsgemäß definiert wurde.

Die Triggersegmentadresse bestimmt die Lage der Hardware-Triggeradressen die zur Steuerung der Datenübertragung über die Bypass-Kanäle genutzt werden. Die Lage der Hardware-Triggeradressen in Relation zur Triggersegmentadresse ist beim DISTAB-Datenaustauschverfahren konstant.

Die Anfangsadresse eines 64 Byte Triggersegments (bei 8 oder 16 Bit breitem Buszugriff) muss auf einer geraden, durch 64 teilbaren Adresse liegen, die Anfangsadresse eines 128 Byte Triggersegments (bei 32 Bit breiten Buszugriff) entsprechend auf einer durch 128 teilbaren Adresse.

Das Format des Trigger Segments ist in der DISTAB 12 und DISTAB 13 Schnittstellenbeschreibung definiert.

Trigger Segment

Hier muss die Größe des Triggersegments eingestellt werden.

Ist dem ETK-CTRL Subsystem ein ETK-BYPASS Item mit passendem ASAM-MCD-2MC-Projekt zugeordnet, so ist diese Einstellung nicht editierbar und wird automatisch vorgenommen, sofern die entsprechende Größe im ASAM-MCD-2MC-Projekt ordnungsgemäß definiert wurde.

Der Adressbereich des Triggersegments darf vom Steuergerät nicht verwendet werden. Bei 16 Bit breiten Triggeradressen ist das Triggersegment 64 Byte groß und bei 32 Bit breiten Triggeradressen ist das Triggersegment 128 Byte groß.

Trigger

Hier wird die Anzahl der für den Bypass verwendeten Hardware-Triggeradressen (2 oder 16) eingestellt.

Ist dem ETK-CTRL Subsystem ein ETK-Bypass Item mit passendem ASAM-MCD-2MC-Projekt zugeordnet, so ist diese Einstellung nicht editierbar und wird automatisch vorgenommen, sofern die entsprechende Größe im ASAM-MCD-2MC-Projekt ordnungsgemäß definiert wurde.

10.3.3 Globals (ETK-BYPASS Device)

Dieses ETK-BYPASS Device dient zur Definition aller notwendigen Größen um einen Bypass-Betrieb mit einem Bypass fähigen Steuergerät, welches mit ETK und der passenden Software ausgerüstet ist, zu realisieren.

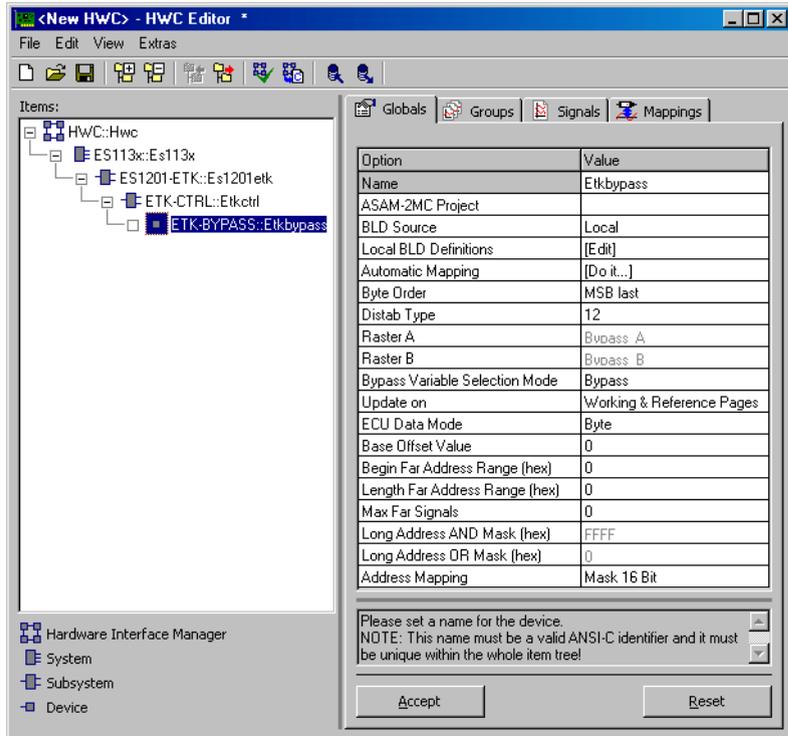


Abb. 10-7 Die Registerkarte „Globals“ des ETK-BYPASS Device

ASAM-2MC Project

Der ETK-Bypass benötigt ein ASAM-MCD-2MC-Projekt (AML V1.1), das in der Datenbank durch Einlesen einer ASAM-MCD-2MC-Beschreibungsdatei generiert werden kann.

Hinweis

ASAM-MCD-2MC-Projekte, die AML V1.2 oder höher verwenden, können **nicht** eingelesen werden. Folgende Fehlermeldung erscheint, wenn dem Device ein entsprechendes Projekt zugeordnet wird:

```
Error: Incompatible version 0x<version> of QP_BLOB
found in SOURCE '<sName>'. Expected version = 0x1.
Selected ASAM-2MC Project is not suitable for the basic
ETK-Controller (ES1232/ETK-CTRL-BAS, ES1231/ETK-CTRL,
ES1201/ETK-CTRL) ! Please use an advanced ETK-Controller
(ES1232/ETK-CTRL-ADV)!
```

Die ASAM-MCD-2MC-Datei enthält unter IF_DATA ETK (in älteren Versionen auch IF_DATA ASAP1B_ETK) die Beschreibung des Bypass. Für jeden Bypass-Kanal existiert ein QP_BLOB mit den Kanaleinstellungen; ein Beispiel finden Sie im Abschnitt „Bypass-Kommunikation (AML V1.1)“ auf Seite 133. Im TP_BLOB sind allgemeine Einstellungen eingetragen.

```
/begin TP_BLOB
    0x1000100    /* TP_BLOB version          */
    2           /* Project Base Address    */
    0x0         /* RESET_CFG (only PPC family CPU)*/
/begin DISTAB_CFG 0xC 0x1 MSB_LAST 0x383000 0x0
    TRG_MOD 0xB7
/end DISTAB_CFG
CODE_CHK /* check whether program and data */
         /* are matching                    */
    0x0 /* program ID address in data range */
    0x0 /* program ID length in data range  */
    0x0 /* program ID address in external RAM */
    0x0 /* program ID length in external RAM */
ETK_CFG 0xF 0xF0 0xFF 0x3 0xFD 0xEE 0xFF 0x1
         /* ETK configuration                */
```

```

RESERVED 0x810000 /* start address */
          0x8103F9 /* length */
EXTERN   /* memory attribute */
0x-1 0x-1 0x-1 0x-1
          /* mirror offsets 1 - 5 */

```

/end TP_BLOB

In der Datei nicht enthaltene notwendige Informationen können in den Registern des HWC-Editors von Hand eingegeben werden.

Klicken Sie neben der Option „ASAM-2MC Project“ in die Spalte „Value“. Ein Dialogfenster öffnet sich, in dem das gewünschte ASAM-MCD-2MC-Projekt ausgewählt werden kann.

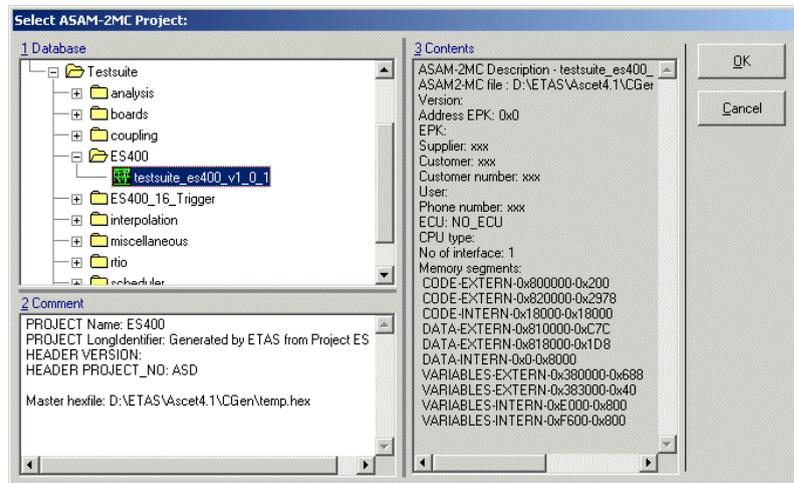


Abb. 10-8 Auswahlfenster „Select ASAM-2MC Project“

BLD Source

Mit dieser Option wird festgelegt, von welcher Quelle aus die Abhängigkeit zwischen den benötigten ASAM-MCD-2MC-Größen für die Bypass-Ausgangsgrößen definiert wird (BLD = Bypass Label Dependence).

Prinzipiell stehen hierfür folgende Möglichkeiten zur Verfügung:

- ASAM-2MC File Hier findet die Definition der Abhängigkeiten zwischen den Bypass-Ausgangsgrößen innerhalb der ASAM-MCD-2MC-Datei statt. Diese Möglichkeit wird aber z.Z. noch von den allerwenigsten ASAM-MCD-2MC-Dateien unterstützt.
- Local Hier wird die lokale Definition vom ETK-BYPASS Device übernommen (siehe nächster Abschnitt „Local BLD Definitions“).

Local BLD Definitions

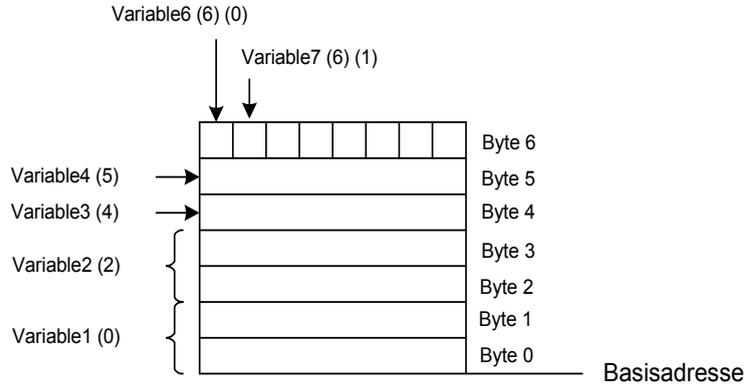
Die Datenübertragung der Bypass-Ausgangsgrößen von ASCET zum Steuergerät erfolgt jeweils über einen eigenen Datenpuffer für jedes Raster (winkelsynchron / zeitsynchron).

Einige Steuergeräteprogramme unterstützen eine flexible Zuordnung von Bypassgrößen zu diesem Datenpuffer. Diese flexible Zuordnung ist spätestens dann notwendig, wenn mehr Bypassgrößen von der Steuergerätesoftware zur Verfügung gestellt werden als über den zur Verfügung stehenden Datenpuffer übertragen werden können.

Für jede Bypass-Ausgangsgröße (als Measurement definiert) sind je nach Steuergeräte-Programm bis zu 3 zusätzliche Verstellgrößen (Characteristics) notwendig, um die Bypass-Größe zu definieren:

Byte Offset- bzw.Vektor- Parameter	Definiert den Byte-Offset der Bypass-Ausgangsgröße relativ zum Anfang des Datenpuffers
Bit Offset	Ist nur für Bit-Größen vorhanden und kennzeichnet die Bitposition innerhalb eines Bytes
Source	Einige Steuergeräteprogramme unterstützen eine wahlfreie Zuordnung einer Bypassgröße zu einem Bypass-Raster bzw. Source (winkelsynchron / zeitsynchron). Der Parameter enthält hierfür den passenden Wert (0 = winkelsynchron / 1 = zeitsynchron)

Die folgende Abbildung zeigt ein Beispiel:



In diesem Fall liegt die Bitgröße `Variable6` auf dem Bit 0 des Byte 6 im Datenpuffer für die Bypass-Ausgangsgrößen, die Offset-Werte sind daher (Byte-Offset: 6 / Bit-Offset: 0).

`Variable7`, ebenfalls eine Bitgröße, liegt auf Bit 1 im Byte 6, die Werte sind entsprechend (Byte-Offset: 6 / Bit-Offset: 1).

`Variable3` ist ein Byte und hat den Byte-Offset 4, da es auf Byte 4 des Speicherbereichs liegt, `Variable4` hat den Byte-Offset 5.

`Variable1` steht am Anfang des Datenpuffers und hat den Byte-Offset 0. `Variable2` hat den Byte-Offset 2, da es sich bei `Variable1` um ein Word handelt, welches zwei Bytes belegt.

Klicken Sie auf den Eintrag `[Edit]` in der Spalte „Value“, um einen kleinen Texteditor zu öffnen. In diesem wird die lokale Definition der Abhängigkeiten zwischen den Bypass-Labels vorgenommen (BLD Bypass Label Dependencies).

Der Text ist zeilenweise aufgebaut, wobei jede Zeile genau ein Bypass-Ausgangssignal beschreibt. Die Definition der Label-Abhängigkeiten eines Bypass-Ausgangssignals hat folgende Syntax:

```
<Raster>, <Signal>, [<Byte Offset>], [<Bit Offset>], [<Source>]
```

Raster: „A“ (winkelsynchron), „B“ (zeitsynchron) oder „AB“ (winkel- und zeitsynchron)

Signal: Name des ASAM-MCD-2MC-Measurements der Bypass-Ausgangsgröße

Byte Offset: Name des optionalen ASAM-MCD-2MC-Characteristic, welches in der Regel aber immer vorhanden ist, und der Definition des Byte-Offsets dient

Bit Offset: Name des optionalen ASAM-MCD-2MC-Characteristic, welches der Definition der Bit-Position bei Bit-Größen dient

Source: Name des optionalen ASAM-MCD-2MC-Characteristic. Bei mehr als einem Raster (AB) wird hier das aktuell ausgewählte Raster eingetragen..

// Dient zur Einleitung eines Kommentars bis Zeilenende

Die Offsets für die Größen, die hier definiert sind, können automatisch mit ASCET konfiguriert werden.

Beispiel für eine Local BLD Definition:

```

1 // current version of the BLD
sgetas2 // (optional) name of the dependent
// ASAP2 file
// (is ignored by BLD reader)
A, tlb_8, EBOTT1_8 // definition of a bypass signal
// available for angle sync sample
// grid
B, B_tlb, EBOTBT1, BOBBT1
// definition of a bypass bit
// signal available for time
// sync grid
AB, B_Test, B_Test_Vector,, B_Test_Channel
// definition of a bypass signal
// available for both sample grids
// with definition for the actual
// sample grid (source) parameter
A, log_uint8_0_A, log_uint8_0_offset_A.Model_Byp_A,
log_uint8_0_bitOffset_A.Model_Byp_A
// definition of a bypass signal
// available for angle sync sample
// grid

```

Dieses letzte Beispiel entspricht in der ASAM-MCD-2MC-Datei (AML V1.1) dem nachstehenden Abschnitt MEASUREMENT:

```

/begin MEASUREMENT log_uint8_0_A
" "
UBYTE
ident
1

```

```
100
0.0
1.0
READ_ONLY
BIT_MASK 0x8
ECU_ADDRESS 0xFD00
/end MEASUREMENT
```

Automatic Mapping

Siehe hierzu die Beschreibung „Automatic Mapping“ im Abschnitt 10.4.3 auf Seite 169.

Byte Order

In dieser Zeile wird das Wort-Datenablageformat (MSB first / MSB last) des Steuergeräteprozessors angezeigt. Diese Information wird aus dem zugeordneten ASAM-MCD-2MC-Projekt ausgelesen (siehe Seite 132).

Übersicht über die gebräuchlichen „Byte Order“ Bezeichnungen:

MSB first	big endian	Motorola
MSB last	little endian	Intel

Distab Type

In dieser Zeile wird dargestellt, welches DISTAB-Verfahren zum Datenaustausch mit dem Steuergerät verwendet wird. Diese Information wird aus dem zugeordneten ASAM-MCD-2MC-Projekt ausgelesen (siehe Seite 132).

DISTAB 12 unterstützt Signale mit bis zu zwei Byte Länge

DISTAB 13 unterstützt 1, 2, 4 und 8 Byte lange Signale

Raster A / B

In diesen Zeilen wird die aus dem ASAM-MCD-2MC-Projekt ausgelesene Bezeichnungen der beiden Raster A und B (winkelsynchron und zeitsynchron) angezeigt.

Bypass Variable Selection Mode

In dieser Zeile kann ausgewählt werden, ob in der Auswahlliste für die Bypass-Ausgangsgrößen („send“ Signalgruppen im Register „Groups“) alle Größen des Steuergeräts, die in der ASAM-MCD-Beschreibung als „Measurements“ definiert wurden, angezeigt werden („all“) oder nur die Größen, die über die BLD Definition als Bypass-Ausgangsgrößen identifiziert wurden („Bypass“).

Für die Bypass-Eingangsgrößen („receive“ Signalgruppen im Register „Groups“), die von dieser Einstellung nicht betroffen sind, stehen unabhängig davon immer alle definierten Messgrößen des ASAM-MCD-2MC-Projekts zur Verfügung.

Update on

In dieser Zeile kann gewählt werden, ob während der Bypass-Kommunikation die Datenübergabe zwischen Simulationsrechner und Steuergerät auf der Arbeitsseite (Working Page), Referenzseite (Reference Page) oder auf Arbeits- und Referenzseite (Working & Reference Page) des ETK durchgeführt werden soll.

ECU Data Mode

In dieser Zeile wird dargestellt mit welcher Zugriffsart (Byte-Zugriff / Wort-Zugriff) der Steuergeräte-Prozessor auf den Datenspeicher zugreift.

Diese Einstellung wird aus dem ASAM-MCD-2MC-Projekt übernommen (s. Seite 132).

Base Offset Value

In dieser Zeile kann der Byte-Offset jeder Bypass-Ausgangsgröße um 0, 2 oder 8 Byte nach oben verschoben werden.

Der Standardwert für den Byte-Offset bei DISTAB 12 ist 0. Bei DISTAB 13 ist der Standardwert für den Byte-Offset 8. Abweichende Einstellungen sind Sonderlösungen und bedürfen der Information durch den Steuergeräteprogrammierer.

Begin Far Address Range (hex)

In dieser Zeile kann die Anfangsadresse des FAR-Adressbereichs angegeben werden. Dies ist in der Regel nur für DISTAB12-Verfahren notwendig.

Variablen aus dem FAR-Adressbereich des Steuergeräts können bei Steuergeräten mit C16x-Mikrocontroller, bei denen das DISTAB12-Verfahren verwendet wird, nicht ausgelesen oder beschrieben werden. Alle Werte im FAR-Adressbereich werden durch einzelne ETK-Zugriffe gelesen, was den Datentransfer verlangsamt.

Length Far Address Range (hex)

In dieser Zeile wird die Größe des FAR-Adressbereichs angegeben. Diese Angabe ist in der Regel nur für DISATB12-Verfahren notwendig.

Max Far Signals

In dieser Zeile wird festgelegt, wieviele Variablen maximal aus dem FAR-Bereich gelesen werden dürfen. Diese Angabe ist in der Regel nur für DISTAB 12 Verfahren notwendig.

Das Lesen vieler Variablen aus dem FAR-Bereich wirkt sich negativ auf die Laufzeit des ETK-Bypass aus. Deshalb sollten immer so wenig Werte wie möglich aus diesem Bereich gelesen werden.

Long Address AND / OR Mask (hex)

Diese beiden Zeilen zeigen die Masken an, die in der Zeile „Address Mapping“ eingestellt wurden.

Address Mapping

Diese Zeile kann nur editiert werden, wenn das DISTAB12-Verfahren gewählt ist.

Mit dieser Zeile können Bit-Masken definiert werden, mit deren Hilfe eine Adresstransformation durchgeführt werden kann. Bei Steuergeräten mit C16x Mikrocontroller, die Speicheradressen größer als 16 Bit verwenden, können diese mit den Bit-Masken auf Adressen innerhalb des 16 Bit-Adressbereichs der DISTAB 12 umgesetzt werden.

Die Formel dafür lautet:

$$\text{<Steuergeräte-Adresse> := <Speicheradresse> \& \text{<AND Mask>} \mid \text{<OR Mask>}$$

Die Masken werden in den Zeilen „Long Address AND / OR Mask (hex)“ angezeigt. Die nachstehende Tabelle enthält die Auswahlmöglichkeiten für „Address Mapping“ sowie die zugehörigen Masken.

Address Mapping	Long Address	
	AND Mask (hex)	OR Mask (hex)
Mask 16 Bit	0xFFFF	0x0000
C167x DPP0	0x3FFF	0x0000
C167x DPP1	0x3FFF	0x4000
C167x DPP2	0x3FFF	0x8000
C167x DPP3	0x3FFF	0xC000

10.3.4 Groups (ETK-BYPASS Device)

Die Anzahl der Signalgruppen (4) ist für das ETK-BYPASS Device fest vorgegeben. Die ersten beiden Signalgruppen betreffen das erste, in der Regel winkelsynchrone Raster. Die beiden letzten Signalgruppen betreffen das zweite Bypass-Raster, welches in der Regel das zeitsynchrone Raster ist.

Jedes Raster besteht aus zwei Signalgruppen; einer Sende- und eine Empfang-Signalgruppe.

Die Daten der Empfangs-Signalgruppe werden über das Standard-DISTAB-Verfahren übertragen, wohingegen die Daten der Sende-Signalgruppe über das Bypass-Tabellenverfahren zwischen Steuergerät und RTIO ausgetauscht werden.

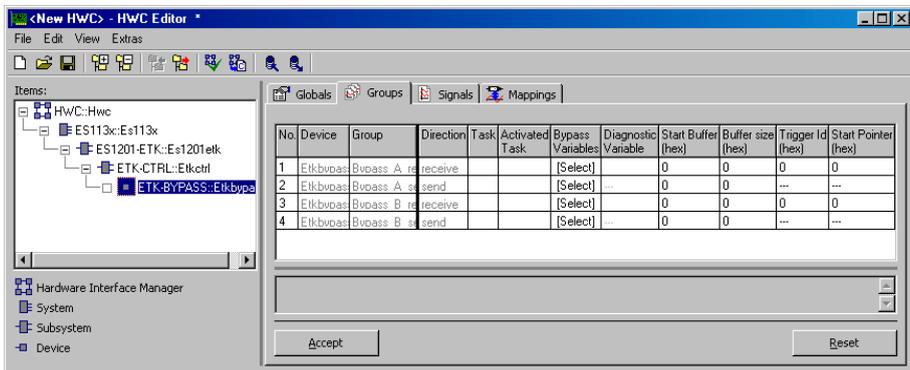


Abb. 10-9 Die Registerkarte „Groups“ des ETK-BYPASS Devices

Hinweis

Bei Änderung des Signalgruppennamens, des Signalnamens oder der Signalrichtung ('Direction') wird die die eventuell zugeordnete ASCET-Message u.U. nicht mehr automatisch zugeordnet und muss manuell neu zugeordnet werden.

Task

Für eine ordnungsgemäße Bypass-Funktion sollte folgende Task-Zuordnung vorgenommen werden:

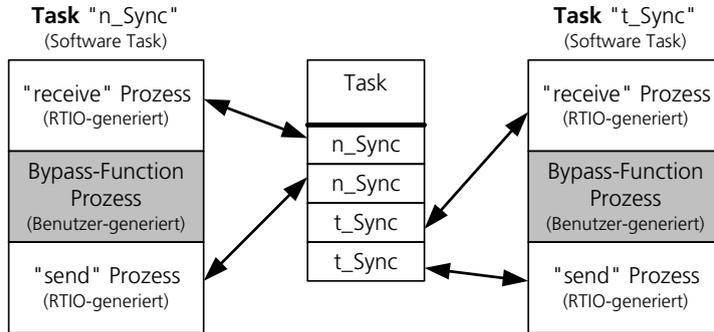


Abb. 10-10 Die ETK-Bypass Standard Task-Zuordnung

Activated Task

Werden in diese Spalte Software-Tasks eingegeben, so werden diese jeweils beim Empfang bzw. beim Versenden der Daten durch einen impliziten „activate Task“-Aufruf aktiviert.

Bypass Variables

Siehe Abschnitt „Bypass Variables“ auf Seite 187 in Kapitel 10.5.4.

Diagnostic Variable

Siehe Abschnitt „Diagnostic Variable“ auf Seite 188 in Kapitel 10.5.4.

Start Buffer (hex)

In dieser Spalte müssen die Basisadressen der jeweiligen Datenpuffer für die Bypass-Eingangsdaten und -Ausgangsdaten eingetragen werden.

Normalerweise sind diese Adressangaben im ASAM-MCD-2MC-Projekt enthalten, sofern es sich um ein ASAM-MCD-2MC-Projekt handelt, das auch den Bypass-Betrieb korrekt unterstützt. In diesem Fall sind die entsprechenden Werte aus dem ASAM-MCD-2MC-Projekt eingetragen und die Eingabe ist gesperrt.

Die zuständigen Parameter hierfür lauten (Tab. 9-1, Spalte „Parameter“):

CHNL_S, CHNL_T, CHNL_X, CHNL_Y

Buffer size (hex)

In dieser Spalte müssen die Größen der jeweiligen Datenpuffer für die Bypass-Eingangsdaten und -Ausgangsdaten eingetragen werden.

Normalerweise sind diese Angaben im ASAM-MCD-2MC-Projekt enthalten, sofern es sich um ein ASAM-MCD-2MC-Projekt handelt, das auch den Bypass-Betrieb korrekt unterstützt. In diesem Fall sind die entsprechenden Werte aus dem ASAM-MCD-2MC-Projekt eingetragen und die Eingabe ist gesperrt.

Die zuständigen Parameter hierfür lauten (Tab. 9-1, Spalte „Parameter“):

`BPMAX_S, BPMAX_T, BPMAX_X, BPMAX_Y`

Trigger Id (hex)

In dieser Spalte müssen für die „receive“-Signalgruppen die Adressen für die Trigger-Identifizier für das verwendete DISTAB-Verfahren eingetragen werden.

Normalerweise sind diese Angaben im ASAM-MCD-2MC-Projekt enthalten, sofern es sich um ein ASAM-MCD-2MC-Projekt handelt, das auch den Bypass-Betrieb korrekt unterstützt. In diesem Fall sind die entsprechenden Werte aus dem ASAM-MCD-2MC-Projekt eingetragen und die Eingabe ist gesperrt.

Die zuständigen Parameter hierfür lauten (Tab. 9-1, Spalte „Parameter“):

`TRGID_S, ---, TRGID_X, ---`

Start Pointer (hex)

In dieser Spalte müssen für die „receive“ Signalgruppen die Startadressen der Pointerlisten für das verwendete DISTAB-Verfahren eingetragen werden.

Normalerweise sind diese Angaben im ASAM-MCD-2MC-Projekt enthalten, sofern es sich um ein ASAM-MCD-2MC-Projekt handelt, das auch den Bypass-Betrieb korrekt unterstützt. In diesem Fall sind die entsprechenden Werte aus dem ASAM-MCD-2MC-Projekt eingetragen und die Eingabe ist gesperrt.

Die zuständigen Parameter hierfür lauten (Tab. 9-1, Spalte „Parameter“):

`BYPASS_S, ---, BYPASS_X, ---`

10.3.5 Signals (ETK-BYPASS Device)

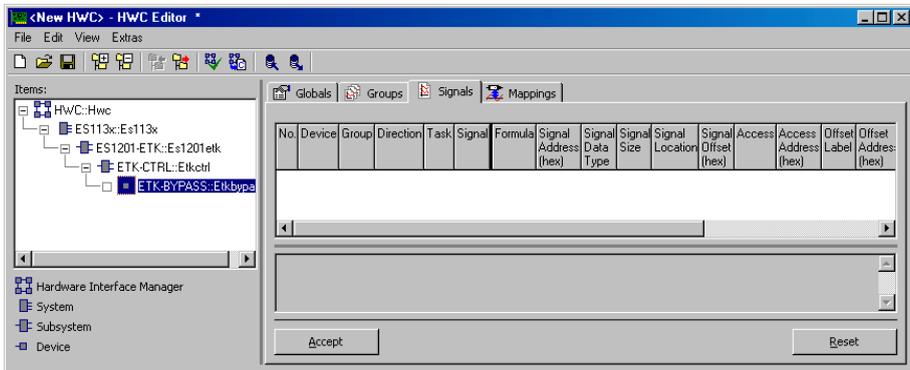


Abb. 10-11 Die Registerkarte „Signals“ des ETK-BYPASS Devices

Hinweis

Keine der Spalten der Registerkarte „Signals“ ist vom Anwender editierbar. Sie dienen ausschließlich zur Anzeige von Statuswerten für die Bypassgrößen.

Die Sortierung der Bypassgrößen innerhalb der jeweiligen Signalgruppe erfolgt gemäß ihrer Bytegröße. Für „send“-Signale gilt die Reihenfolge 8-4-2-1 Byte.

Für „receive“-Signale muss unterschieden werden, ob es sich um interne Größen des Steuergeräte-Kontrollers handelt, die nur indirekt über das DISTAB-Verfahren gelesen werden können, oder ob die Größen im externen Speicherbereich des Steuergeräte-Kontrollers liegen, der auch vom ETK direkt gelesen werden kann. Hierfür gilt somit die Reihenfolge 8-4-2-1 Byte direkt und anschließend 8-4-2-1 indirekt.

Signal Address (hex)

In dieser Spalte werden die ASAM-MCD-2MC-Measurement Speicheradressen der Bypassgrößen angezeigt.

Signal Data Type

In dieser Spalte werden die Datentypen der Bypassgrößen angezeigt.

Signal Size

In dieser Spalte wird die Datengröße der Bypassgrößen in Byte angezeigt.

Signal Location

Diese Spalte ist nur für „receive“-Signale relevant und zeigt die Speicherlocation (Intern / Extern) der Bypassgrößen an.

- Intern:
„Intern“ bedeutet, dass die Bypassgröße im internen RAM des Steuergeräte-Kontrollers liegt und somit nur über den DISTAB Mechanismus ausgelesen werden kann.
- Extern
„Extern“ bedeutet, dass die Größe im externen RAM des Steuergeräte-controllers liegt und u.U. direkt vom ETK ausgelesen werden kann.

Signal Offset (hex)

In dieser Spalte wird der (0 basierte) Index der Größe innerhalb des Datenpuffers angezeigt.

Access

Diese Spalte ist nur für „receive“ Signale relevant und zeigt die tatsächliche Zugriffsart auf die Messgröße an:

„Distab“ Die Größe wird über DISTAB-Verfahren gelesen
„Direct“ Die Größe wird mittels direktem ETK-Zugriff gelesen

Access Address (hex)

Diese Spalte zeigt die ggf. umgerechneten Zieladressen der Bypassgrößen an. Das ASAM-MCD-2MC-Projekt kann für den ETK-Zugriff sogenannte „Memory-Segmente“ enthalten, die für bestimmte Adressen eine Umrechnung enthalten um schließlich auf die Zieladresse zu gelangen.

Zusätzlich wird für das DISTAB 12 Verfahren u.U. auch die Adressmasken „AND Mask“ und „OR Mask“ aus dem „Globals“ Register berücksichtigt.

Die Formel dafür lautet:

$$\langle \text{Zieladresse} \rangle := (\langle \text{Signaladresse} \rangle \& \langle \text{AND Mask} \rangle \mid \langle \text{OR Mask} \rangle)$$

(eventuell noch mit „Memory-Segment“ Umrechnung)

Offset Label

Diese Spalte ist nur für „send“-Größen relevant und zeigt den Namen des Vektor-Parameters an, sofern für die Bypassgröße eine BLD Referenz definiert ist (siehe hierzu auch „Local BLD Definitions“ auf Seite 152).

Offset Address (hex)

Diese Spalte ist nur für „send“-Größen relevant und zeigt die Adresse des Vektor-Parameters an, sofern für die Bypassgröße eine BLD Referenz definiert ist (siehe hierzu auch „Local BLD Definitions“ auf Seite 152).

Offset Value (hex)

Diese Spalte ist nur für „send“-Größen relevant und zeigt den Wert an, auf den der Parameter beim Starten des Bypass verstellt wird, um die betreffende Bypassgröße zu aktivieren. Dieser Wert ist nur vorhanden, falls für die Bypassgröße eine BLD Referenz definiert ist (siehe hierzu auch „Local BLD Definitions“ auf Seite 152).

Formel für die Berechnung des Wertes:

$$\text{Offset Value} = \text{Signal Offset} + \text{Base Offset Value}$$

(Der Wert für `Base Offset Value` stammt aus dem Register „Globals“.)

Bit Label

Diese Spalte ist nur für „send“-Größen relevant und zeigt den Namen des Bit-Offset-Parameters an, sofern für die Bypassgröße eine BLD Referenz definiert ist und es sich hierbei um eine Bitgröße handelt (siehe hierzu auch „Local BLD Definitions“ auf Seite 152).

Bit Address (hex)

Diese Spalte ist nur für „send“ Größen relevant und zeigt die Adresse des Bit-Offset-Parameters an, sofern für die Bypassgröße eine BLD Referenz definiert ist und es sich hierbei um eine Bitgröße handelt (siehe hierzu auch „Local BLD Definitions“ auf Seite 152).

Bit Value

Diese Spalte ist nur für „send“-Größen relevant und zeigt den Wert an, auf den der Bit-Offset-Parameter beim Starten des Bypass verstellt wird, um die betreffende Bitposition innerhalb eines Datenbytes anzugeben (0 basiert). Dieser Wert ist nur vorhanden, falls es sich hierbei um eine Bitgröße handelt und falls für die Bypassgröße eine BLD Referenz definiert ist (siehe hierzu auch „Local BLD Definitions“ auf Seite 152).

Src Label

Diese Spalte ist nur für „send“-Größen relevant. Sie zeigt den Namen des „Source“-Parameters an, sofern für die Bypassgröße eine BLD-Referenz definiert ist (siehe hierzu auch „Local BLD Definitions“ auf Seite 152).

Src Address (hex)

Diese Spalte ist nur für „send“-Größen relevant. Sie zeigt die Adresse des „Source“-Parameters (Src Label) an, sofern für die Bypassgröße eine BLD-Referenz definiert ist (siehe hierzu auch „Local BLD Definitions“ auf Seite 152).

Src Value

Diese Spalte ist nur für „send“-Größen relevant. Sie zeigt den Wert an, auf den der Parameter (Src Label) beim Starten des Bypass verstellt wird, um dem Steuergerät mitzuteilen, in welchem Raster die betreffende Bypassgröße aktualisiert werden soll. Soll die Größe im winkelsynchronen Raster (A) angezeigt werden, hat `Src Value` den Wert 0, soll die Größe im zeitsynchronen Raster (B) angezeigt werden, hat `Src Value` den Wert 1.

Dieser Wert ist nur vorhanden, falls für die Bypass-Größe eine BLD-Referenz definiert ist (siehe hierzu auch „Local BLD Definitions“ auf Seite 152).

10.3.6 Mappings (ETK-BYPASS Device)

Die Einstellmöglichkeiten sind für alle Items gleich. Sie werden in Kapitel 7.3.5 auf Seite 120 beschrieben.

10.4 ES1222-CAN (CAN-IO)

Die ES1222-Einschubkarte dient als CAN-Schnittstelle in VMEbus-Systemen. Die Karte stellt vier CAN-Kanäle zur Verfügung, die über getrennte CAN-Controller vom Typ Intel 82527 verfügen.

Die Karte bietet zudem die Möglichkeit, bei Empfang von CAN-Botschaften Interrupts auf dem Simulationsrechner auszulösen, sodass ein dauerndes Pollen nach Botschaften entfallen kann. Jeder CAN-Controller kann bis zu 255 Sende- und 255 Empfangsbotschaften verarbeiten. Durch Einsatz eines eigenen Prozessors wird der System-Controller wesentlich entlastet.

Hinweis

Da der FIFO-Speicher nur 82 Botschaften pro Kanal fasst, ist es nicht möglich, die gesamten 255 Botschaften zum gleichen Zeitpunkt zu versenden. Sie müssen von mehreren Tasks zu unterschiedlichen Zeitpunkten an den FIFO-Speicher übergeben werden, damit dieser nicht überfüllt wird.

Dieser Abschnitt beschreibt die RTIO-Einbindung der ES1222-Karte (alte Bezeichnung: VSIC-Karte). Im HWC-Editor wird die ES1222-Karte durch Auswahl des Items „ES1222-CAN“ eingebunden.

Grundsätzlich können über diese Karte alle Gegenstellen angeschlossen werden, die über CAN mit fest definierten Botschaften kommunizieren.

10.4.1 Globals (ES1222-CAN Subsystem)

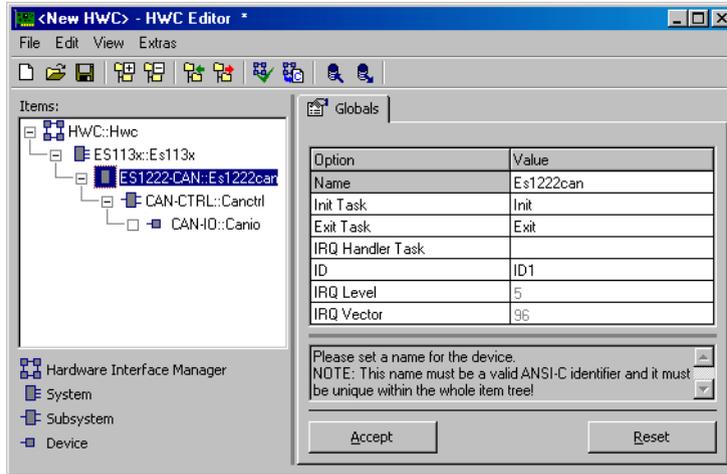


Abb. 10-12 Die Registerkarte „Globals“ des ES1222-CAN Subsystems

Dem ES1222-CAN Subsystem können bis zu 4 CAN-CTRL Subsysteme zugeordnet werden. Diese „CAN-CTRL“ Items entsprechen den 4 auf der Karte vorhandenen Intel 82527 CAN-Controllern.

IRQ Handler Task

Für die Unterstützung von Interrupts der ES1222-Karte wird eine zusätzliche „IRQ Handler Task“ benötigt. Diese Task muss im ASCET-Projekteditor in der Task-Liste als „Software“-Task angelegt werden, wobei bei „Max. No. of Activations“ mindestens die Anzahl 2 (maximal 50) einzutragen ist.

ID

Im Auswahlfeld „ID“ muss die Kartennummer der Karte eingegeben werden, welche angesprochen werden soll.

Hinweis

Bei der ES1222 Karte handelt es sich um eine sogenannte „Auto-ID“ Karte, die beim Einschalten des Systems vom Hardware-Manager automatisch eine ID-Nummer und einen freien Adressbereich zugewiesen bekommt. Karten vom selben Typ (z.B. zwei ES1222) werden von links nach rechts nummeriert, d.h. die linke Karte erhält die Nummer Eins, anschließend die Nummer Zwei usw. Es stehen 4 IDs zur Verfügung; somit können bis zu 4 Karten in einem Experimentalsystem betrieben werden.

10.4.2 Globals (CAN-CTRL Subsystem)

In der Registerkarte „Globals“ eines CAN-CTRL Subsystems wird dem CAN-CTRL Subsystem ein physikalischer CAN-Controller bzw. ein CAN-Anschluss zugeordnet.

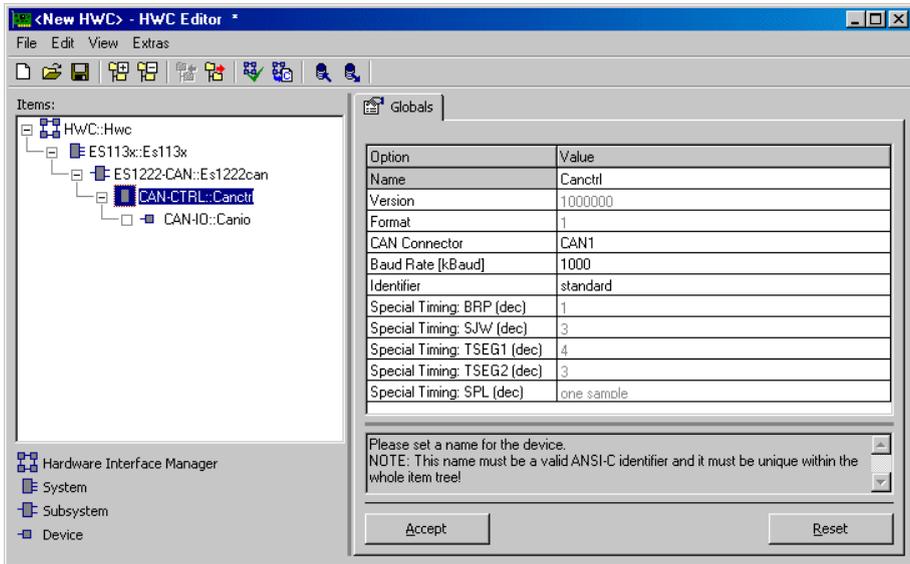


Abb. 10-13 Die Registerkarte „Globals“ des CAN-CTRL Subsystems

CAN Connector

Hier muss der gewünschte CAN-Controller bzw. CAN-Anschluss (Port A, Port B, Port C oder Port D) ausgewählt werden.

Baud Rate [kBaud]

Hier kann die gewünschte Baud-Rate ausgewählt werden. Es stehen hierfür die 8 Standard-Baud-Raten (1000, 500, 250, 125, 100, 50, 20, 10 kBaud) zur Verfügung. Zusätzlich besteht die Möglichkeit, die Einstellung `<Special Timing>` zu wählen, die die Low-Level-Ansteuerung des CAN-Controllers bzgl. Bit-Timing und Baud-Rate aktiviert. Die nachfolgend beschriebenen 5 Optionen sind für die Vorgabe dieser Special Timing-Einstellungen notwendig.

Identifizier

Bei CAN-Botschaften kann zwischen *standard frames* mit 11 Bit-Identifiern oder *extended frames* mit 29 Bit-Identifiern gewählt werden. In dieser Zeile wird die Länge des Identifier-Felds (*standard / extended*) festgelegt.

Hinweis

Für ein CAN-CTRL Device wird eine Mischung aus *standard frames* und *extended frames* **nicht** unterstützt.

Wenn der Identifier *standard* gewählt wird, ist im Register „Groups“ des CAN-IO Device (s. „Identifier dec/hex“ auf Seite 176) nur die Eingabe von 11 Bit-Werten erlaubt. Wird ein größerer Wert eingegeben, werden die *most significant Bits* (MSB) abgeschnitten. Eine Warnung wird nicht angezeigt.

Wenn der Identifier *extended* gewählt wird, ist nur die Eingabe von 29 Bit-Werten erlaubt. Wird ein größerer Wert eingegeben, werden die *most significant Bits* (MSB) abgeschnitten. Eine Warnung wird nicht angezeigt.

Special Timing: BRP (dec)

Diese Option ist standardmäßig ausgeblendet und kann nur bearbeitet werden, wenn die Option „Baud Rate“ auf `<Special Timing>` eingestellt ist.

Dieser Parameter dient zur Einstellung des „Baud Rate Prescaler“, welcher aus dem Eingangstakt des CAN-Controllers die Baud Rate bestimmt. Der Wertebereich dieser Einstellung beträgt 0 - 63.

Mehr Informationen zu dieser Einstellung finden Sie im Datenblatt des Intel 82527 CAN Controllers.

Special Timing: SJW (dec)

Diese Option ist standardmäßig ausgeblendet und kann nur bearbeitet werden, wenn die Option „Baud Rate“ auf `<Special Timing>` eingestellt ist.

Dieser Parameter dient zur Einstellung der „Synchronization Jump Width“. Der Wertebereich dieser Einstellung beträgt 0 - 3.

Mehr Informationen zu dieser Einstellung finden Sie im Datenblatt des Intel 82527 CAN Controllers.

Special Timing: TSEG1 (dec)

Diese Option ist standardmäßig ausgeblendet und kann nur bearbeitet werden, wenn die Option „Baud Rate“ auf `<Special Timing>` eingestellt ist.

Dieser Parameter dient zur Einstellung des „Time Segment 1“ und bestimmt damit die Zeitspanne vor dem Abtastzeitpunkt. Der Wertebereich dieser Einstellung beträgt 2 - 15.

Mehr Informationen zu dieser Einstellung finden Sie im Datenblatt des Intel 82527 CAN Controllers.

Special Timing: TSEG2 (dec)

Diese Option ist standardmäßig ausgeblendet und kann nur bearbeitet werden, wenn die Option „Baud Rate“ auf <Special Timing> eingestellt ist.

Dieser Parameter dient zur Einstellung des „Time Segment 2“ und bestimmt damit die Zeitspanne nach dem Abtastzeitpunkt. Der Wertebereich dieser Einstellung beträgt 1 – 7.

Mehr Informationen zu dieser Einstellung finden Sie im Datenblatt des Intel 82527 CAN Controllers.

Special Timing: SPL (dec)

Diese Option ist standardmäßig ausgeblendet und kann nur bearbeitet werden, wenn die Option „Baud Rate“ auf <Special Timing> eingestellt ist.

Dieser Parameter dient zur Einstellung des „Sampling Mode“ und bestimmt, wie oft das Signal abgetastet wird, um den logischen Zustand zu ermitteln.

Mehr Informationen zu dieser Einstellung finden Sie im Datenblatt des Intel 82527 CAN Controllers.

Formel zur Berechnung der CAN Bus Frequenz (aus Intel 82527 CAN Controller Datenblatt):

$$\text{CAN bus frequency} = 10 \text{ MHz} / [(\text{BRP} + 1) \times (3 + \text{TSEG1} + \text{TSEG2})]$$

10.4.3 Globals (CAN-IO Device)

Mit diesem CAN-IO Device kann sehr einfach ein CAN Bus Teilnehmer nachgebildet werden, der CAN Botschaften empfangen und senden kann.

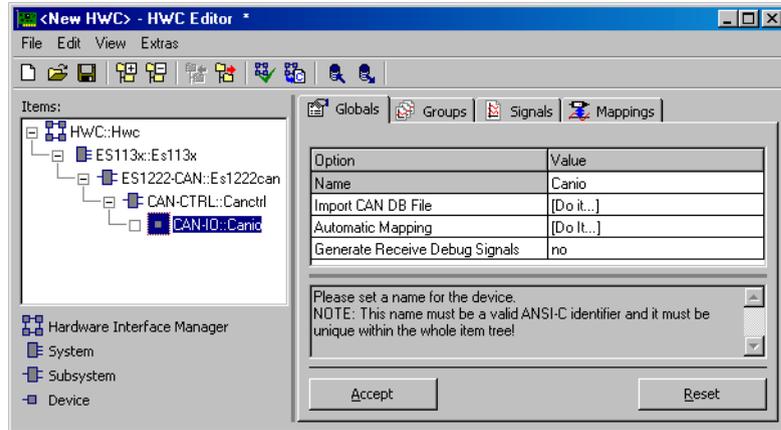
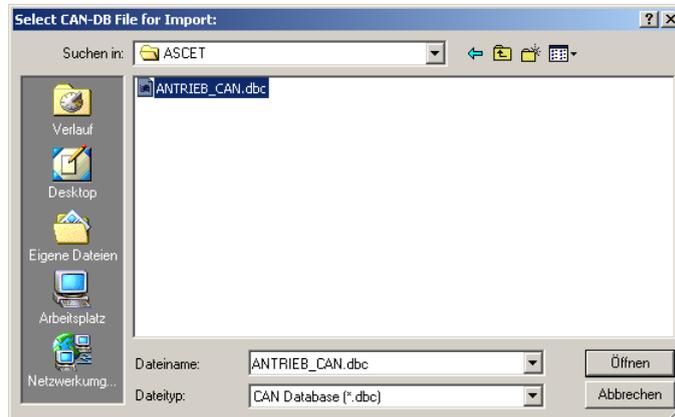


Abb. 10-14 Die Registerkarte „Globals“ des CAN-IO Devices

Import CAN DB File

Über diese Option kann eine CAN Datenbank-Datei eingelesen werden, welche mit dem CANdb Datenverwaltungsprogramm der Firma Vector Informatik erstellt wurde. Mit Hilfe dieser Datei können nun bei Bedarf automatisch CAN-Botschaften und -Signale angelegt werden.

Nach Drücken von [Do it . . .] öffnet sich folgendes Dialogfenster:



In diesem Dialogfenster kann die gewünschte zu importierende CAN DB-Datei ausgewählt werden.

Nach Betätigen der Schaltfläche **Open** öffnet sich nachfolgendes Dialogfenster, in dem das weitere Verfahren spezifiziert werden kann:

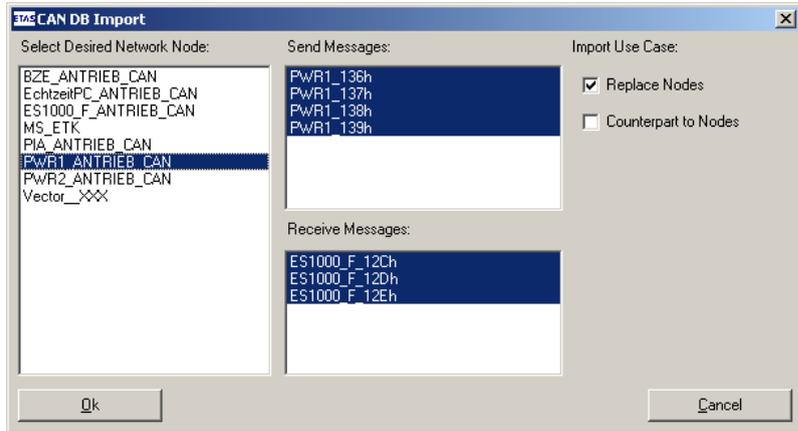


Abb. 10-15 CAN DB Import Dialog

In einer CAN DB Datei werden üblicherweise mehrere Knoten eines CAN-Netzwerkes beschrieben. Alle vorhandenen Knoten werden in der Liste „Select Desired Network Nodes“ aufgelistet. Die beiden Listen rechts davon („Send Messages“ / „Receive Messages“) listen alle für den aktuell selektierten Knoten definierten CAN-Botschaften auf. Die Botschaften, die selektiert sind, werden für den Import herangezogen. Zusätzlich kann der „Import Use Case“ noch ausgewählt werden. Die Option **Replace Nodes** bedeutet, dass das CAN-IO Device die Rolle des Netzwerkknotens übernimmt; d.h. eine Sendebotschaft des Knotens wird auch als Sendebotschaft des CAN-IO Devices umgesetzt usw.

Die Option **Counterpart to Nodes** bedeutet, dass das CAN-IO Device das Gegenstück zum Netzwerkknoten bildet; d.h. eine Sendebotschaft des Knotens ruft eine Receive-Botschaft hervor usw.

Nach Drücken der Schaltfläche **OK** erfolgt eine Überprüfung der zu importierenden Daten (CAN-Messages und -Signale) mit den vorhandenen Signalgruppen und Signalen. Das Resultat dieser Überprüfung wird anschließend in dem nachfolgenden Dialogfenster angezeigt:

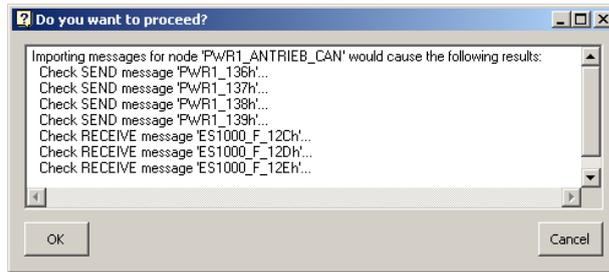


Abb. 10-16 Dialogfenster „CAN DB Check“

Bis jetzt ist noch nichts an dem vorhandenen CAN-IO Device verändert worden. Erst nach Anklicken der Schaltfläche **OK** wird der tatsächliche Importvorgang gestartet.

Der Import-Vorgang fügt die importierten Messages den Signalgruppen hinzu und sorgt auch für die Definition der entsprechenden Signale.

Nach Beendigung des Imports erscheint im Fenster „Monitor“ das detaillierte Protokoll des Importvorgangs:

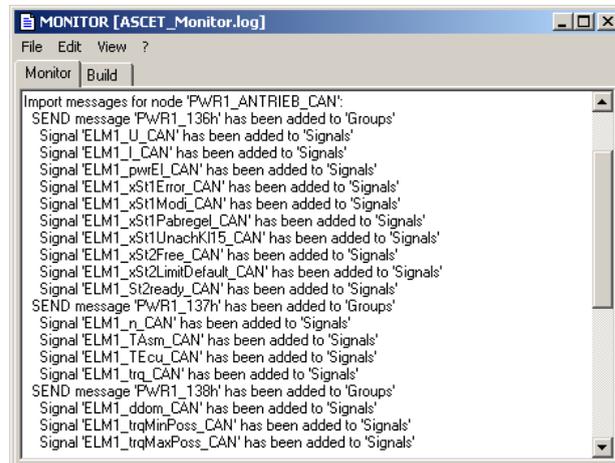


Abb. 10-17 Fenster „Monitor“ mit Importprotokoll

Beim Import beachten :

Beim Importieren einer CAN DB-Datei werden nur 29 Bit (Identifizier *extended*) bzw. 11 Bit (Identifizier *standard*; s. „Identifizier“ auf Seite 167) automatisch in das Identifizier-Feld eingefügt.

Wenn im CAN-Controller Item der Identifier *standard* gewählt wurde, die CAN DB-Datei aber Signale mit 29 Bit-Identifiern ($ID > 2^{31}$) enthält, geschieht folgendes:

- 11 Bit (Bits [28...18]) werden automatisch in das Identifier-Feld geschrieben. Die übrigen Bits (MSB) werden verworfen.
- Im Monitorfenster werden Warnungen angezeigt.

Da nur ein Identifier ausgewählt werden kann, kommt es bei CAN DB-Dateien, die sowohl *standard* als auch *extended* Identifier enthalten, zu Konflikten.

Automatic Mapping

Diese Option ermöglicht die automatische Zuordnung zwischen Signalen und (ASCET-)Messages.

Nach Drücken von [Do it . . .] öffnet sich folgendes Dialogfenster, welches das gegenwärtige Mapping aufzeigt:

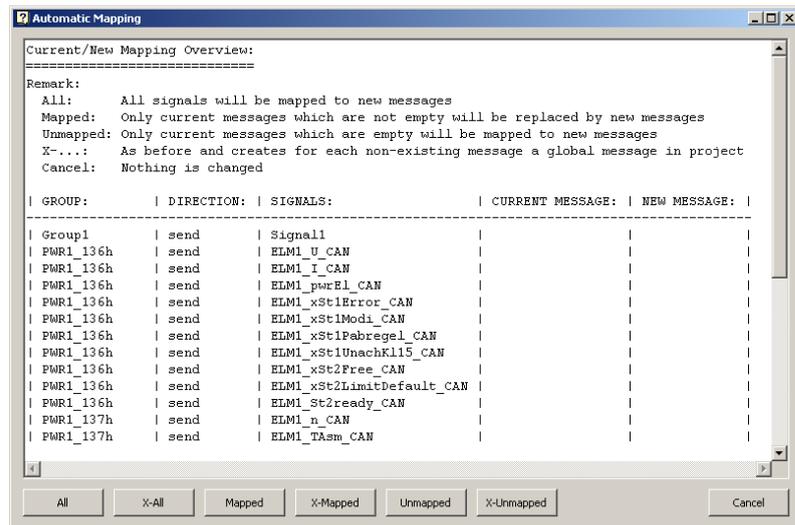


Abb. 10-18 Dialogfenster „Automatic Mapping“ mit Mappingstatus

Vom Fenster aus kann nun je nach Bedarf das weitere Vorgehen ausgewählt werden. Folgende Möglichkeiten stehen zur Verfügung:

- **All:**
Das vorhandene Mapping wird komplett ersetzt. Dabei wird so vorgegangen, dass für jedes Signal eine geeignete namensgleiche ASCET-Message gesucht wird. Ist eine vorhanden, so wird sie eingetragen; falls nicht, so wird das Signal nicht abgebildet.
- **X-All:**
Funktioniert wie „All“, nur wird bei dieser erweiterten Funktionalität zusätzlich für jede fehlende ASCET-Message eine globale Send-Receive Message angelegt. D.h. nach dieser Aktion sind grundsätzlich alle Signale abgebildet.
- **Mapped:**
Es werden nur die Signale neu abgebildet, die bereits abgebildet waren, d.h. für jedes abgebildete Signal wird eine namensgleiche ASCET-Message gesucht. Wird keine gefunden, so wird das entsprechende Signal nicht mehr abgebildet (= leere ASCET-Message).
- **X-Mapped:**
Funktioniert wie „Mapped“, nur werden bei dieser erweiterten Aktion ggf. neue globale Send-Receive Messages angelegt, falls keine namensgleichen ASCET-Messages gefunden wurden.
- **Unmapped:**
Es werden nur die Signale neu abgebildet, die bisher nicht abgebildet waren (= leere ASCET-Message), d.h. für jedes dieser Signale wird eine namensgleiche ASCET-Message gesucht. Wird keine gefunden, so bleibt das entsprechende Signal nicht abgebildet.
- **X-Unmapped:**
Funktioniert wie „Unmapped“, nur werden mit dieser erweiterten Aktion ggf. neue globale Send-Receive-Messages angelegt, falls keine namensgleiche ASCET-Message gefunden wurden.
- **Cancel:**
Bricht die automatische Zuordnung ab.

Ob die mit **X-*** generierten Messages im Projekt oder in einem der zum Projekt gehörigen Module generiert werden, legen Sie im Register „Mapping“ im Optionsfenster des HWC-Editors (s. Seite 108) fest.

Generate Receive Debug Signals

Ist diese Option eingeschaltet (= **yes**), so wird für jede „receive“-Signalgruppe zwei zusätzliche Signale generiert.

- `<GroupName>_Diag_dT`
- `<GroupName>_Diag_Rec`

Das Signal `...dT` gibt die Differenz in Sekunden zur vorangegangenen empfangenen Botschaft an.

Hinweis

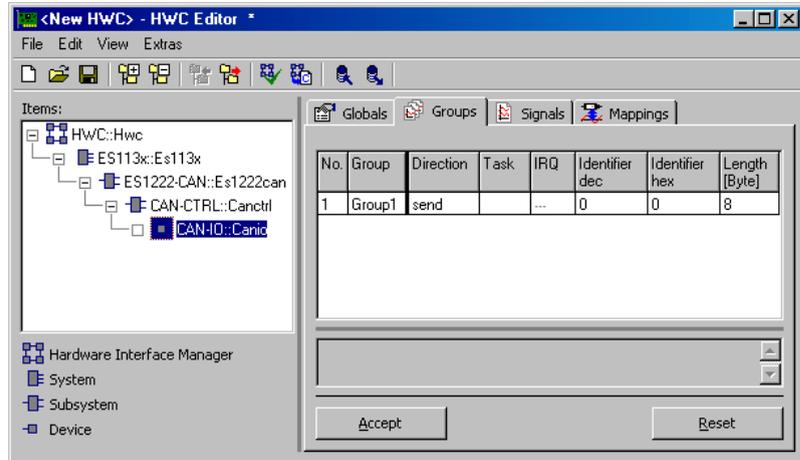
Dieses Signal kann beim normalen Botschaftsempfang (`IRQ = no`) verwendet werden, um den Empfang zu überwachen. Wird z.B. der CANbus unterbrochen, so erhöht sich der Wert laufend im Raster der Empfangstask. Aus Performance-Gründen ist dieser Wert nicht gegen Überlauf geschützt, der bei ca. 300 s auftritt!

Im Gegensatz hierzu kann beim Botschaftsempfang mit Interrupt (`IRQ = yes`) keine echte Empfangsüberwachung durchgeführt werden, da die Berechnung des Wertes erst in der Interrupt-Task erfolgt. Da beim Ausbleiben des Empfangs diese Berechnung auch nicht stattfindet, bleibt somit der alte Wert eingefroren. Damit kann beim Interrupt-Empfang nicht unterschieden werden, ob der Empfang in einem exakt regelmäßigen Raster erfolgt oder unterbrochen ist.

Das Signal `...Rec` wird bei jedem Botschaftsempfang mit dem Wert `true` beschrieben. Wird dieses Signal auf eine Send-Receive-Message abgebildet und diese von der Anwendung nach jedem Lesen auf `false` zurückgesetzt, so kann die Anwendung sehr leicht feststellen, ob zwischen den Rechenzyklen ein Botschaftsempfang stattgefunden hat.

10.4.4 Groups (CAN-IO Device)

In dieser CAN-IO Registerkarte werden die CAN-Botschaften in Form von Signalgruppen spezifiziert.



Hinweis

Neue Signalgruppen bzw. CAN-Botschaften kann man über das Kontextmenü im Register „Groups“ anlegen (siehe hierzu Abschnitt Menü „View“ auf Seite 104).

Hinweis

Bei Änderung des Signalgruppennamens, des Signalnamens oder der Signalrichtung ('Direction') wird die die eventuell zugeordnete ASCET-Message u.U. nicht mehr automatisch zugeordnet und muss manuell neu zugeordnet werden.

Direction

Hier kann die Richtung der CAN-Botschaft bestimmt werden („send“ = Sendebotschaft, „receive“ = Empfangsbotschaft).

Task

Hier wird die Task angegeben, in welcher die Botschaft gesendet, bzw. empfangen werden soll. Soll eine Empfangsbotschaft im Interrupt-Betrieb empfangen werden, so wird diese Einstellung zurückgesetzt und gesperrt.

IRQ

Diese Option gibt an, ob die betreffende Empfangsbotschaft im Interrupt-Betrieb empfangen werden soll oder nicht. Für den „normalen“ Empfang muss die CAN-Botschaft in einer Task mindestens so oft „gepollt“ werden wie sie von der Gegenstation versendet werden kann. Problematisch sind aber bei dieser Betriebsart CAN-Botschaften, die in keinem festen Raster versendet werden oder sogar nur sporadisch auftauchen. Hierfür ist der Interrupt-Empfang ideal geeignet, da er eine Botschaftsverarbeitung genau dann anstößt, wenn die Botschaft empfangen wurde.

Identifizier dec/hex

Hier muss der Botschafts-Identifizier eingegeben werden. Je nach gewählter Identifiertyp-Einstellung in dem darüberliegenden CAN-Controller Item (`standard` oder `extended`) kann hier der Wert unterschiedlich groß werden:

Standard identifizier:	11 Bit	2 047 dec	7 FF hex
Extended identifizier:	29 Bit	536 870 911 dec	F FF FF FF hex

Jede Signalgruppe bzw. CAN-Botschaft muss einen anderen Identifizier besitzen.

Length [Byte]

Legt fest, wieviel Nutzdatenbytes die betreffende Botschaft übertragen kann (1..8).

*Activated Task**

Für „receive“-CAN-Botschaften¹ (Signalgruppen) kann hier eine Software-Task angegeben werden, die immer dann aktiviert wird, wenn die relevante Signalgruppe empfangen und die Empfangstask ausgeführt wurde. Die hier eingegebene Task kann z.B. zur Nachbearbeitung verwendet werden.

Diese Spalte ist standardmäßig nicht eingeblendet.

*Prescaler**

Hier kann bei einer im Interrupt-Betrieb empfangenen Botschaft eingestellt werden, ab wann tatsächlich ein VMEbus-Interrupt ausgelöst wird, um die Daten zu übertragen. Die Standardeinstellung „1“ besagt, dass mit jeder empfangenen Botschaft auch eine Datenübertragung stattfindet. Wird dieser Wert

¹. Es ist möglich, eine solche Task auch für „send“-Botschaften zu spezifizieren, auch wenn es keine empfohlene Anwendung für diese Verwendung gibt. (Da der Sendeprozess asynchron abläuft, ist nicht sichergestellt, ob die CAN-Botschaft gesendet ist, wenn die Task aktiviert wird.)

z.B. auf „2“ erhöht, so wird der VMEbus-Interrupt erst mit jeder zweiten empfangenen Botschaft ausgelöst, welcher dann die Daten der zuletzt empfangenen Botschaft überträgt.

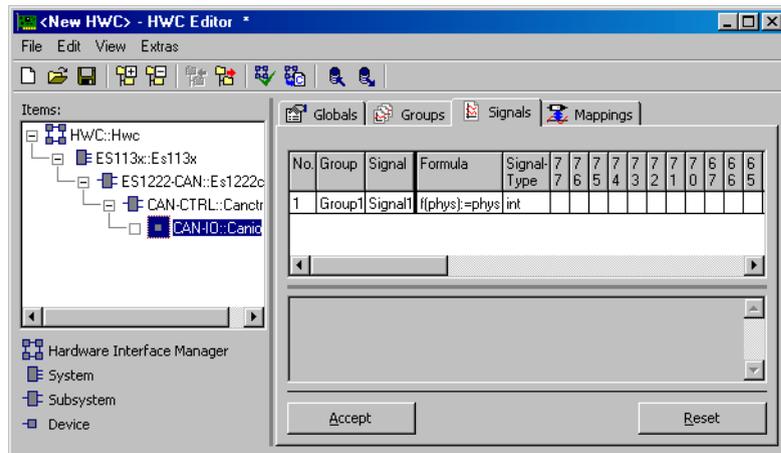
Diese Spalte ist standardmäßig nicht eingeblendet.

Hinweis

Ein Vergrößerung dieses Wertes sollte erwogen werden, wenn die entsprechende Botschaft in sehr schneller Folge empfangen wird und damit die VMEbus- Interruptlast zu sehr erhöht.

10.4.5 Signals (CAN-IO Device)

In dieser CAN-IO Registerkarte werden die CAN-Signale spezifiziert.



Hinweis

Neue Signale lassen sich über das Kontextmenü im Register „Signals“ anlegen (siehe hierzu Abschnitt „Menü „View““ auf Seite 104).

Group

Hier erfolgt die Zuordnung eines Signals zu der gewünschten Signalgruppe.

Signal Type

Hier wird der Signaltyp, in welchem das Signal über den CAN-Bus übertragen wird, festgelegt.

Es stehen folgende Einstellungen zur Verfügung:

Signaltyp	Datentyp
int	Kennzeichnet ein vorzeichenbehaftetes Signal im Standard 2er-Komplement Datenformat (max. 32 Bit)
(s)int	Kennzeichnet ein vorzeichenbehaftetes Signal, in dem als höchstwertiges Bit das Vorzeichen und anschließend der Absolutwert des Signals übertragen wird. Ist das Vorzeichen-Bit gesetzt, so handelt es sich dabei um eine negative Zahl (max. 32 Bit)
uint	Kennzeichnet ein vorzeichenloses Signal (max. 32 Bit)
bool	Kennzeichnet ein Boolesches Signal. Hierbei darf in der Bit-Matrix nur ein einzelnes Bit markiert sein.
real	Kennzeichnet ein Fließkommawert im „Standard IEEE Float (4 Byte)“ Format. Hierbei dürfen entsprechend nur 32 Bit in der Bit-Matrix markiert sein.

Eine Übersicht über die IEEE-Fließkomma-Formate finden Sie in folgender Tabelle:

Format	Sign	Exponent	Fraction
Float	1 Bit	8 Bit	24 Bit
Double (nicht unterstützt)	1 Bit	11 Bit	52 Bit

7654321.. (Bitmatrix)

Eine CAN-Botschaft kann bis zu 8 Datenbytes übertragen. In einer Bit-Matrix kann für jedes Signal festgelegt werden, welche Bits dieses Signal benötigt, bzw. belegt (ein Signal = eine Reihe).

Der Aufbau der Spalten ist wie folgt:

7	7	...	0	0	Byte-Nummer
7	6	...	1	0	Bit-Nummer

Bedeutung der Bit-Felder:

Leeres Feld	Das entsprechende Signal verwendet das Bit nicht
Belegtes Feld	Das Signal benötigt dieses Bit an dieser Position
„X“-Feld	Das entsprechende Bit steht zur Datenübertragung nicht zur Verfügung, weil das Signal weniger Nutzdatenbytes hat (siehe Einstellung „Length“ im Register „Groups“).

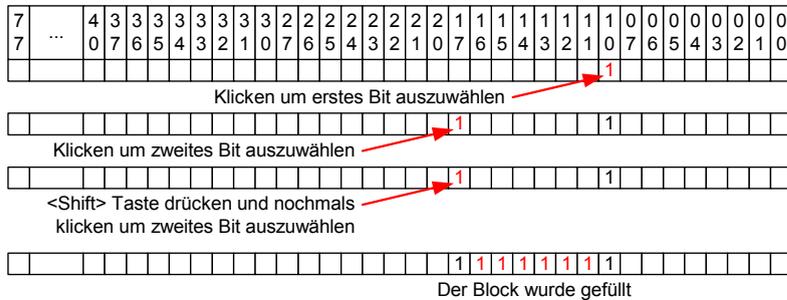
Bedienung der Bit-Felder:

Über die Pfeiltasten der Tastatur kann die gewünschte Bit-Zelle ausgewählt werden.

Mit den Zifferntasten kann dem Bit der richtige Wert zugeordnet werden (verschiedene Werte sind für die „Blockbildung“ wichtig, siehe unten).

Ein Anklicken mit der Maus „toggelt“ die entsprechende Zelle ebenso zwischen „unbelegt“ und „1“. Wird während des Klickens zusätzlich noch die <ALT>-Taste gedrückt, so wird der Wert „1“ bis „9“ inkrementiert (wichtig für „Blockbildung“).

Mehrere Bits gleichzeitig kann man folgendermaßen selektieren:



Mit Hilfe der verschiedenen Zahlen („1111 2222...“) lassen sich Datenblöcke bilden, mit denen praktisch jedes übertragene Signal beschrieben werden kann. Die Zahlen, aus denen die Datenblöcke gebildet werden, haben dabei die Bedeutung, dass der Block mit der höchsten Zahl („2222“) den Block angibt, der bei der Übertragung die höchstwertigsten Bits enthält. Der Block mit der kleinsten Zahl („1111“) enthält bei der Übertragung die niederwertigsten Bits. Mit den zur Verfügung stehenden Zahlen (1...9) lässt sich folglich ein Signal mit bis zu 9 Bit-Blöcken beschreiben. Da die Darstellung der Bits genau der Form entspricht, in der ein Intel-Signal übertragen wird, ist die Blockbildung für die Darstellung von Signalen im Motorola Format notwendig, sobald das Signal eine Länge von 8 Bit überschreitet.

Die Nutzung des CAN Bypass Protokolls im Zusammenwirken von ASCET-RP und Steuergeräten der Robert Bosch GmbH, in denen dieses Protokoll integriert ist, ist ohne weitere Lizenzvereinbarung möglich.

Auskunft über geeignete Steuergeräte mit CBP-Unterstützung erhalten Sie vom Steuergerätehersteller.

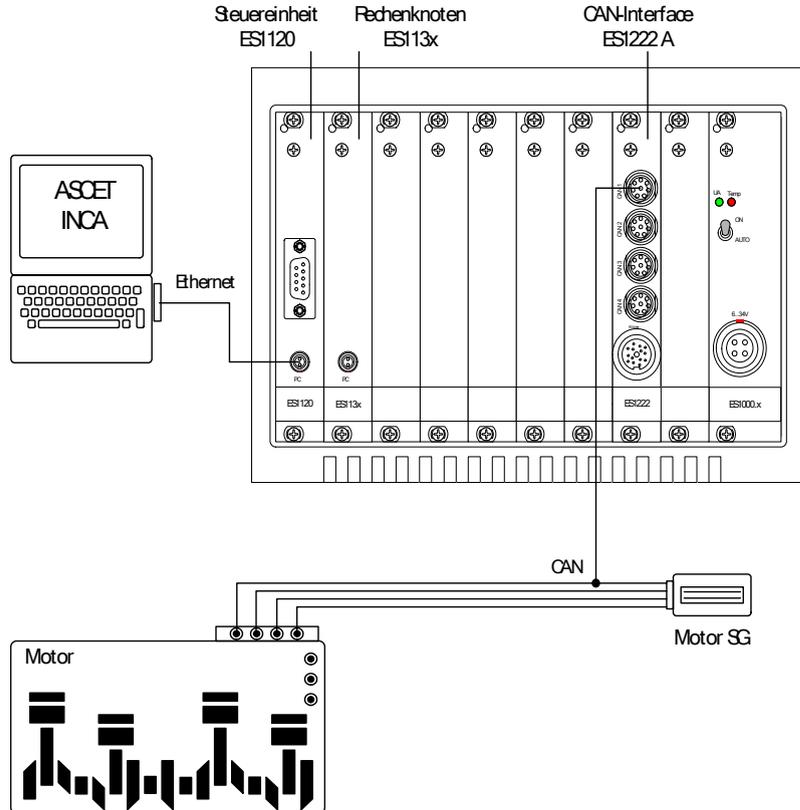
Eine Schnittstellenbeschreibung des CBP in Zusammenhang mit ASCET-RP ist bei der ETAS GmbH auf Anfrage erhältlich.

Hinweis

Die Überlassung der Schnittstellenbeschreibung erfolgt nur, wenn die entsprechende Überlassungsvereinbarung akzeptiert wird. Ein Anrecht auf diese Vereinbarung/Überlassung aufgrund einer ASCET-RP Lizenzvereinbarung besteht nicht!

10.5.2 Hardwarekonfiguration eines CAN-Bypass

Die folgende Abbildung zeigt eine Beispielkonfiguration für eine CAN-Bypass Anwendung mit der ES1000.x:



Wie bei allen Rapid Prototyping-Anwendungen, ist auch hier der Host-PC über die Host-Link Schnittstelle mit dem ETAS Experimentalsystem verbunden. Die mit ASCET entwickelten Funktionen laufen auf dem PPC-Modul ESI13x. Der Anschluss des Steuergeräts an das ETAS Experimentalsystem erfolgt über die CAN-Interface Karte ESI222.

Die Kommunikation mit dem Steuergerät wird beim CAN-Bypass über das CAN-Bypass Protokoll (CBP) abgewickelt (© Copyright Robert Bosch GmbH).

10.5.3 Globals (CAN-Bypass Device)

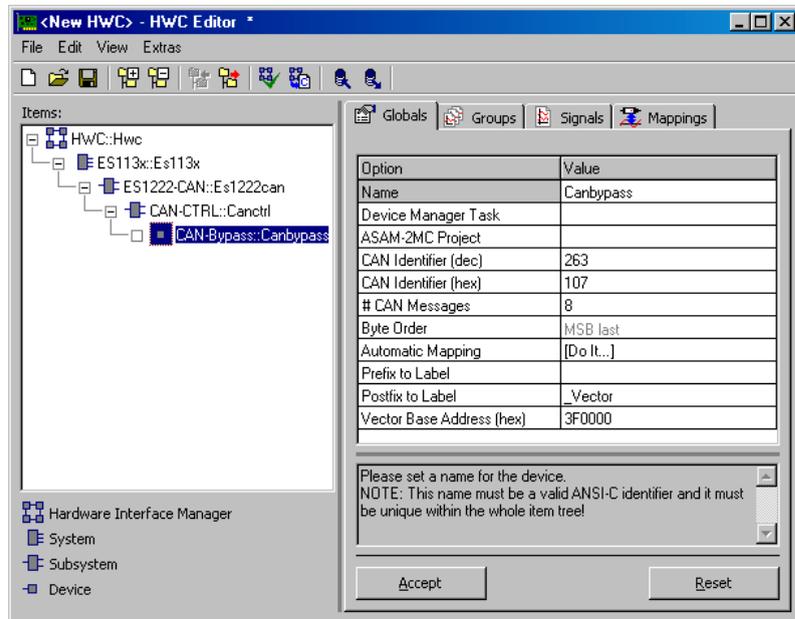


Abb. 10-19 Die Registerkarte „Globals“ des CAN-Bypass Devices

Device Manager Task

Hier muss eine Timer-Task zugeordnet werden, die diesem Bypass exklusiv zur Verfügung steht. Die Periodendauer der Timer Task muss dabei zwischen 0.05 sec und 0.8 sec eingestellt werden.

ASAM-2MC Project

Hier muss ein zum Steuergerät passendes ASAM-MCD-2MC-Projekt ausgewählt werden, welches zuvor in die Datenbank eingelesen wurde.

CAN Identifier (dec)

Hier muss der CAN Identifier der Kommando CAN-Botschaft in dezimaler Schreibweise angegeben werden. Die Kommando CAN-Botschaft ist die Botschaft mit dem höchsten Identifier des CAN-Botschafts Blocks (der Standardwert für CBP ist 263). Die Länge des Identifiers (11 Bit / 29 Bit) hängt von der „Identifier“ Einstellung in der Registerkarte „Globals“ des CAN-CTRL Subsystems ab.

Das Format des CAN-Botschaftsblocks und der Kommandobotschaft ist in der CBP Schnittstellenbeschreibung definiert (siehe Kapitel 10.5.1 auf Seite 180).

CAN Identifier (hex)

Hier kann der CAN Identifier der Kommando CAN-Botschaft in hexadezimaler Schreibweise angegeben werden (der Standardwert für CBP ist 107H).

CAN Messages

Hier muss die Anzahl der für den CAN-Botschafts Block verfügbaren CAN-Botschaften eingestellt werden (4 .. 16). In der Regel besteht beim CAN-Bypass Protokoll der CAN-Botschafts Block aus 8 CAN-Botschaften.

Hinweis

Je höher die Anzahl der CAN Messages, desto mehr Bypass-Größen können übertragen werden. Da die Steuergerätesoftware aber ebenso die Ressourcen bereitstellen muss, kann dieser Eintrag nicht beliebig gesetzt werden, sondern muss mit dem Software-Stand des Steuergeräts abgestimmt sein.

Byte Order

In dieser Zeile wird das Wort-Datenablageformat (MSB first / MSB last) des Steuergeräteprozessors angezeigt. Diese Information wird aus dem zugeordneten ASAM-MCD-2MC-Projekt ausgelesen.

Übersicht über die gebräuchlichen „Byte Order“ Bezeichnungen:

MSB first	big endian	Motorola
MSB last	little endian	Intel

Automatic Mapping

Siehe hierzu die Beschreibung im Kapitel 10.4.3, Abschnitt „Automatic Mapping“ auf Seite 172.

Prefix / Postfix to Label

An diesen Stellen können die Namenserverweiterungen angegeben werden, um von der Bypass-Ausgangsgröße (ASAM-MCD-2MC-Measurement) auf die zugehörigen Bypass-Parameter zu schließen. Diese Namensreferenz dient zur Detektion der zur Verfügung stehenden Bypass-Ausgangsgrößen („send“ Signale zum Steuergerät) und zur Bestimmung aller notwendigen Informationen für ein Bypass-Signal.

Die nachfolgende Abbildung verdeutlicht die Zusammenhänge:

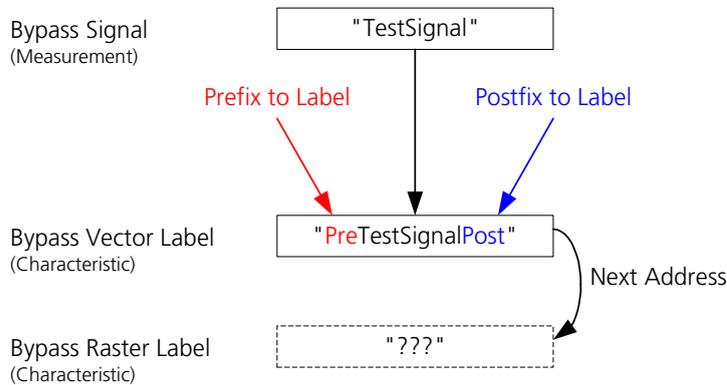


Abb. 10-20 Zusammenhang zwischen den Bypass-Labels

Aufgabe der Bypass Labels:

- Bypass Vector Label Gibt die Vektor-Adresse der Bypass-Größe an. Eine 0 bedeutet, dass die Größe nicht übertragen wird.
- Bypass Raster Label Gibt an, ob die Bypass-Größe im winkelsynchronen Raster (Wert = 0) oder im zeitsynchronen Raster (Wert = 1) übertragen wird.

Vector Base Address (hex)

Hier kann eine Basisadresse angegeben werden, die für die Berechnung der Zieladresse der Bypass-Ausgangsgrößen („send“ Signale zum Steuergerät) im Steuergerät verwendet wird. Dies ist dann notwendig, wenn die Zieladresse außerhalb des Adressbereichs liegt, der über die im CAN-Bypass Protokoll zur Verfügung stehenden 16 Bit hinaus adressiert werden kann.

Die Zieladresse wird folgendermaßen berechnet:

```
<bypass address> :=  
<vector address> + <ecu calibration offset> - <base address>
```

Die Größe <ecu calibration offset> wird über das ASAM-MCD-2MC-Projekt festgelegt.

10.5.4 Groups (CAN-Bypass Device)

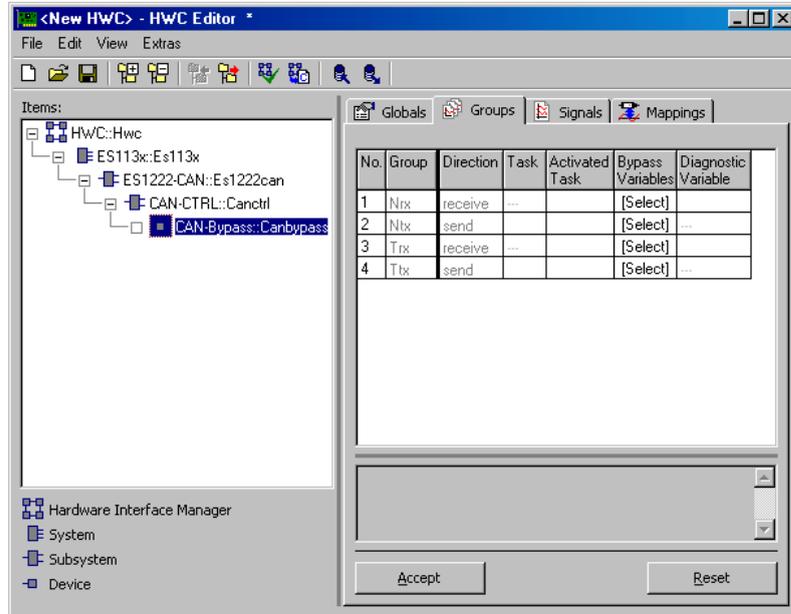


Abb. 10-21 Die Registerkarte „Groups“ des CAN-Bypass Devices

Hinweis

Bei Änderung des Signalgruppennamens, des Signalnamens oder der Signalrichtung ('Direction') wird die die eventuell zugeordnete ASCET-Message u.U. nicht mehr automatisch zugeordnet und muss manuell neu zugeordnet werden.

Group

Das CAN-Bypass Device unterstützt bis zu zwei Raster jeweils in Sende- und Empfangsrichtung. Die Signalgruppen haben folgende Bedeutung:

- Nrx Drehzahlsynchrone Daten vom Steuergerät zum CAN-Bypass Device
- Ntx Drehzahlsynchrone Daten vom CAN-Bypass Device zum Steuergerät
- Trx Zeitsynchrone Daten vom Steuergerät zum CAN-Bypass Device
- Ttx Zeitsynchrone Daten vom CAN-Bypass Device zum Steuergerät

Activated Task

In dieser Spalte müssen üblicherweise Softwaretasks angegeben werden, die durch den Empfang der drehzahl- bzw. zeitsynchronen Daten aktiviert werden sollen. Zweckmäßigerweise wird in diesen Tasks auch die Berechnung der Bypass-Funktionen sowie das Zurücksenden der Ergebnisse an das Steuergerät durchgeführt (vergleiche entsprechende Einträge in der „Task“ Spalte).

Bypass Variables

Durch Selektion einer Zelle dieser Spalte wird der Variablen-Auswahl-Dialog geöffnet, in welchem die Bypass-Größen (Measurements) für die entsprechende Signalgruppe ausgewählt werden kann.

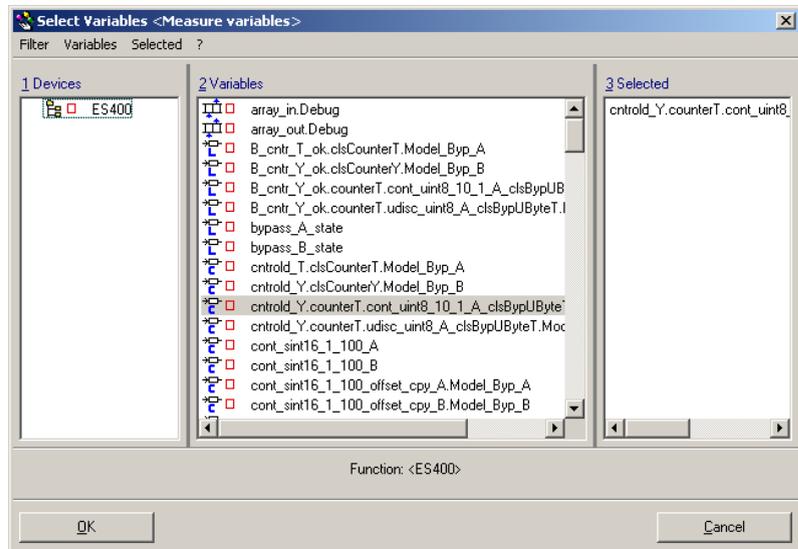


Abb. 10-22 Das Dialogfenster für die Variablen-Auswahl

Hinweis

Nichtlineare Umrechnungsformeln werden nicht unterstützt. Wird dennoch eine Variable mit nichtlinearer Umrechnungsformel ausgewählt, erscheint eine Warnung folgender Art:

<RTIO Toolbox WARNING>

Error on processing signal '<signal_name>' formula: Can't handle '<formula_name>' for bypass; formula is converted to 'Formulald'!

Für die „receive“-Signalgruppen stehen prinzipiell alle im ASAM-MCD-2MC-Projekt definierten Messgrößen zur Verfügung. Für die „send“-Signalgruppen stehen nur jene Messgrößen zur Verfügung, die von der Software des Steuergeräts als Bypass-Eingriffsgrößen unterstützt und von der Namensreferenz als Bypass-Eingriffsgrößen ausgewiesen werden (siehe „Prefix / Postfix to Label“ im „Globals“ Register).

Hinweis

Die Anzahl der Messgrößen ist begrenzt und hängt vom Datentyp der Messgrößen (Byte, Wort...) sowie der Anzahl der zur Verfügung stehenden Bypass CAN-Botschaften ab (Eintrag „# Bypass Messages“ im Register „Globals“).

Diagnostic Variable

Hier kann nach Bedarf für jede „receive“-Signalgruppe, eine zusätzliche Größe des Steuergeräts für Diagnosezwecke ausgewählt werden (z.B. zur Kontrolle des Bypass-Betriebs).

Die Auswahl funktioniert so wie unter „Bypass Variables“ beschrieben.

10.5.5 Signals (CAN-Bypass Device)

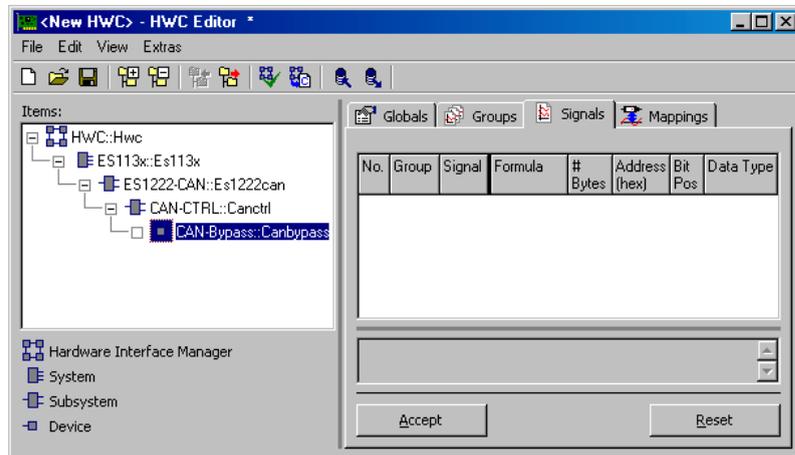


Abb. 10-23 Die Registerkarte „Signals“ des CAN-Bypass Devices

Hinweis

Keine der Spalten der Registerkarte „Signals“ ist vom Anwender editierbar. Sie dienen ausschließlich zur Anzeige von Statuswerten für die Bypassgrößen.

Die Sortierung der Bypassgrößen innerhalb der jeweiligen Signalgruppe erfolgt gemäß ihrer Bytegröße (8-4-2-1 Byte).

Bytes

In dieser Spalte wird die Anzahl der Bytes je Bypassgröße angezeigt.

Address (hex)

In dieser Spalte werden die Adressen der Bypassgrößen angezeigt.

Empfangsgrößen haben einen 32 Bit Adressbereich, Sendegrößen einen Adressbereich von 16 Bit.

Bit Pos

Diese Spalte ist unbenutzt, da die zur Zeit vorherrschende Steuergeräte-Software keinen Bypass von Bitgrößen unterstützt.

Data Type

In dieser Spalte wird der Datentyp der Bypassgrößen angezeigt.

Byte Offset

Anzeige des Offsets der Bypassgröße innerhalb der jeweiligen Signalgruppe.

10.5.6 Mappings (CAN-Bypass Device)

Die Einstellmöglichkeiten sind für alle Items gleich. Sie werden in Kapitel 7.3.5 auf Seite 120 beschrieben.

10.6 ES1223-LIN

Die ES1223-Einschubkarte dient als LIN-Schnittstelle in VMEbus-Systemen. Die Karte stellt vier LIN-Kanäle zur Verfügung. Sie bietet zudem die Möglichkeit, bei empfangenen Botschaften VMEbus-Interrupts auszulösen, so dass ein dauerndes Pollen nach Botschaften entfallen kann. Jeder LIN-Controller kann bis zu 64 Botschaften verarbeiten. Durch Einsatz eines eigenen Prozessors wird der System-Controller wesentlich entlastet.

Dieser Abschnitt beschreibt die RTIO-Einbindung der ES1223-Karte. Im HWC-Editor wird die ES1223-Karte durch Auswahl des Items „ES1223-LIN“ eingebunden.

10.6.1 Globals (ES1223-LIN Subsystem)

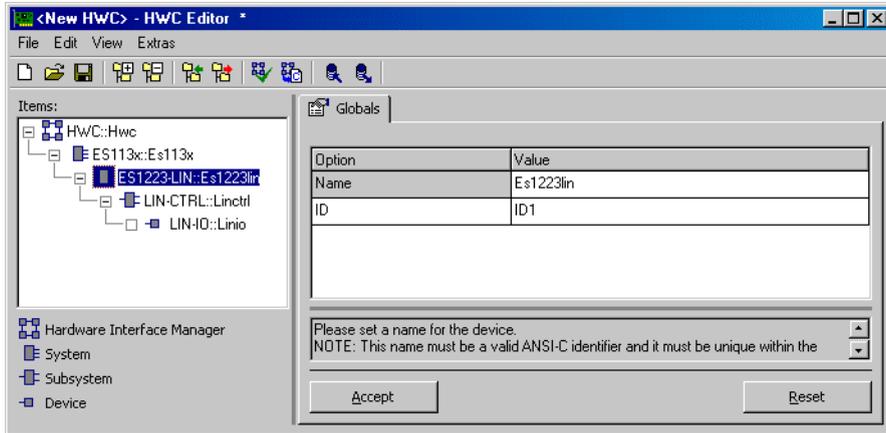


Abb. 10-24 Die Registerkarte „Globals“ des ES1223-LIN Subsystems

ID

Im Auswahlfeld „ID“ muss die Kartennummer von der Karte eingegeben werden, die angesprochen werden soll.

Hinweis

Bei der ES1223.1 Karte handelt es sich um eine sogenannte „Auto-ID“ Karte, die beim Einschalten des Systems vom Hardware-Manager automatisch eine ID-Nummer und einen freien Adressbereich zugewiesen bekommt. Karten vom selben Typ (z.B. zwei ES1223.1) werden von links nach rechts nummeriert, d.h. die linke Karte erhält die Nummer Eins, die nächste die Nummer Zwei. Es stehen 2 IDs zur Verfügung; somit können bis zu 2 ES1223.1 Karten in einem Experimentalsystem betrieben werden.

10.6.2 Globals (LIN-CTRL Subsystem)

In der Registerkarte „Globals“ werden dem LIN-CTRL Subsystem eine Init-Task, eine Exit-Task und eine IRQ Handler-Task zugewiesen.

Dem LIN-CTRL Subsystem können bis zu vier LIN-IO Devices zugeordnet werden, entsprechend den vier LIN-Schnittstellen auf der Einschubkarte.

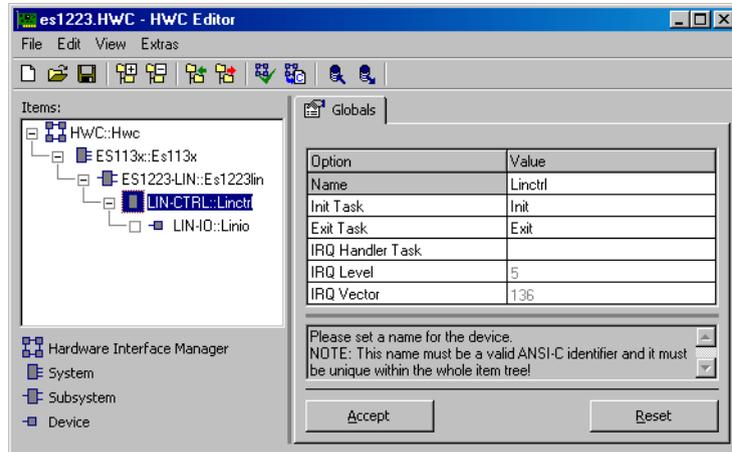


Abb. 10-25 Die Registerkarte „Globals“ des LIN-CTRL Subsystems

IRQ Handler Task

Für die Unterstützung von Interrupts der ES1223-Karte wird eine zusätzliche „IRQ Handler Task“ benötigt. Diese Task muss im ASCET-Projekteditor in der Task-Liste als „Software“-Task angelegt werden, wobei bei „Max. No. of Activations“ mindestens die Anzahl 2 (maximal 50) einzutragen ist.

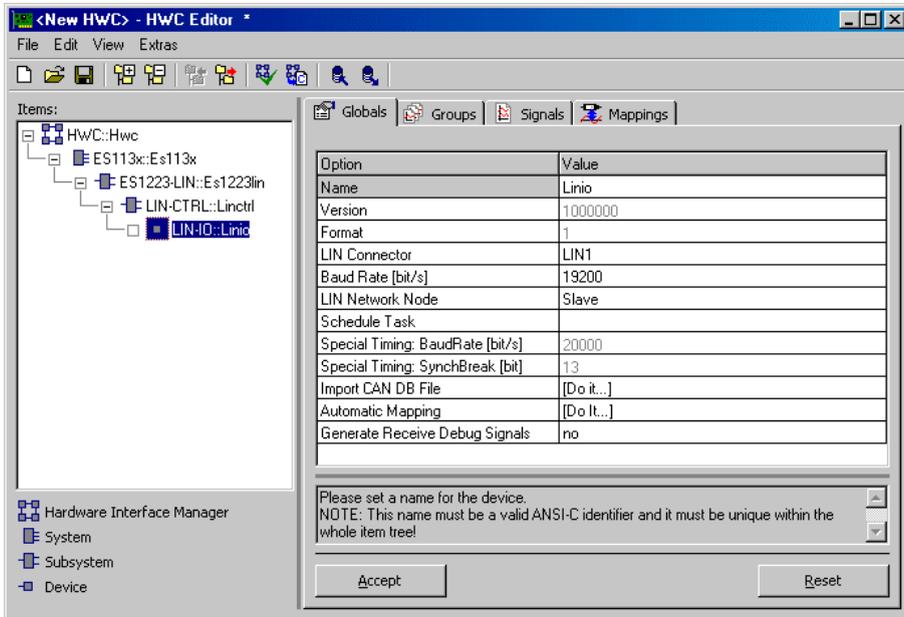


Abb. 10-26 Die Registerkarte „Globals“ des LIN-IO Devices

LIN Connector

Hier wird der gewünschte LIN-Anschluss (LIN1 bis LIN4) ausgewählt. Jedes LIN-IO Device muss einem anderen Anschluss zugeordnet sein.

Baud Rate [bit/s]

Hier kann die gewünschte Baud Rate ausgewählt werden. Es stehen hierfür die Standard-Baudraten (19200, 9600, 2400 bit/s) zur Verfügung. Zusätzlich besteht die Möglichkeit, die Einstellung <Special Timing> zu wählen, welche die Low-Level-Ansteuerung des LIN-Controllers bezüglich Bit-Timing und Baud Rate aktiviert.

In den auf Seite 193 beschriebenen „Special Timing“-Optionen können diese Einstellungen angegeben werden.

LIN Network Node

Mit dieser Option legen Sie fest, ob das gewählte LIN-IO Device als Master- oder Slave-Kontrolleinheit arbeitet. Beachten Sie dabei, dass das LIN-Netzwerk nur eine Master-Einheit haben darf.

Schedule Task

Diese Option ist nur verfügbar, wenn für das gewählte LIN-IO Device die Option „LIN Network Node“ auf `Master` gesetzt ist. Sie wird verwendet, um für eine Master-Kontrolleinheit eine Task zu definieren, die immer dann ausgeführt wird, wenn die Send-Queue leer ist. Dies ist sinnvoll für das kontinuierliche Senden von Botschaften.

*Special Timing: BaudRate [bit/s]**

Diese Option ist standardmäßig ausgeblendet und kann nur bearbeitet werden, wenn die Option „Baud Rate“ auf `<Special Timing>` eingestellt ist.

Hier kann, abweichend von den Standard-Übertragungsraten (siehe „Baud Rate [bit/s]“ auf Seite 192), eine beliebige Baudrate zwischen 10 und 20.000 bit/s eingegeben werden.

*Special Timing: SynchBreak [bit]**

Diese Option ist standardmäßig ausgeblendet und kann nur bearbeitet werden, wenn die Option „Baud Rate“ auf `<Special Timing>` eingestellt ist.

Die Synchbreak-Länge umfasst den eigentlichen Synchbreak (low phase) sowie den Synch-Delimiter (1 Bit). Hier kann, abweichend vom Standard (mindestens 14 Bit), eine beliebige Synchbreak-Länge zwischen 8 Bit und 15 Bit (low time) eingegeben werden.

Import CAN DB File

Über diese Option kann eine CAN/LIN Datenbank-Datei eingelesen werden, welche mit dem CANdb Datenverwaltungsprogramm der Firma Vector Informatik erstellt wurde. Mit Hilfe dieser Datei können nun bei Bedarf automatisch LIN-Botschaften und -Signale angelegt werden.

Wie Sie den Import ausführen, ist im Abschnitt „Import CAN DB File“ auf Seite 169 beschrieben.

Automatic Mapping

Diese Option ermöglicht die automatische Zuordnung zwischen Signalen und (ASCET-)Messages.

Die genaue Vorgehensweise ist im Abschnitt „Automatic Mapping“ auf Seite 172 beschrieben.

Generate Receive Debug Signals

Ist diese Option eingeschaltet (= `yes`), so wird für jede „receive“-Signalgruppe zwei zusätzliche Signale generiert.

- <GroupName>_Diag_dT
- <GroupName>_Diag_Rec

Weitere Informationen finden Sie im Abschnitt „Generate Receive Debug Signals“ auf Seite 173.

10.6.4 Groups (LIN-IO Device)

In dieser LIN-IO Registerkarte werden die LIN-Botschaften in Form von Signalgruppen spezifiziert.

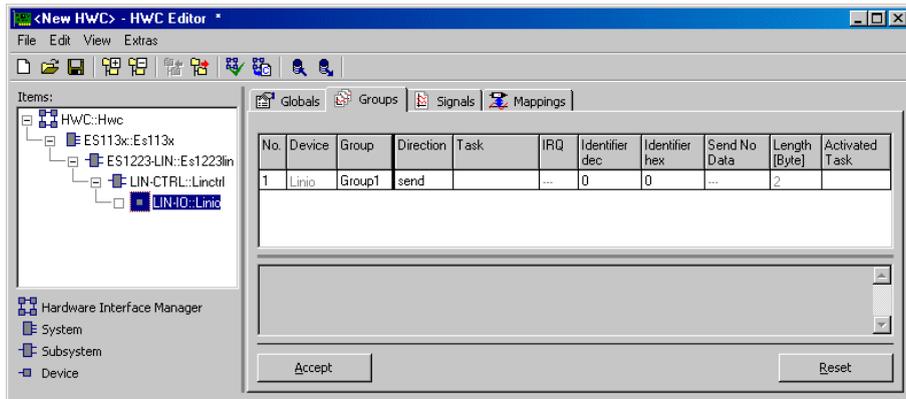


Abb. 10-27 Die Registerkarte „Groups“ des LIN-IO Devices

Neue Signalgruppen bzw. LIN-Botschaften können Sie über das Kontextmenü im Register „Groups“ anlegen (siehe hierzu Abschnitt Menü „View“ auf Seite 104).

Hinweis

Bei Änderung des Signalgruppennamens, des Signalnamens oder der Signalrichtung ('Direction') wird die eventuell zugeordnete ASCET-Message u. U. nicht mehr automatisch zugeordnet und muss manuell neu zugeordnet werden.

Eine Beschreibung der Optionen finden Sie im Kapitel 10.4.4 „Groups (CAN-IO Device)“ auf Seite 175. Das LIN-IO Device weist folgende Besonderheiten auf:

- Die Option „Length [Byte]“ ist beim LIN-IO Device standardmäßig ausgeblendet und schreibgeschützt, da die Länge der LIN-Botschaft im Identifier codiert ist.
- Die Option „Prescaler“ ist nicht vorhanden.

- Die Option „Send No Data“ ist LIN-spezifisch und wird hier beschrieben.

Send No Data

Diese Option ist nur verfügbar, wenn im Register „Globals“ die Option „LIN Network Node“ auf Master gesetzt wurde und im Register „Groups“ in der Spalte „Direction“ send ausgewählt ist.

Wird „Send No Data“ auf yes gesetzt, wird nur der Header der Botschaft übertragen, nicht die Daten. Die Länge der Botschaft ist damit Length [Byte] = 0.

Hinweis

Um mit einem Master-Knoten Daten von einem Slave-Knoten zu empfangen, müssen zwei Signalgruppen **mit demselben Identifizier** angelegt werden:

- eine Signalgruppe mit der Richtung „send“ und der Option „send no data“,
- eine Signalgruppe mit der Richtung „receive“.

Mit der ersten Signalgruppe sendet der Master-Knoten den Header der Botschaft, mit der zweiten empfängt er die Daten des Slave-Knotens.

10.6.5 Signals (LIN-IO Device)

In dieser LIN-IO Registerkarte werden die LIN-Signale spezifiziert.

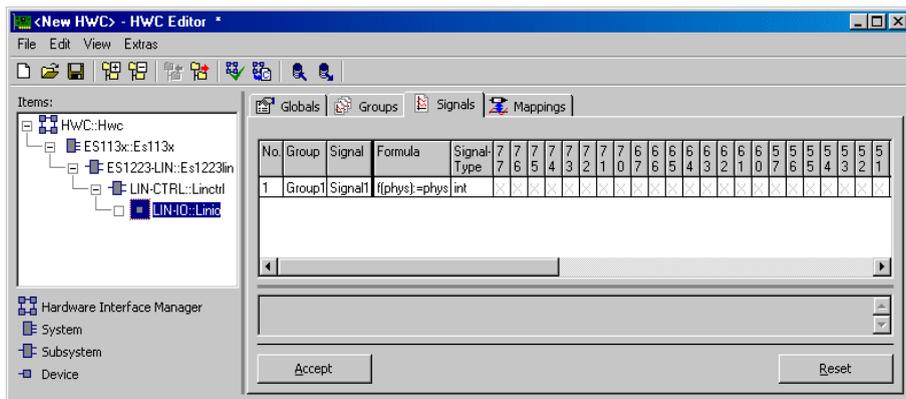


Abb. 10-28 Die Registerkarte „Signals“ des LIN-IO Devices

Neue Signale lassen sich über das Kontextmenü im Register „Signals“ anlegen (siehe hierzu Abschnitt „Menü „View““ auf Seite 104).

Eine Beschreibung der Spalten der Tabelle finden Sie im Kapitel 10.4.5 „Signals (CAN-IO Device)“ auf Seite 177.

Hinweis

Die Länge einer Botschaft wird automatisch im Identifizier (Register „Groups“) codiert. Die obersten 3 Bit des Identifiers geben die Länge an - möglich sind 2, 4, und 8 Byte. Daher sind alle höheren Felder der Bitmatrix deaktiviert.

10.6.6 Mappings (LIN-IO Device)

Die Einstellmöglichkeiten sind für alle Items gleich. Sie werden in Kapitel 7.3.5 auf Seite 120 beschrieben.

10.6.7 Laufzeitverhalten

Der LIN-Bus wird vom Prozessor der ES1223 im Interruptmodus verwaltet. Aufgrund der Interruptverwaltung können Sende- bzw. Empfangsbefehle um bis zu 250 µs verzögert werden.

10.7 ES1231.1-ETK

Die ETK-Interface Karte ES1231 ist der Nachfolger der ES1200-Karte und dient ebenfalls zur Ankopplung eines Steuergerätes mit ETK an ein Experimentalsystem.

Die Karte weist gegenüber ihrem Vorgänger erweiterte Leistungsmerkmale wie Blocktransfermodus und eine höhere Übertragungsrate auf, mit dem bei entsprechender Unterstützung eine erheblich höhere Datenrate erreicht werden kann. Darüber hinaus ist diese Karte mit der speichersparenden „Auto-ID“ Technologie ausgerüstet.

Hinweis

Die ES1231 Karte benötigt zur Kommunikation mit einem ETK zusätzlich noch spezielle Systemdienste, die nur von einer ES1120 (VCU2) Systemcontroller Karte bereitgestellt werden kann. (Siehe hierzu auch Abschnitt „Hardware – Experimentalsystem ES1000.x“ auf Seite 71)

10.7.1 Globals (ES1231-ETK Subsystem)

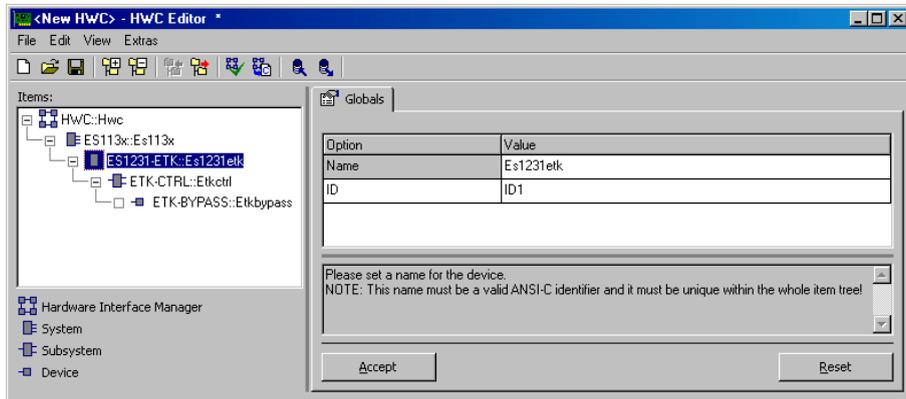


Abb. 10-29 Die Registerkarte „Globals“ des ES1231-ETK Items

ID

Im Auswahlfeld „ID“ muss die Kartennummer von der Karte eingegeben werden, die angesprochen werden soll.

Hinweis

Bei der ES1231 handelt es sich um eine sogenannte „Auto-ID“ Karte, die beim Einschalten des Systems vom Hardware-Manager automatisch eine ID-Nummer und einen freien Adressbereich zugewiesen bekommt. Karten vom selben Typ (z.B. zwei ES1231) werden von links nach rechts nummeriert, d.h. die linke Karte erhält die Nummer Eins, die nächste die Nummer Zwei usw. Es stehen 4 IDs zur Verfügung; somit können bis zu 4 Karten in einem Experimentalsystem betrieben werden.

Da die ES1231 nur einen ETK-Controller aufweist, darf hierfür nur ein „ETK-CTRL“ Item eingefügt werden (nur Port A wird unterstützt).

10.7.2 Globals (ETK-CTRL Subsystem)

Siehe Kapitel 10.3.2 „Globals (ETK-CTRL Subsystem)“ auf Seite 147.

10.7.3 Globals (ETK-BYPASS Device)

Siehe Kapitel 10.3.3 „Globals (ETK-BYPASS Device)“ auf Seite 149.

10.7.4 Groups (ETK-BYPASS Device)

Siehe Kapitel 10.3.4 „Groups (ETK-BYPASS Device)“ auf Seite 158.

10.7.5 Signals (ETK-BYPASS Device)

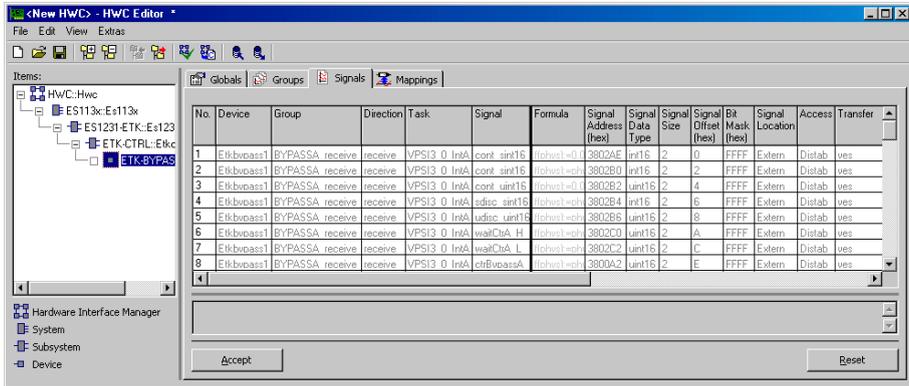


Abb. 10-30 Die Registerkarte „Signals“ des ETK-BYPASS Devices

Hinweis

Keine der Spalten der Registerkarte „Signals“ ist vom Anwender editierbar. Sie dienen ausschließlich zur Anzeige von Statuswerten für die Bypassgrößen.

Die Registerkarte „Signals“ entspricht bis auf zwei zusätzliche Spalten der Beschreibung in Kapitel 10.3.5 „Signals (ETK-BYPASS Device)“ auf Seite 161. An dieser Stelle werden nur die neuen Spalten beschrieben.

Bit Mask (hex)

Die ES1231 liest immer ein Byte (8 Bit) oder ein Wort (16 Bit) vom ETK-Controller. ASAM-MCD-2MC und der HWC-Editor ermöglichen dennoch die Verwendung von Bit-Signalen. Die Spalte „Bit Mask (hex)“ enthält die Maske, anhand derer aus dem vom Controller gelesenen Wert das Signal bestimmt wird (bitweises AND mit der Maske).

Im Prinzip können so mit den entsprechenden Masken (0x1, 0x2, 0x4, 0x8, 0x10, 0x20, 0x40, 0x80) aus einem Byte-Wert acht Signale erzeugt werden. Bei FAR-Adressierung werden aus einem Wort-Wert zwei Byte-Werte erzeugt (Masken 0x00FF und 0xFF00).

Die Spalte ist standardmäßig ausgeblendet.

Transfer

Wenn aus einem Byte-Wert mehrere Signale erzeugt werden, merkt das System, dass der Wert bereits gelesen wurde. Durch die Verwendung einer lokalen Kopie bei weiteren Zugriffen kann Übertragungszeit gespart werden. Um diese Möglichkeit zu nutzen, wird beim ersten Zugriff auf eine Adresse in der Spalte „Transfer“ \forall es angezeigt, bei allen weiteren Zugriffen auf dieselbe Adresse No.

Die Spalte ist standardmäßig ausgeblendet.

10.7.6 Mappings (ETK-BYPASS Device)

Die Einstellmöglichkeiten sind für alle Items gleich. Sie werden in Kapitel 7.3.5 auf Seite 120 beschrieben.

10.8 ES1232-ETK

Die ETK-Interface Karte ES1232 ist der Nachfolger der ES1231-Karte und dient ebenfalls zur Ankopplung eines Steuergerätes mit ETK an ein Experimentalsystem.

Die Karte weist gegenüber ihrem Vorgänger erweiterte Leistungsmerkmale auf: indirekte Adressierung, Übertragungsraten von 8 MBit/s und 100 MBit/s, die Verwendung von bis zu 32 Messrastern, Einlesen neuer ASAM-MCD-2MC-Dateien (AML V1.2.0; ETK-CTRL-ADV Subsystem, s. Kapitel 10.8.5) und die Unterstützung serieller ETKs.

10.8.1 Globals (ES1232-ETK Subsystem)

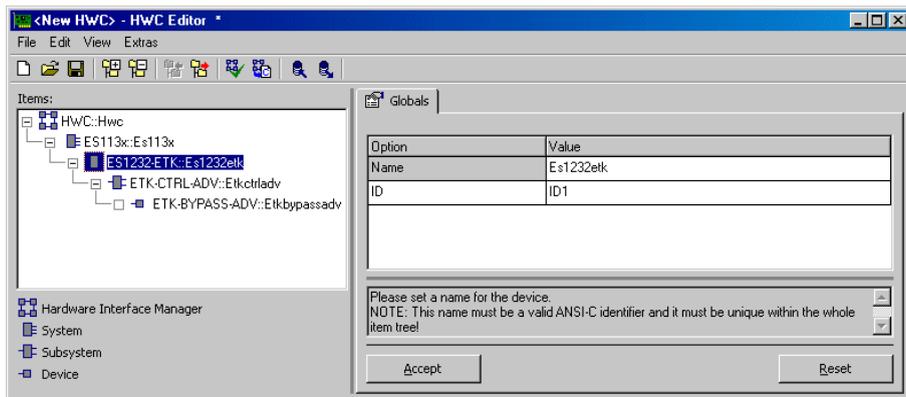


Abb. 10-31 Die Registerkarte „Globals“ des ES1232-ETK Items

Im Auswahlfeld „ID“ muss die Kartenummer der Karte eingegeben werden, die angesprochen werden soll. Die Kartenummer gibt gleichzeitig die Position der Karte im VME-System an; ID1 ist die erste ES1232 von links, ID2 die zweite, usw.

Hinweis

Bei der ES1232 handelt es sich um eine sogenannte „Auto-ID“ Karte, welche beim Einschalten des Systems vom Hardware-Manager automatisch eine ID-Nummer und einen freien Adressbereich zugewiesen bekommt. Karten vom selben Typ (z.B. zwei ES1232) werden von links nach rechts nummeriert, d.h. die linke Karte erhält die Nummer Eins, die nächste die Nummer Zwei usw. Es stehen 4 IDs zur Verfügung; somit können bis zu 4 Karten in einem Experimentalsystem betrieben werden.

Da die ES1232 nur einen ETK-Controller aufweist, kann hierfür nur ein „ETK-CTRL“ Item eingefügt werden (nur Port A wird unterstützt).

10.8.2 ETK-CTRL-BAS Subsystem

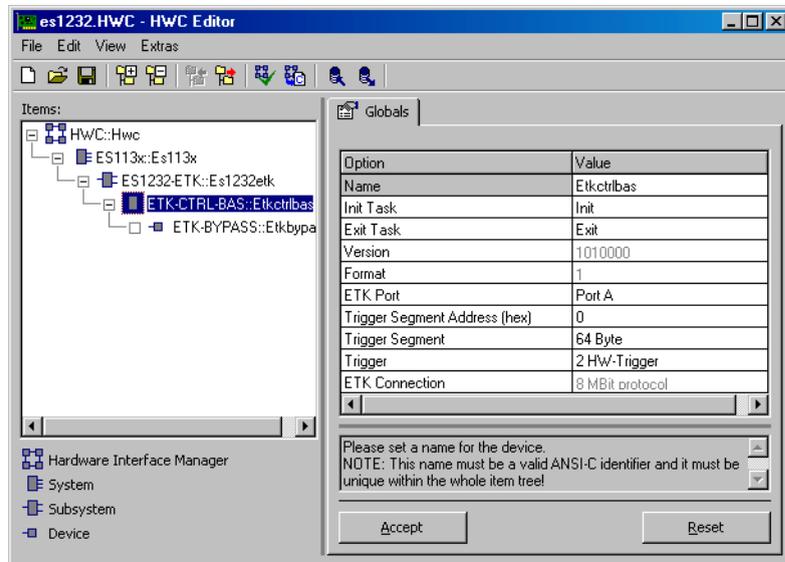


Abb. 10-32 Die Registerkarte „Globals“ des ETK-CTRL-BAS Subsystems

Die Einstellungen des ETK-CTRL-BAS Subsystems entsprechen mit Ausnahme der Zeile „ETK Connection“ dem Item ETK-CTRL der ES1231-ETK Karte (Kapitel „ES1231.1-ETK“ auf Seite 196).

ETK Connection

In dieser Zeile wird die Übertragungsrate (8 MBit/s oder 100 MBit/s) angezeigt. Das Feld kann nicht bearbeitet werden; der Wert wird aus dem ASAM-MCD-2MC-Projekt übernommen (siehe Seite 207).

10.8.3 ETK-BYPASS Device

Die Einstellungen des zum ETK-CTRL-BAS Subsystem gehörigen ETK-BYPASS Device entsprechen in Struktur und Funktionalität dem Item ETK-BYPASS der ES1231-ETK Karte (Kapitel „ES1231.1-ETK“ auf Seite 196).

10.8.4 100 Mbit/s für existierende Projekte (ETK-CTRL-BAS Subsystem)

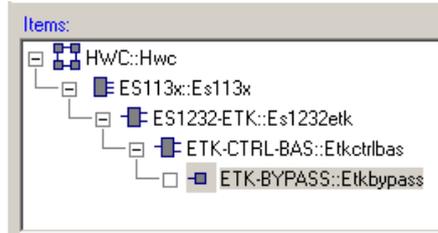
Es ist möglich, existierende Projekte (AML V1.1.0) mit der ES1232 und dem ETK-CTRL-BAS Subsystem zu verwenden.

Die Übertragungsrate wird in der ASAM-MCD-2MC-Datei (*.a21) im TP_BLOB festgelegt. Dazu gibt es ab der Version AML1.1.1 den Parameter `INTERFACE_SPEED`.

Wenn Sie das alte Projekt wie bisher benutzen wollen, ohne die hohe Übertragungsrate von 100 MBit/s zu verwenden, ist keine Änderung der ASAM-MCD-2MC-Datei erforderlich. Gehen Sie wie folgt vor.

Existierendes Projekt mit 8 MBit/s verwenden:

- Legen Sie im HWC-Editor die nötige Item-Struktur an (siehe hierzu Kapitel 7.2 und Kapitel 11.3.3).



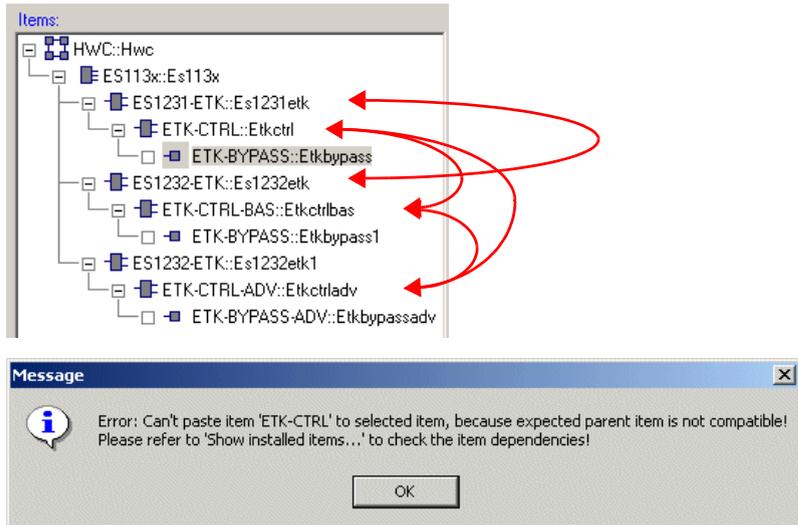
Sie können ggf. mit **Edit** → **Copy** → **Item(s)** und **Edit** → **Paste** → **to selected item** das vorhandene ETK-BYPASS Device der ES1231 mit allen Einstellungen zur ES1232 Karte kopieren.

Sie können mit **Edit** → **Copy** → **Item Data** und **Edit** → **Paste** → **Data to selected item** die Daten des ETK-BYPASS Device der ES1231 zur ES1232 Karte kopieren.

Hinweis

Nur das **ETK-BYPASS Item oder dessen Daten** können kopiert werden, und zwar nur **zwischen** der **ES1231** und der **ES1232 mit ETK-CTRL-BAS** Subsystem.

Alle übrigen Versuche, Items oder Daten zu kopieren (siehe nachstehende Abbildung), werden mit einer entsprechenden Fehlermeldung abgebrochen.



- Wählen Sie im Register „Globals“ des ETK-BYPASS Device (ES1232 / ETK-CTRL-BAS) in der Zeile „ASAM-2MC Project“ die ASAM-MCD-2MC-Datei (siehe „ASAM-2MC Project“ auf Seite 150).

Im ASCET-Monitorfenster erscheint folgende Warnung:

Parameter 'Interface Speed' is missing in A2L/TP_BLOB: old TP_BLOB version 0x1000001. Using default: 8 MBit

Im Register „Globals“ des ETK-CTRL-BAS Items wird in der Zeile „ETK Connection“ die Übertragungsrate von 8 MBit/s angezeigt.

Wenn Sie das alte Projekt mit der hohen Übertragungsrate von 100 MBit/s verwenden wollen, müssen Sie neben den Einstellungen im HWC-Editor auch die ASAM-MCD-2MC-Datei anpassen. Gehen Sie wie folgt vor.

Existierendes Projekt mit 100 MBit/s verwenden:

- Legen Sie im HWC-Editor die nötige Item-Struktur an, wie auf Seite 202 beschrieben.

- Öffnen Sie die ASAM-MCD-2MC-Datei in einem Texteditor.

In AML1.1.0 ist der TP_BLOB wie folgt definiert:

```
/begin TP_BLOB 0x1000001 0x0 0x0
  /begin DISTAB_CFG  0xC 0x1 MSB_LAST 0x383000 0x0
    TRG_MOD 0xB7
  /end DISTAB_CFG
CODE_CHK
0x0
0x0
0x0
0x0
ETK_CFG 0xF 0xF0 0xFF 0x3 0xFD 0xEE 0xFF 0x1
RESERVED 0x810000 0x8103F9 EXTERN 0x-1 0x-1 0x-1
        0x-1 0x-1
/end TP_BLOB
```

Zum besseren Verständnis noch einmal die erste Zeile mit Kommentaren:

```
/begin TP_BLOB
  0x1000001  /* TP_BLOB version      */
  0x0        /* Project Base Address    */
  0x0        /* RstCfg parameter (MPC) */
```

- Damit die 100 MBit/s genutzt werden, aktualisieren Sie die Version des TP_BLOB und ersetzen Sie den Parameter `Project Base Address` durch den Parameter `INTERFACE_SPEED`.

Hinweis

Achten Sie darauf, dass Sie den TP_BLOB für **IF_DATA ETK** ändern. Nur dann ist die Umschaltung auf 100 MBit gültig.

Die geänderte Zeile muss wie folgt lauten (entspricht AML1.1.1/1.2):

```
/begin TP_BLOB
  0x1000100 /* TP_BLOP version */
  2 /* InterfaceSpeed 1=8MBit, 2=100MBit */
  0x0 /* RstCfg parameter (MPC) */
```

- Sichern Sie die Datei.
- Wählen Sie im Komponenten-Manager **Project** → **Read ASAM-2MC**, um die geänderte Datei in die Datenbank einzulesen.
- Aktivieren Sie im HWC-Editor das Register „Globals“ des ETK-BYPASS Devices (ES1232 / ETK-CTRL-BAS).
- Wählen Sie in der Zeile „ASAM-2MC Project“ das neu importierte ASAM-MCD-2MC-Projekt, wie im Abschnitt „ASAM-2MC Project“ auf Seite 150 beschrieben.

Im Register „Globals“ des ETK-CTRL-BAS Items wird in der Zeile „ETK Connection“ die Übertragungsrate von 100 MBit/s angezeigt.

10.8.5 Globals (ETK-CTRL-ADV Subsystem)

In der Registerkarte „Globals“ des ETK-CTRL-ADV Subsystems wird dem Subsystem ein physikalischer ETK-Controller bzw. ein ETK-Anschluss zugeordnet.

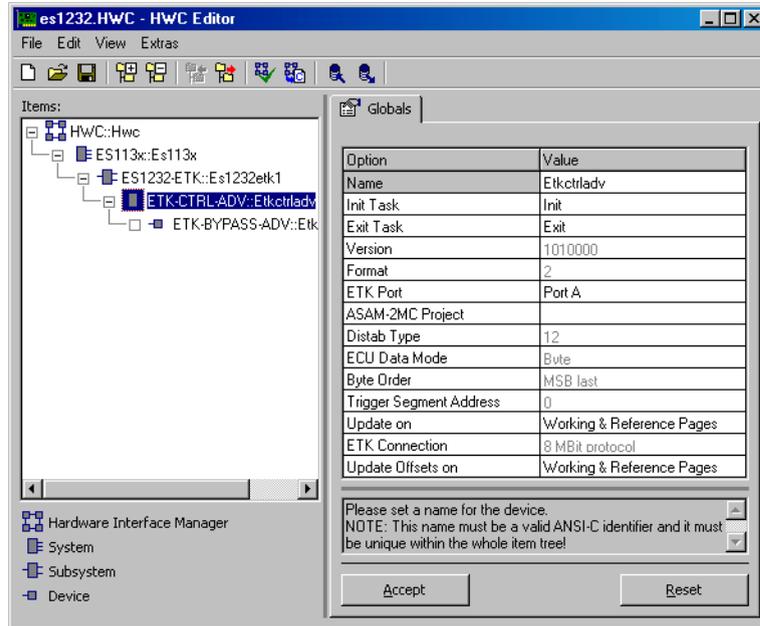


Abb. 10-33 Die Registerkarte „Globals“ des ETK-CTRL-ADV Subsystems

ETK Port

In dieser Zeile wird bei der ES1201 (s. Kapitel 10.3.2) einer der beiden unabhängigen ETK-Kanäle (Port A, Port B) und damit einer der beiden ETK-Controller der Karte als ETK-Schnittstelle zugeordnet. Bei der ES1232 kann hier nur der Port A gewählt werden.

ASAM-2MC Project

Der ETK-Bypass benötigt ein ASAM-MCD-2MC-Projekt (AML V1.2), das in der Datenbank durch Einlesen einer ASAM-MCD-2MC-Beschreibungsdatei generiert werden kann. Der Auswahldialog für diese Projekte ist in Abb. 10-8 auf Seite 151 abgebildet.

Hinweis

ASAM-MCD-2MC-Projekte, die AML V1.1.x verwenden, können nicht eingelesen werden. Folgende Fehlermeldung erscheint, wenn dem Device ein entsprechendes Projekt zugeordnet wird:

```
Error: Incompatible version 0x<version> of QP_BLOB
found in SOURCE '<sName>'. Expected version >= 0x100.
Selected ASAM-2MC Project is not suitable for the
advanced ETK Controller ETK-CTRL-ADV ! Please select
another one!
```

Die ASAM-MCD-2MC-Datei enthält unter IF_DATA ETK die Beschreibung des Bypass. Die in älteren Versionen auch mögliche Bezeichnung IF_DATA ASAP1B_ETK wird von der ES1232 mit ETK-CTRL-ADV Subsystem *nicht* unterstützt.

Für jeden Bypass-Kanal existiert ein QP_BLOB mit den Kanaleinstellungen; ein Beispiel finden Sie im Abschnitt „Andere AML-Versionen“ auf Seite 138). Im TP_BLOB sind allgemeine Einstellungen eingetragen.

```
/begin TP_BLOB
    0x1000100    /* TP_BLOP version          */
    2           /* InterfaceSpeed 1=8MBit,    */
               /*                2=100MBit */
    0x0         /* RstCfg parameter (MPC)    */
/begin DISTAB_CFG  0xC 0x1 MSB_LAST 0x383000 0x0
    TRG_MOD 0xB7
/end DISTAB_CFG
CODE_CHK 0x0 0x0 0x0 0x0
ETK_CFG 0xF 0xF0 0xFF 0x3 0xFD 0xEE 0xFF 0x1
RESERVED 0x810000 0x8103F9 EXTERN 0x-1 0x-1 0x-1
    0x-1 0x-1
```

```
/end TP_BLOB
```

Hinweis

Die Bedeutung der Parameter im Abschnitt DISTAB_CFG ist auf Seite 132 beschrieben, die Bedeutung der Parameter in den Abschnitten CODE_CHK und RESERVED auf .Seite 150.

Anders als bei ES1201, ES1231 und ES1232/ETK-CTRL-BAS *müssen* hier alle notwendigen Einstellungen in der ASAM-MCD-2MC-Datei enthalten sein. Sie haben keine Möglichkeit, die Daten manuell einzugeben.

Dies gilt auch für die lokalen BLD-Definitionen, die für die älteren Versionen wahlweise über die ASAM-MCD-2MC-Datei oder über einen Texteditor eingegeben werden konnten (Abschnitt „Local BLD Definitions“ auf Seite 152). Für die ES1232/ETK-CTRL-ADV ist die Bereitstellung der Informationen in einem Abschnitt MEASUREMENT mit KP_BLOB in der ASAM-MCD-2MC-Datei zwingend erforderlich.

```
/begin MEASUREMENT log_uint8_0_A
  " "
  UBYTE
  ident
  1
  100
  0.0
  1.0
  READ_ONLY
  BIT_MASK 0x8
  ECU_ADDRESS 0xFD00
  /begin IF_DATA ASAP1B_Bypass
  /begin KP_BLOB
    BUFFER_OFFSET
      "log_uint8_0_offset_A.Model_Byp_A"
    /* no SOURCE_ID */
    BIT_OFFSET
      "log_uint8_0_bitOffset_A.Model_Byp_A"
    POSSIBLE_SOURCES 5
  /end KP_BLOB
```

```
/end IF_DATA
/end MEASUREMENT
```

Hier wird dieselbe Variable definiert wie auf Seite 154.

Distab Type

In dieser Zeile wird dargestellt, welches DISTAB-Verfahren zum Datenaustausch mit dem Steuergerät verwendet wird. Diese Information wird aus dem zugeordneten ASAM-MCD-2MC-Projekt ausgelesen (siehe Seite 132).

- DISTAB 12 unterstützt Signale mit bis zu zwei Byte Länge
- DISTAB 13 (Standard) unterstützt 1, 2, 4 und 8 Byte lange Signale

ECU Data Mode

In dieser Zeile wird dargestellt, mit welcher Zugriffsart (Byte-Zugriff / Wort-Zugriff) der Steuergeräte-Prozessor auf den Datenspeicher zugreift. Das Feld ist nur relevant, wenn DISTAB 12 gewählt wurde.

Das Feld kann nicht bearbeitet werden; der Wert wird aus dem ASAM-MCD-2MC-Projekt übernommen (s. Seite 132).

Byte Order

In dieser Zeile wird das Wort-Datenablageformat (MSB first / MSB last) des Steuergeräteprozessors angezeigt. Diese Information wird aus dem zugeordneten ASAM-MCD-2MC-Projekt ausgelesen (siehe Seite 132).

Trigger Segment Address

Hier wird die Triggersegmentadresse des DISTAB-Verfahrens angegeben. Das Feld kann nicht bearbeitet werden; der Wert wird aus dem ASAM-MCD-2MC-Projekt übernommen (s. Seite 132), sofern die entsprechende Größe ordnungsgemäß definiert wurde.

Die Triggersegmentadresse bestimmt die Lage der Hardware-Triggeradressen die zur Steuerung der Datenübertragung über die Bypass-Kanäle genutzt werden. Die Lage der Hardware-Triggeradressen in Relation zur Triggersegmentadresse ist beim DISTAB-Datenaustauschverfahren konstant.

Die Anfangsadresse eines 64 Byte Triggersegments (bei 8 oder 16 Bit breitem Buszugriff) muss auf einer geraden, durch 64 teilbaren Adresse liegen, die Anfangsadresse eines 128 Byte Triggersegments (bei 32 Bit breiten Buszugriff) entsprechend auf einer durch 128 teilbaren Adresse.

Das Format des Triggersegments ist in der Schnittstellenbeschreibung von DISTAB 12 und DISTAB 13 definiert.

Update On

In dieser Zeile wird (für parallele ETKs) angegeben, ob während der Bypass-Kommunikation die Übergabe der Display-Tabelle (s. „Bypass-Kommunikation (AML V1.1)“ auf Seite 133) zwischen Simulationsrechner und Steuergerät auf der Arbeitsseite (*Working Page*), Referenzseite (*Reference Page*) oder auf Arbeits- und Referenzseite (*Working & Reference Pages*) des ETK durchgeführt werden soll.

Für serielle ETKs, bei denen alle Display-Tabellen im RAM-Bereich des Speichers liegen, wird automatisch *RAM Page* eingetragen. Die Zeile kann in diesem Fall nicht bearbeitet werden.

ETK Connection

In dieser Zeile wird die Übertragungsrate (8MBit/s oder 100MBit/s) angezeigt. Das Feld kann nicht bearbeitet werden; der Wert wird aus dem ASAM-MCD-2MC-Projekt übernommen (s. Seite 207).

Update Offsets On

In dieser Zeile wird für parallele und serielle ETKs angegeben, ob während der Bypass-Kommunikation die Übergabe der Bypass-Offsets (s. Seite 137) zwischen Simulationsrechner und Steuergerät auf der Arbeitsseite (*Working Page*), Referenzseite (*Reference Page*), auf Arbeits- und Referenzseite (*Working & Reference Pages*) oder im RAM (*RAM Page*, nur serielle ETKs) des ETK durchgeführt werden soll.

Für serielle ETKs wird standardmäßig *RAM Page* eingetragen.

10.8.6 Globals (ETK-BYPASS-ADV Device)

Das ETK-BYPASS-ADV Device dient zur Definition aller notwendigen Größen um einen Bypass-Betrieb mit der ES1232 zu realisieren.

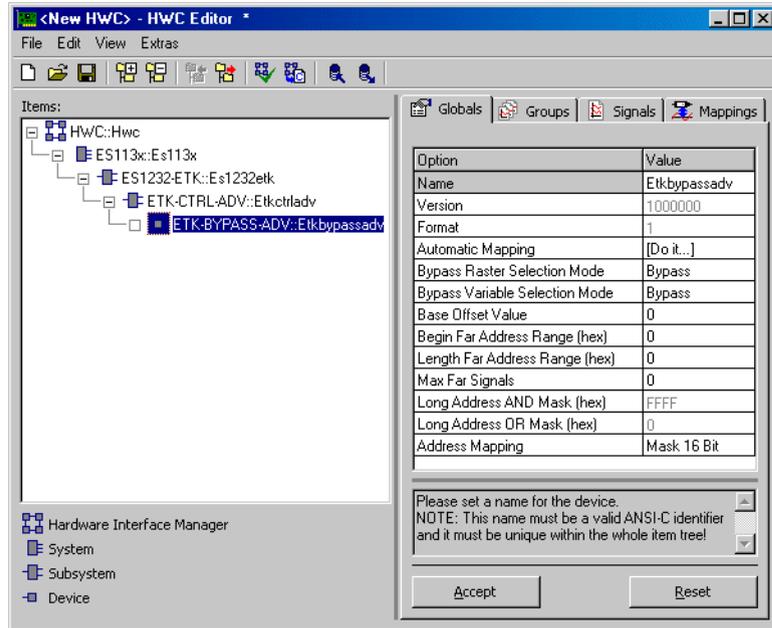


Abb. 10-34 Die Registerkarte „Globals“ des ETK-BYPASS-ADV Device

Automatic Mapping

Siehe hierzu die Beschreibung „Automatic Mapping“ auf Seite 172 in Kapitel 10.4.3.

Bypass Raster Selection Mode

In dieser Zeile legen Sie fest, ob im Register „Groups“ Bypass- und Messraster angezeigt werden (*Bypass&Measurement*) oder nur Bypass-Raster (*Bypass*; Standard).

Die Signale der Messraster können bei Auswahl von *Bypass&Measurement* selbst dann in einem ASCET-Experiment gemessen werden, wenn für das Steuergerät nur INCA-Hooks definiert sind.

Wenn Sie die Einstellung wechseln, bleiben die Signale, die Sie bereits für die Bypass-Raster ausgewählt haben, erhalten. Die für Messraster ausgewählten Signale jedoch gehen verloren, wenn Sie die Auswahl von `Bypass&Measurement` nach `Bypass` ändern.

Bypass- und Messraster werden gleich behandelt, mit einer Ausnahme: Der RTIO-Treiber initialisiert Bypass-Raster mit Master-Zugriffsrechten, während Messraster mit Slave-Zugriffsrechten initialisiert werden müssen. Da INCA immer Master-Zugriffsrechte für die Messraster verwendet, kann es in zwei Fällen zu Fehlern kommen:

1. Zu dem Zeitpunkt, an dem der ASCET-Bypass gestartet wird, läuft bereits eine Messung mit INCA. Der RTIO-Treiber kann die Messraster aufgrund fehlender Zugriffsrechte nicht initialisieren.
2. Eine Messung mit INCA wird gestartet, während der ASCET-Bypass läuft. INCA „stiehlt“ aufgrund höherer Zugriffsrechte ASCET die Messraster.

In beiden Fällen werden entsprechende Fehlermeldungen ausgegeben.

Bypass Variable Selection Mode

Siehe hierzu die Beschreibung „Bypass Variable Selection Mode“ auf Seite 156 in Kapitel 10.3.3.

Base Offset Value

In dieser Zeile kann der Byte-Offset jeder Bypass-Ausgangsgröße um 0, 2, 4 oder 8 Byte nach oben verschoben werden.

Der Standardwert für den Byte-Offset bei DISTAB 12 ist 0. Bei DISTAB 13 ist der Standardwert für den Byte-Offset 8.

Begin Far Address Range (hex)

Siehe hierzu die Beschreibung „Begin Far Address Range (hex)“ in Kapitel 10.3.3.

Length Far Address Range (hex)

In dieser Zeile wird die Größe des FAR-Adressbereichs angegeben. Diese Angabe ist in der Regel nur für DISTAB 12 Verfahren notwendig.

Max Far Signals

Siehe hierzu die Beschreibung „Max Far Signals“ in Kapitel 10.3.3.

Long Address AND / OR Mask (hex)

Siehe hierzu die Beschreibung „Long Address AND / OR Mask (hex)“ in Kapitel 10.3.3.

Address Mapping

Siehe hierzu die Beschreibung „Address Mapping“ in Kapitel 10.3.3.

10.8.7 Groups (ETK-BYPASS-ADV Device)

Die Anzahl der Signalgruppen im ETK-BYPASS-ADV Device ist in der ASAM-MCD-2MC-Datei vorgegeben. Abhängig von der Einstellung der Option „Bypass Raster Selection Mode“ im Register „Globals“ (s. Kapitel 10.8.6) werden die Signalgruppen für Bypass-Raster und Messraster automatisch angelegt oder gelöscht.

Hinweis

Wenn die Option „Bypass Raster Selection Mode“ von Bypass&Measurement auf Bypass geändert wird, bleiben die Signale, die Sie bereits für die Bypass-Trigger ausgewählt haben, erhalten. Die für Messtrigger ausgewählten Signale gehen jedoch verloren.

Jedes Bypass-Raster besteht aus zwei Signalgruppen, einer Sende- und einer Empfangs-Signalgruppe. Jedes Messraster besteht aus einer Empfangs-Signalgruppe.

Die Daten der Empfangs-Signalgruppe werden über das Standard-DISTAB-Verfahren übertragen, wohingegen die Daten der Sende-Signalgruppe über das Bypass-Tabellenverfahren zwischen Steuergerät und RTIO ausgetauscht werden.

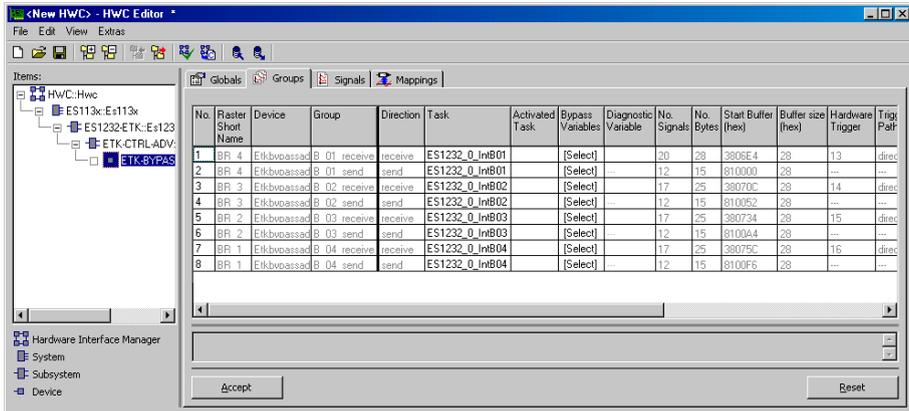


Abb. 10-35 Die Registerkarte „Groups“ des ETK-BYPASS-ADV Devices

Raster Short Name

Diese Spalte zeigt den Typ (MT für Messraster oder BT für Bypass-Raster) und die Nummer (Priorität) des Rasters, z.B. MT_12 oder BT_31. Die Signalgruppen werden standardmäßig nach absteigender Nummer (und damit gleichzeitiger Priorität) sortiert.

Die Spalte kann nicht bearbeitet werden; die Werte werden aus dem ASAM-MCD-2MC-Projekt übernommen.

Task

Siehe hierzu die Beschreibung „Task“ auf Seite 159 in Kapitel 10.3.4.

Activated Task

Siehe hierzu die Beschreibung „Activated Task“ auf Seite 159 in Kapitel 10.3.4.

Bypass Variables

Siehe hierzu die Beschreibung „Bypass Variables“ auf Seite 187 in Kapitel 10.5.4.

Diagnostic Variable

Siehe hierzu die Beschreibung „Diagnostic Variable“ auf Seite 188 in Kapitel 10.5.4.

No. Signals

Hier wird die Anzahl der ausgewählten Signale dieser Signalgruppe angezeigt. Der Wert wird automatisch aktualisiert, wenn Sie die Signalauswahl ändern.

No. Bytes

Hier wird die Anzahl der von den ausgewählten Signalen dieser Signalgruppe benötigten Bytes angezeigt. Der Wert wird automatisch aktualisiert, wenn Sie die Signalauswahl ändern.

Start Buffer (hex)

In dieser Spalte werden die Basisadressen der jeweiligen Datenpuffer für die Bypass-Eingangsdaten und -Ausgangsdaten angezeigt.

Diese Adressangaben werden aus dem ASAM-MCD-2MC-Projekt übernommen; die Eingabe ist gesperrt.

Buffer Size (hex)

In dieser Spalte werden die Größen der jeweiligen Datenpuffer für die Bypass-Eingangsdaten und -Ausgangsdaten angezeigt.

Diese Angaben werden aus dem ASAM-MCD-2MC-Projekt übernommen; die Eingabe ist gesperrt.

Hardware Trigger

In dieser Spalte werden die Nummern der zu den „receive“-Signalgruppen gehörigen Hardware-Trigger angezeigt.

Diese Angaben werden aus dem ASAM-MCD-2MC-Projekt übernommen; die Eingabe ist gesperrt.

Trigger Path

Diese Spalte zeigt für „receive“-Signalgruppen an, auf welche Weise die Signalübertragung mittels Trigger geschieht:

- **direct** – Das Raster ist direkt einem Hardware-Trigger zugeordnet. Nach dem Beschreiben der Triggeradresse durch das Steuergerät kann die Datenerfassung direkt (ohne Lesen eines Trigger-Flags) erfolgen.

- `indirect` – Mehrere Raster sind dem selben Hardware-Trigger zugeordnet. Damit unterschieden werden kann, welches Raster den Trigger ausgelöst hat, muss eine rasterspezifische „Trigger Flag Address“ gelesen werden.

Direkte Übertragung ist immer dann möglich, wenn genügend Hardware-Triggere zur Verfügung stehen. Bei seriellen ETKs ist das oft nicht der Fall; für diese muss indirekte Übertragung verwendet werden.

Die Art der Signalübertragung wird aus dem ASAM-MCD-2MC-Projekt übernommen; die Eingabe ist gesperrt.

Für „send“-Signalgruppen ist das Feld ohne Bedeutung; „---“ wird angezeigt.

Trigger Flag Address

Diese Option ist nur von Bedeutung, wenn der „Trigger Path“ `indirect` ausgewählt wurde. In diesem Fall wird der rasterspezifische Flag angezeigt, der für die korrekte Datenerfassung benötigt wird. Der Wert wird aus dem ASAM-MCD-2MC-Projekt übernommen.

Im 8 MBit/s-Modus entspricht der Trigger-Flag dem Trigger-Identifizierer anderer ETK-Bypass-Devices (siehe Abschnitt „Trigger Id (hex)“ auf Seite 160).

Wenn der „Trigger Path“ `direct` ausgewählt wurde, wird „---“ angezeigt.

Die Eingabe ist gesperrt.

Start Pointer (hex)

In dieser Spalte werden für die „receive“ Signalgruppen die Startadressen der Pointerlisten für das verwendete DISTAB-Verfahren angezeigt.

Diese Adressangaben werden aus dem ASAM-MCD-2MC-Projekt übernommen; die Eingabe ist gesperrt.

10.8.8 Signals (ETK-BYPASS-ADV Device)

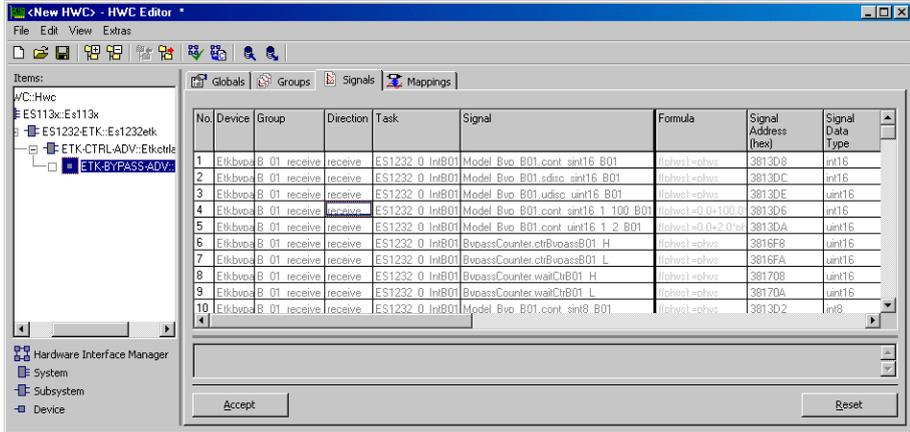


Abb. 10-36 Die Registerkarte „Signals“ des ETK-BYPASS-ADV Devices
Siehe hierzu Kapitel 10.3.5 auf Seite 161 und Kapitel 10.7.5 auf Seite 198.

10.8.9 Mappings (ETK-BYPASS-ADV Device)

Die Einstellmöglichkeiten sind für alle Items gleich. Sie werden in Kapitel 7.3.5 auf Seite 120 beschrieben.

10.9 ES1300-AD

Die analoge Eingangskarte ES1300 ermöglicht die Erfassung von analogen Signalen sowohl differentiell (max. 8 Kanäle) als auch single-ended (max. 16 Kanäle).

10.9.1 Globals (ES1300-AD Device)

In diesem Abschnitt werden die globalen Eigenschaften des ES1300-AD Devices beschrieben.

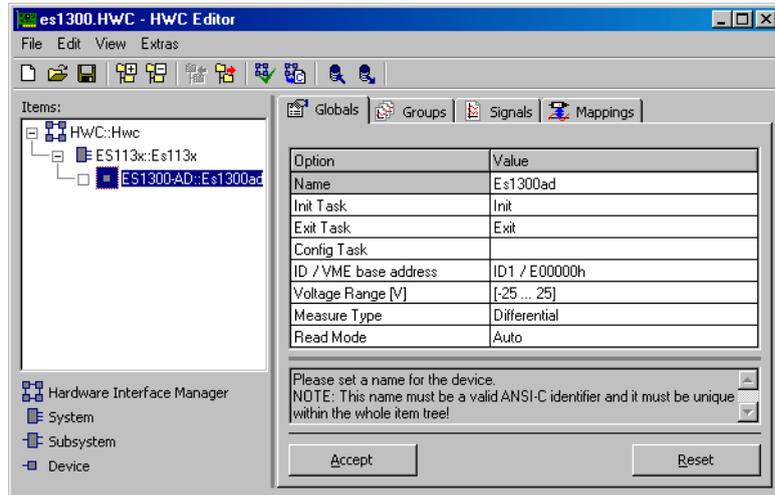


Abb. 10-37 Die Registerkarte „Globals“ des ES1300-AD Devices

Init Task

In dieser Zeile wird die Task zur Initialisierung der ES1300 zugeordnet (Type: Init / Application Mode: active).

Exit Task

In dieser Zeile wird die Task zugeordnet, die bei einem Stopp des Experiments mit der ES1300 ausgeführt werden soll (Type: Init / Application Mode: inactive).

ID / VME base address

In dieser Zeile wird die Einstellung der VME-Basisadresse vorgenommen. Diese Einstellung muss mit den Jumper-Einstellungen der ES1300 übereinstimmen. Für die ES1300 können vier unterschiedliche VME-Basisadressen gewählt werden (ID1 / E00000h, ID2 / E00100h, ID3 / E00200h, ID4 / E00300h),

dadurch können bis zu vier ES1300 in einem ETAS Experimentalsystem betrieben werden. Die ES1300 belegt einen Adressbereich von 256 Worten ab der Basisadresse.

Voltage Range

In dieser Zeile wird der Eingangsspannungsbereich für jeden Kanal der ES1300 eingestellt. Diese Einstellung muss mit den Jumper-Einstellungen der ES1300 übereinstimmen. Folgende Eingangsspannungsbereiche können gewählt werden: -25V bis 25V, -50V bis 50V und 0V bis 50V.

Measure Type

In dieser Zeile wird die Messbetriebsart differentiell (max. 8 Kanäle) oder single-ended (max. 16 Kanäle) festgelegt. Diese Einstellung muss mit den Jumper-Einstellungen der ES1300 übereinstimmen.

Read Mode

In dieser Zeile wird der Lesemodus eingestellt. Bei der ES1300 kann gewählt werden, ob die gemessenen A/D-Werte vom Hardwaretreiber vor oder nach einer erneuten A/D-Abtastung an ASCET übergeben werden. Es gibt folgende Einstellmöglichkeiten:

- `auto`:
Der Treiber entscheidet anhand der verstrichenen Zeit zwischen zwei ASCET Zugriffen ob er die Werte vom letzten Rechenraster oder die Werte vom aktuellen Rechenraster übergibt.
- `wait for new values`:
Der Treiber wartet vor der Übergabe auf die Werte vom aktuellen Rechenraster.
- `get old values`:
Der Treiber übergibt die Werte vom letzten Rechenraster sofort.

10.9.2 Groups (ES1300-AD Device)

In diesem Abschnitt werden die signalgruppenspezifischen Einstellungen des ES1300-AD Devices beschrieben.

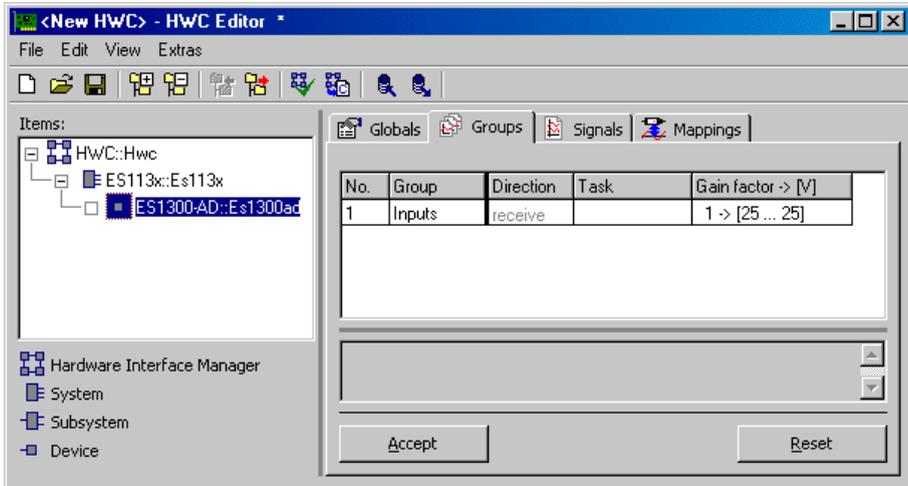


Abb. 10-38 Die Registerkarte „Groups“ des ES1300-AD Devices

Gain factor -> [V]

In dieser Spalte kann der programmierbare Verstärkungsfaktor der ES1300 für die analogen Eingangssignale eingestellt werden. Folgende Verstärkungsfaktoren können gewählt werden: 1, 5, 10, 50, 100 und 500. Die Verstärkungsfaktoren bewirken eine höhere Auflösung des Messbereichs; dies wiederum führt dazu, dass nur ein eingeschränkter Bereich der Eingangsspannung als effektiver Messbereich zur Verfügung steht.

Der resultierende effektive Messbereich wird in der Spalte „Gain Factor“ mit angegeben. Er hängt vom Eingangsspannungsbereich (Register „Globals“, s. Seite 220) und dem Verstärkungsfaktor ab.

10.9.3 Signals (ES1300-AD Device)

In diesem Abschnitt werden die signalspezifischen Einstellungen des ES1300-AD Devices beschrieben.

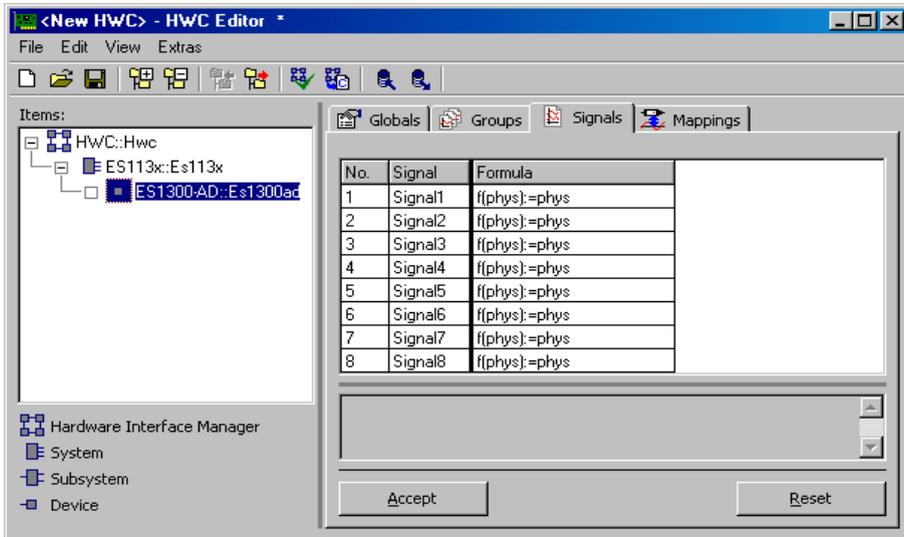


Abb. 10-39 Die Registerkarte „Signals“ des ES1300-AD Devices

No.

Die Anzahl der maximal nutzbaren analogen Eingangssignale (8 / 16) wird für die ES1300 mit der *Measure Type* Einstellung (*differential* / *single-ended*) aus der Registerkarte „Globals“ festgelegt.

10.9.4 Mappings (ES1300-AD Device)

Die Einstellmöglichkeiten sind für alle Items gleich. Sie werden in Kapitel 7.3.5 auf Seite 120 beschrieben.

10.10 ES1301-AD

Die analoge Eingangskarte ES1301 ermöglicht die Erfassung von bis zu sieben analogen Signalen.

10.10.1 Globals (ES1301-AD Device)

In diesem Abschnitt werden die globalen Eigenschaften des ES1301-AD Devices beschrieben.

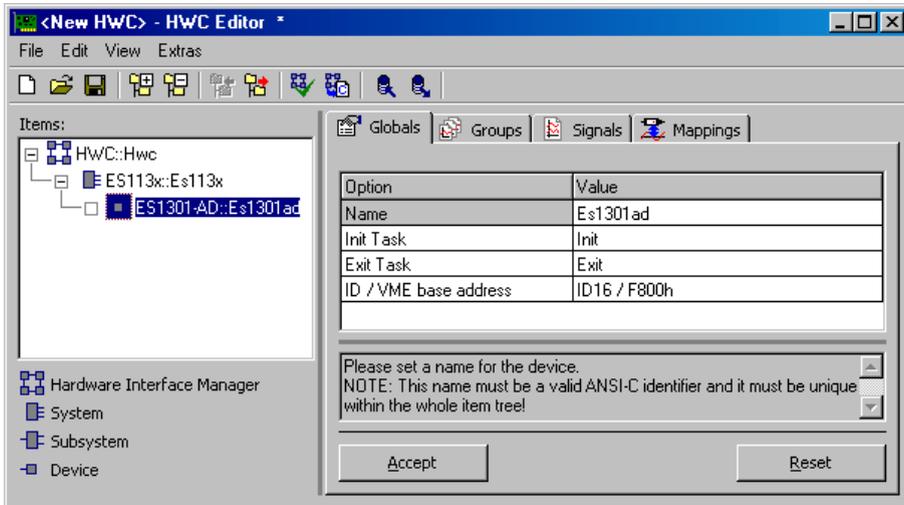


Abb. 10-40 Die Registerkarte „Globals“ des ES1301-AD Devices

Init Task

In dieser Zeile wird die Task zur Initialisierung der ES1301 zugeordnet (Type: Init / Application Mode: active).

Exit Task

In dieser Zeile wird die Task zugeordnet, die bei einem Stop des Experiments mit der ES1301 ausgeführt werden soll (Type: Init / Application Mode: inactive).

ID / VME base address

In dieser Zeile wird die Einstellung der VME-Basisadresse vorgenommen. Diese Einstellung muss mit den durch Lötbrücken codierten Einstellungen der ES1301 übereinstimmen. Für die ES1301 können fünf unterschiedliche VME-Basisadressen gewählt werden (ID16/F800h, ID15/F000h, ID14/E800h, ID12/D800h, ID8/B800h), dadurch können bis zu fünf ES1301 in einem ETAS Experimentalsystem betrieben werden. Die ES1301 belegt einen Adressbereich von 2048 Worten ab der Basisadresse.

10.10.2 Groups (ES1301-AD Device)

In diesem Abschnitt werden signalgruppenspezifischen Eigenschaften des ES1301 Devices beschrieben.

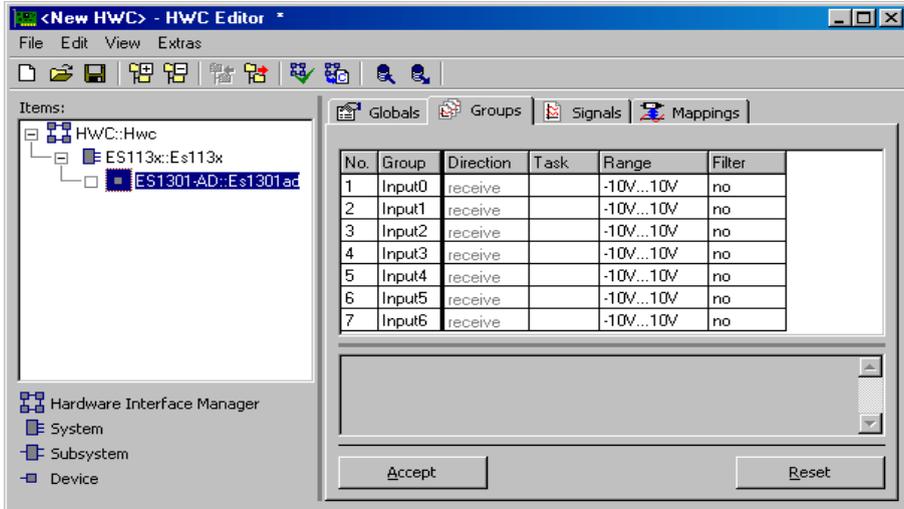


Abb. 10-41 Die Registerkarte „Groups“ des ES1301-AD Devices

Range

In dieser Spalte wird der Eingangsspannungsbereich für jeden Kanal der ES1301 eingestellt. Folgende Eingangsspannungsbereiche können gewählt werden: -10V bis 10V, -5V bis 5V, 0V bis 10V und 0V bis 5V. Die Konfiguration zur Wahl des Eingangsspannungsbereichs geschieht per Software, es sind keine Jumper-Einstellungen notwendig.

Filter

In dieser Spalte kann ein Tiefpassfilter 2. Ordnung für jeden Kanal eingestellt werden. Folgende Tiefpassfilter können gewählt werden: 100 Hz, 200 Hz, 500 Hz, 1 KHz, 2 KHz, 5 KHz und 10 KHz.

10.10.3 Signals (ES1301-AD Device)

In diesem Abschnitt werden die signalspezifischen Eigenschaften des ES1301-AD Devices beschrieben.

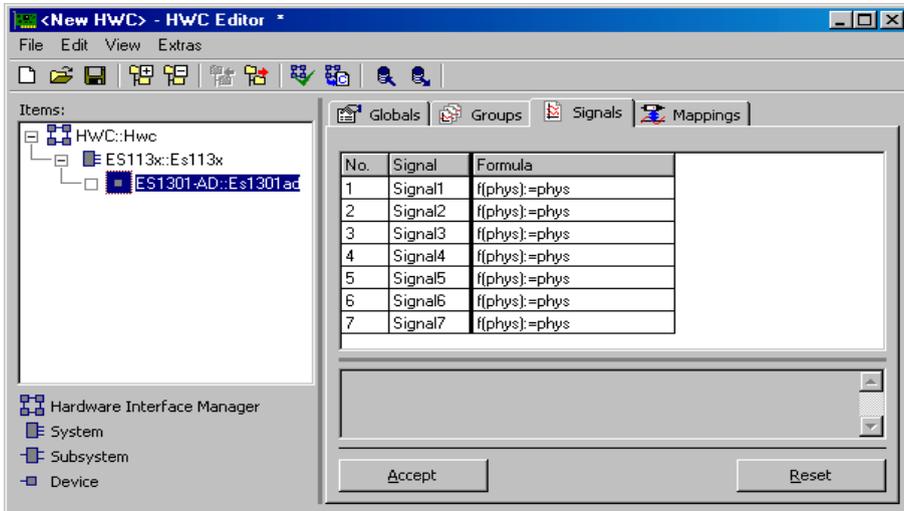


Abb. 10-42 Die Registerkarte „Signals“ des ES1301-AD Devices

Für das ES1301-AD Device sind in der Registerkarte „Signals“ keine Item-spezifischen Spalten definiert.

10.10.4 Mappings (ES1301-AD Device)

Die Einstellmöglichkeiten sind für alle Items gleich. Sie werden in Kapitel 7.3.5 auf Seite 120 beschrieben.

10.11 ES1303-AD

Die analoge Eingangskarte ES1303 ermöglicht die Erfassung von analogen Signalen (max. 16 Kanäle). Sie ist für hohe Abtastraten (max. 100 kHz, erlaubt die Erfassung von Signalen mit max. 50 kHz) und hohe Auflösung konzipiert. Bis zu 4 ES1303 Karten können gleichzeitig betrieben werden.

Durch die beiden zusätzlich vorhandenen digitalen Triggereingänge kann die Signalerfassung zeitlich gesteuert werden (Start/Stop). Dabei sind sowohl eine kontinuierliche Messung als auch ist die Erfassung einzelner Messwerte möglich (Single-Shot, siehe ES1303 Benutzerhandbuch).

Jeder der 16 Analogkanäle kann wahlweise von ASCET-RP oder INCA genutzt werden, ein gemischter Betrieb ist möglich. Wenn jedoch eine ES1303 Konfiguration mit INCA läuft, kann ASCET-RP diese Konfiguration nicht überschreiben; eine Fehlermeldung wird ausgegeben. Die beiden Digitalkanäle können nur von einem der beiden Programme benutzt werden.

10.11.1 Globals (ES1303-AD Device)

In diesem Abschnitt werden die globalen Eigenschaften des ES1303-AD Devices beschrieben.

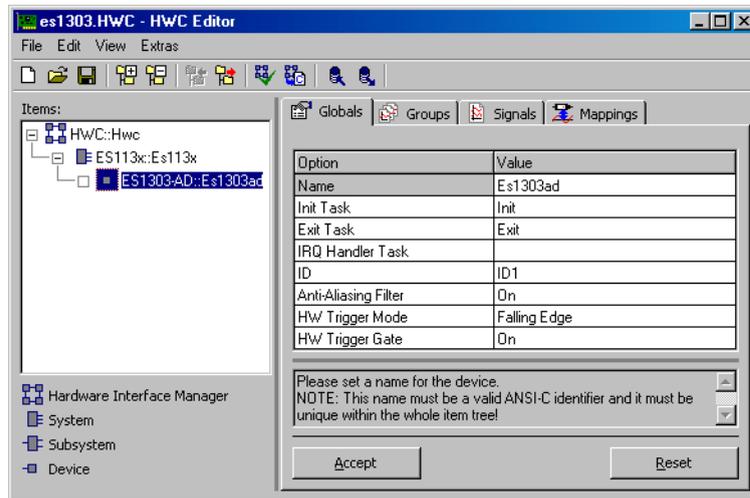


Abb. 10-43 Die Registerkarte „Globals“ des ES1303-AD Devices

Init Task

In dieser Zeile wird die Task zur Initialisierung der ES1303 zugeordnet (Type: Init / Application Mode: active).

Exit Task

In dieser Zeile wird die Task zugeordnet, die bei einem Stopp des Experiments mit der ES1303 ausgeführt werden soll (Type: Init / Application Mode: inactive).

IRQ Handler Task

Die IRQ Handler-Task (Type: Software / Application Mode: active) wird benötigt, wenn Signalkanäle im Interrupt-Betrieb ausgewertet werden sollen.

Diese Task muss im ASCET Projekteditor in der Task-Liste als „Software“-Task angelegt werden, wobei bei „Max. No. of Activations“ mindestens die Anzahl 2 (maximal 50) einzutragen ist.

Hinweis

*Diese Task muss **ausschließlich** für das Element bzw. den betreffenden Hardware-Treiber zur Verfügung stehen und darf von keinem anderen Element bzw. von Anwender-Prozessen mitbenutzt werden. Auch zwei gleichartige Elemente (z.B. zwei ES1303) dürfen sich **nicht** die gleiche Task teilen!*

Diese Option ist gesperrt, wenn der Triggermodus OFF ausgewählt ist (siehe „HW Trigger Mode“).

ID

Im Auswahlfeld „ID“ muss die Kartenummer von der Karte eingegeben werden, welche angesprochen werden soll.

Hinweis

Bei der ES1303 Karte handelt es sich um eine sogenannte „Auto-ID“ Karte, die beim Einschalten des Systems vom Hardware-Manager automatisch eine ID-Nummer und einen freien Adressbereich zugewiesen bekommt. Karten vom selben Typ (z.B. zwei ES1303) werden von links nach rechts nummeriert, d.h. die linke Karte erhält die Nummer Eins, anschließend die Nummer Zwei usw. Es stehen 4 IDs zur Verfügung; somit können bis zu 4 Karten in einem Experimentalsystem betrieben werden.

Anti-Aliasing Filter

In dieser Zeile wird der Anti-Aliasing-Filter aktiviert (ON) oder deaktiviert (OFF). Der Filter wird benötigt, um die Bandbreite des Eingangssignals zu begrenzen, damit das Abtasttheorem (Abtastfrequenz mindestens doppelt so groß wie die Bandbreite des Eingangssignals) nicht verletzt wird.

Der Filter kann nur global für alle Kanäle ein- oder ausgeschaltet werden. Die Hardware erlaubt jedoch verschiedene Einstellungen für vier Vierergruppen (Kanal 1–4, Kanal 5–8, Kanal 9–12, Kanal 13–16). Werden nun sowohl von ASCET als auch von INCA einige Kanäle angesprochen, kann es zu Konflikten kommen, wenn beide Programme unterschiedliche Filtereinstellungen verwenden. Wenn beispielsweise ASCET ohne Filter mit Kanal 1 und 2 arbeiten möchte und INCA mit Filter auf die Kanäle 4 und 5 zugreift, kommt es zu einem Konflikt, da die Kanäle der ersten Gruppe (1 – 4) nur mit derselben Filtereinstellung betrieben werden müssen. Es wird eine Fehlermeldung ausgegeben. Würde INCA die Kanäle 5 und 6 verwenden, wäre der Konflikt gelöst.

Der Anti-Aliasing-Filter ist für die Abtastrate von 100 kHz optimiert. Wird der Filter nicht verwendet, sind ein Tiefpassfilter und eine Abschneidefrequenz von 225 kHz aktiv.

HW Trigger Mode

In dieser Zeile wird der Triggermodus gesetzt, der den Start von getriggerten Messungen bestimmt. Folgende Modi stehen zur Verfügung:

- `OFF` – die Triggersignale werden nicht verwendet.
- `RISING_EDGE` – die Messung wird mit der steigenden Flanke des ersten Triggersignals gestartet (Modus III im ES1303 Benutzerhandbuch).
- `FALLING_EDGE` – die Messung wird mit der fallenden Flanke des ersten Triggersignals gestartet (Modus IV im ES1303 Benutzerhandbuch).

Wenn im Register „Groups“ für eine Signalgruppe die Spalte „IRQ“ auf `No` gesetzt wird (s. „IRQ“ auf Seite 230), ist die Auswahl des Triggermodus bedeutungslos.

HW Trigger Gate

In dieser Zeile wird festgelegt, ob das zweite Triggersignal für die Triggererzeugung benutzt wird (`ON`) oder nicht (`OFF`).

Wenn für das Trigger Gate `ON` ausgewählt ist, startet die unter „HW Trigger Mode“ ausgewählte Flanke des ersten Triggersignals nur dann eine Messung, wenn gleichzeitig am zweiten Triggersignal ein hoher Puls anliegt. Wenn für das Trigger Gate `OFF` ausgewählt ist, startet jede unter „HW Trigger Mode“ ausgewählte Flanke des ersten Triggersignals eine Messung. Die nachstehende Tabelle fasst die Bedingungen für den Start der Messung zusammen.

HW Trigger Mode	HW Trigger Gate	Triggersignal 1	Triggersignal 2
<code>RISING_EDGE</code>	<code>ON</code>	steigende Flanke	hoch
<code>RISING_EDGE</code>	<code>OFF</code>	steigende Flanke	beliebig
<code>FALLING_EDGE</code>	<code>ON</code>	fallende Flanke	hoch
<code>FALLING_EDGE</code>	<code>OFF</code>	fallende Flanke	beliebig

Wenn im Register „Groups“ für eine Signalgruppe die Spalte „IRQ“ auf `No` gesetzt wird (s. „IRQ“ auf Seite 230), ist die Auswahl des Trigger Gate bedeutungslos.

10.11.2 Groups (ES1303-AD Device)

In diesem Abschnitt werden die signalgruppenspezifischen Einstellungen des ES1303-AD Devices beschrieben.

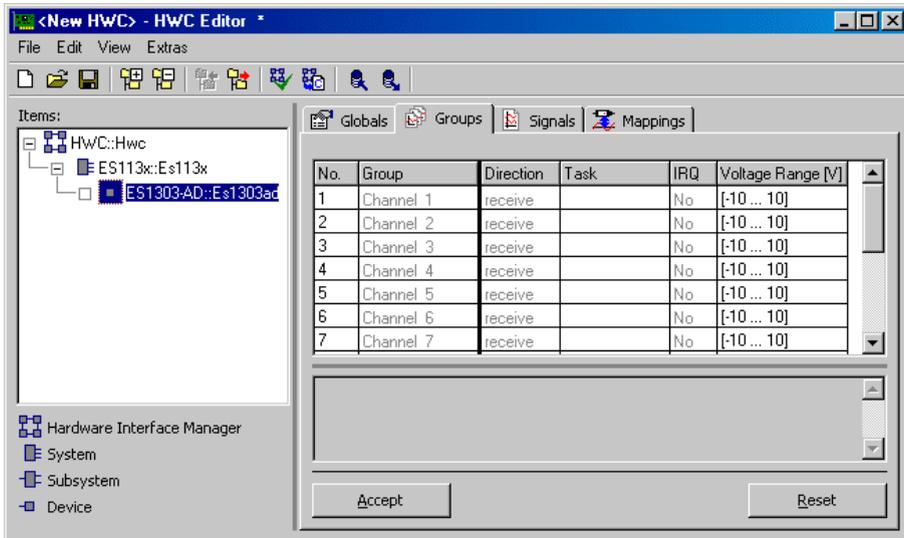


Abb. 10-44 Die Registerkarte „Groups“ des ES1303-AD Devices

Die 16 Signalgruppen sind vordefiniert. Zu jeder Signalgruppe gehört genau ein Signal mit fest vorgegebenem Namen.

Hinweis

Eine zukünftige Variante der ES1303 wird nur 8 Kanäle haben. Wenn im HWC-Editor mehr als 8 Kanäle konfiguriert wurden, die verwendete Karte jedoch nur 8 Kanäle hat, wird beim Start des Modells eine Fehlermeldung ausgegeben; die ES1303 verhält sich in dem Fall als wäre sie nicht vorhanden.

Task

Hier wird die Task angegeben, in der das Signal empfangen werden soll. Sie haben die Möglichkeit, mehrere Tasks auszuwählen, um das Signal in mehr als einem Raster zu erfassen.

Soll eine Empfangsbotschaft im Interrupt-Betrieb empfangen werden, so wird diese Einstellung zurückgesetzt und gesperrt.

IRQ

Diese Spalte gibt an, ob das betreffende Empfangssignal im Interrupt-Betrieb empfangen werden soll (Yes) oder nicht (No). Die Option ist gesperrt, wenn der Triggermodus Off ausgewählt ist (siehe „HW Trigger Mode“).

Für den „normalen“ Empfang („IRQ“ auf No) muss aufgrund des Shannon-Theorems die Abtast- oder Pollingfrequenz in einer Task mindestens doppelt so groß sein wie die Signalfrequenz (siehe Kapitel 3.1.2 im ES1303-Benutzerhandbuch). Problematisch sind aber bei dieser Betriebsart Signale, die in keinem festen Raster versendet werden oder sogar nur sporadisch auftauchen. Hierfür ist der Interrupt-Empfang ideal geeignet, da er eine Botschaftsverarbeitung genau dann anstößt, wenn die Botschaft empfangen wurde.

Jede Signalgruppe kann individuell eingestellt werden. Für alle Signalgruppen, die im Interrupt-Betrieb empfangen werden, gelten die Einstellungen für „HW Trigger Mode“ und „HW Trigger Gate“.

Eine leere Gruppe, für die „IRQ“ auf No gesetzt und keine Task gewählt wurde, generiert keinen Prozess. Im ASCET Monitorfenster wird ein Hinweis ausgegeben.

Voltage Range [V]

In dieser Spalte wird der Eingangsspannungsbereich für jeden Kanal der ES1303 eingestellt. Folgende Eingangsspannungsbereiche können gewählt werden: -10V bis 10V oder -60V bis 60V.

10.11.3 Signals (ES1303-AD Device)

In diesem Abschnitt werden die signalspezifischen Einstellungen des ES1303-AD Devices beschrieben.

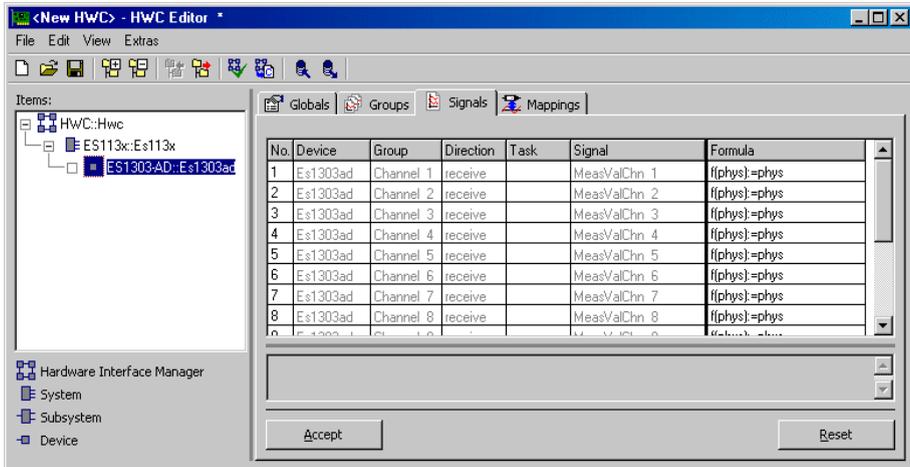


Abb. 10-45 Die Registerkarte „Signals“ des ES1303-AD Devices

Anders als bei der ES1300 (s. Kapitel 10.9.3) können Sie im Register „Signals“ nur die Formel einstellen; der Signalname ist fest vorgegeben. Eine Beschreibung der verschiedenen Spalten finden Sie in Kapitel 7.3.4.

10.11.4 Mappings (ES1303-AD Device)

Die Einstellmöglichkeiten sind für alle Items gleich. Sie werden in Kapitel 7.3.5 auf Seite 120 beschrieben.

10.12 ES1310-DA

Die Analoge Ausgangskarte ES1310 ermöglicht die Ausgabe von acht analogen Signalen.

10.12.1 Globals (ES1310-DA Device)

In diesem Abschnitt werden die globalen Einstellungen des ES1310-DA Devices beschrieben.

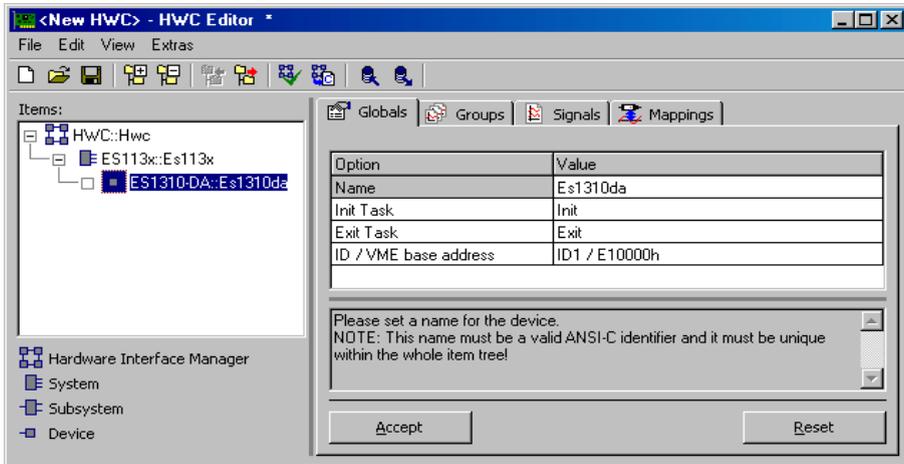


Abb. 10-46 Die Registerkarte „Globals“ des ES1310-DA Devices

Init Task

In dieser Zeile wird die Task zur Initialisierung der ES1310 zugeordnet (Type: Init / Application Mode: active).

Exit Task

In dieser Zeile wird die Task zugeordnet, die bei einem Stopp des Experiments mit der ES1310 ausgeführt werden soll (Type: Init / Application Mode: inactive).

ID / VME base address

In dieser Zeile wird die Einstellung der VME-Basisadresse vorgenommen. Diese Einstellung muss mit den Jumper-Einstellungen der ES1310 übereinstimmen. Für die ES1310 können zwei unterschiedliche VME-Basisadressen gewählt werden (ID1 / E10000h, ID2 / E20000h), dadurch können bis zu zwei ES1310 in einem ETAS Experimentalsystem betrieben werden. Die ES1310 belegt einen Adressbereich von 1024 Worten ab der Basisadresse.

10.12.2 Groups (ES1310-DA Device)

In diesem Abschnitt werden die signalgruppenspezifischen Einstellungen des ES1310-DA Devices beschrieben.

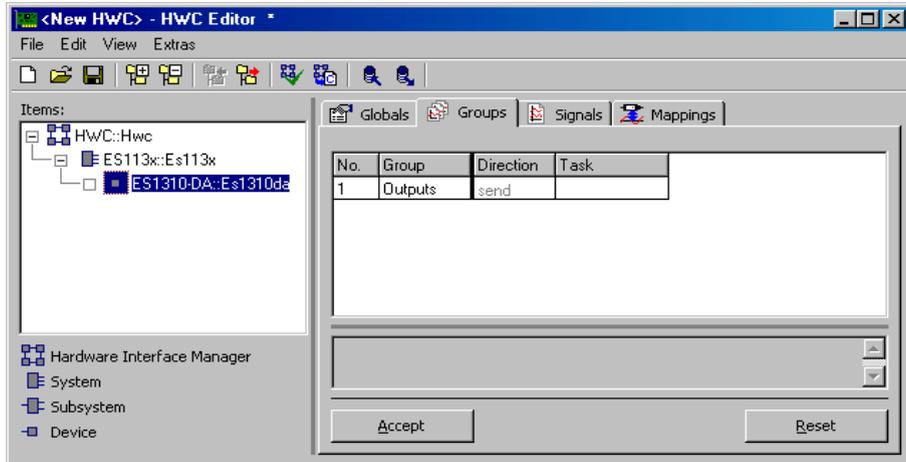


Abb. 10-47 Die Registerkarte „Groups“ des ES1310-DA Devices

Für das ES1310-DA Device sind in der Registerkarte „Groups“ keine Item-spezifischen Spalten definiert.

10.12.3 Signals (ES1310-DA Device)

In diesem Abschnitt werden die signalspezifischen Einstellungen des ES1310-DA Devices beschrieben.

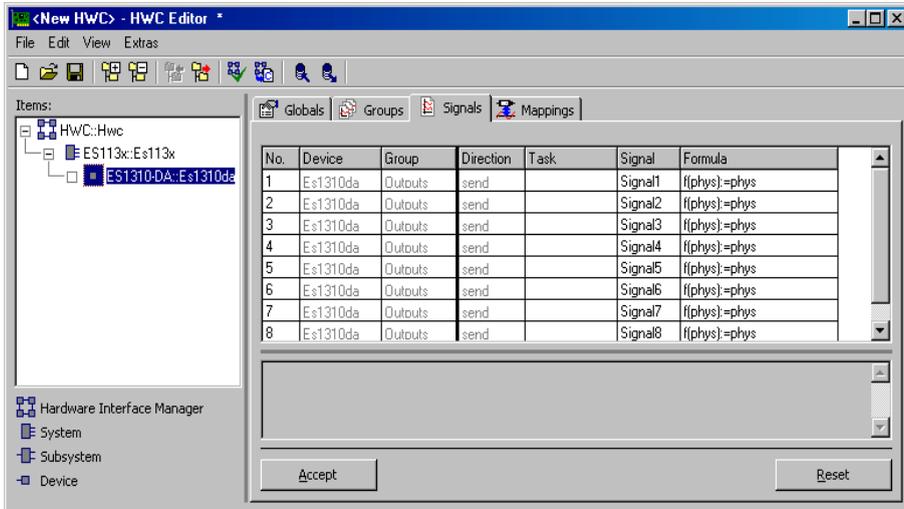


Abb. 10-48 Die Registerkarte „Signals“ des ES1310-DA Devices

Für das ES1310-DA Device sind in der Registerkarte „Signals“ keine Item-spezifischen Spalten definiert.

10.12.4 Mappings (ES1310-DA Device)

Die Einstellmöglichkeiten sind für alle Items gleich. Sie werden in Kapitel 7.3.5 auf Seite 120 beschrieben.

10.13 ES1320-CB (DIO)

Die digitale Ein-/Ausgangskarte ES1320 (DIO) ermöglicht die Erfassung und Ausgabe von jeweils zwanzig digitalen Signalen. Die ES1320 Hardware besteht aus einer Trägerplatine (CB) mit zwei DIO Aufsteckmodulen.

In der Items-Liste des HWC-Editors wird die Trägerplatine als Subsystem und das DIO-Aufsteckmodul als Device eingefügt.

10.13.1 Globals (ES1320-CB Subsystem)

In diesem Abschnitt werden die globalen Einstellungen des ES1320-CB Subsystems beschrieben.

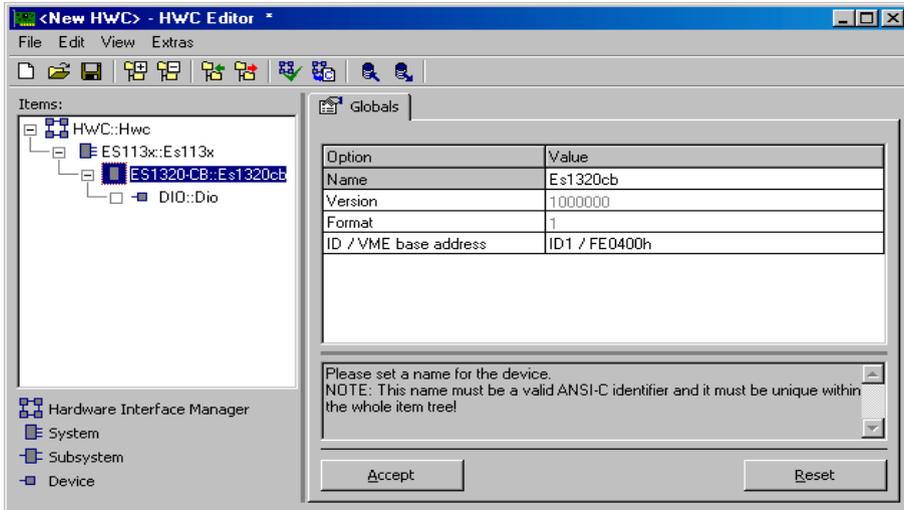


Abb. 10-49 Die Registerkarte „Globals“ des ES1320-CB Subsystems

ID / VME base address

In dieser Zeile wird die Einstellung der VME-Basisadresse vorgenommen. Diese Einstellung muss mit den Jumper-Einstellungen der ES1320 übereinstimmen. Für die ES1320 können acht unterschiedliche VME-Basisadressen gewählt werden (ID1/FE0400h, ID2/FE0C00h, ID3/FE1400h, ID4/FE1C00h, ID5/FE2400h, ID6/FE2C00h, ID7/FE3400h, ID8/FE3C00h, ID9/FE4400h, ID10/FE4C00h, ID11/FE5400h, ID12/FE5C00h, ID13/FE6400h, ID14/FE6C00h, ID15/FE7400h, ID16/FE7C00h), dadurch können bis zu acht ES1320 in einem ETAS Experimentalsystem betrieben werden. Die ES1320 belegt einen Adressbereich von 256 Worten ab der Basisadresse.

10.13.2 Globals (DIO Device)

In diesem Abschnitt werden die globalen Einstellungen des DIO Devices beschrieben.

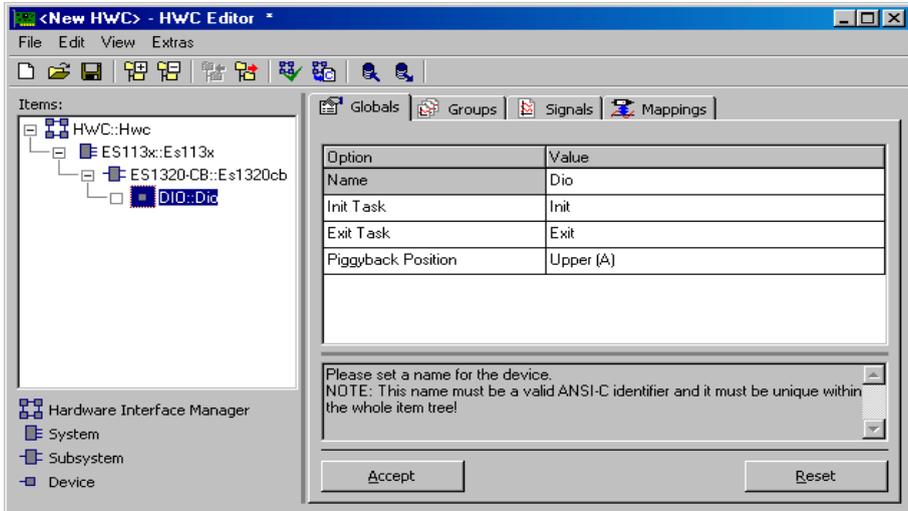


Abb. 10-50 Die Registerkarte „Globals“ des DIO Devices

Init Task

In dieser Zeile wird die Task zur Initialisierung der ES1320 zugeordnet (Type: Init / Application Mode: active).

Exit Task

In dieser Zeile wird die Task zugeordnet, die bei einem Stop des Experiments mit der ES1320 ausgeführt werden soll (Type: Init / Application Mode: inactive).

Piggyback Position

In dieser Zeile wird entweder das obere (Upper (A)) oder das untere (Lower (B)) DIO-Aufsteckmodul als digitale Schnittstelle zugeordnet.

10.13.3 Groups (DIO Device)

In diesem Abschnitt werden die signalgruppenspezifischen Einstellungen des DIO Devices beschrieben.

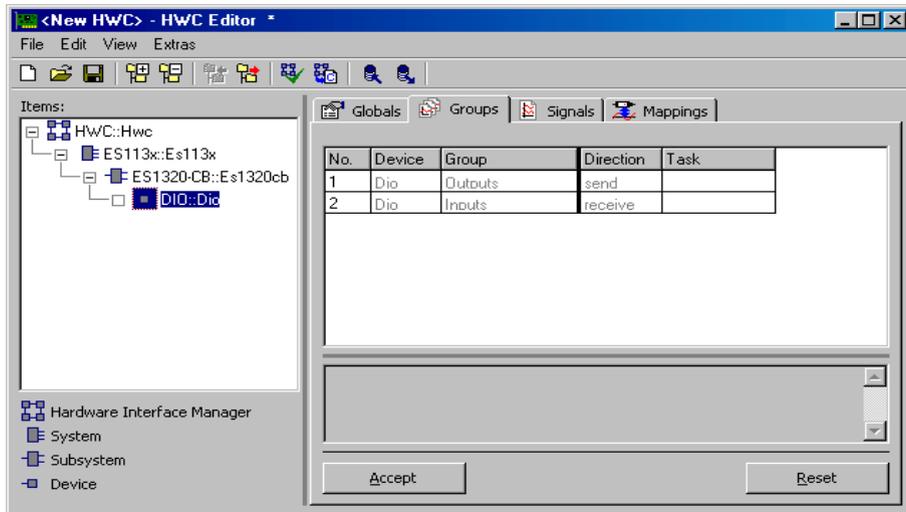


Abb. 10-51 Die Registerkarte „Groups“ des DIO Devices

Für das DIO Device sind in der Registerkarte „Groups“ keine Item-spezifischen Spalten definiert.

10.13.4 Signals (DIO Device)

In diesem Abschnitt werden die signalspezifischen Einstellungen des DIO Devices beschrieben.

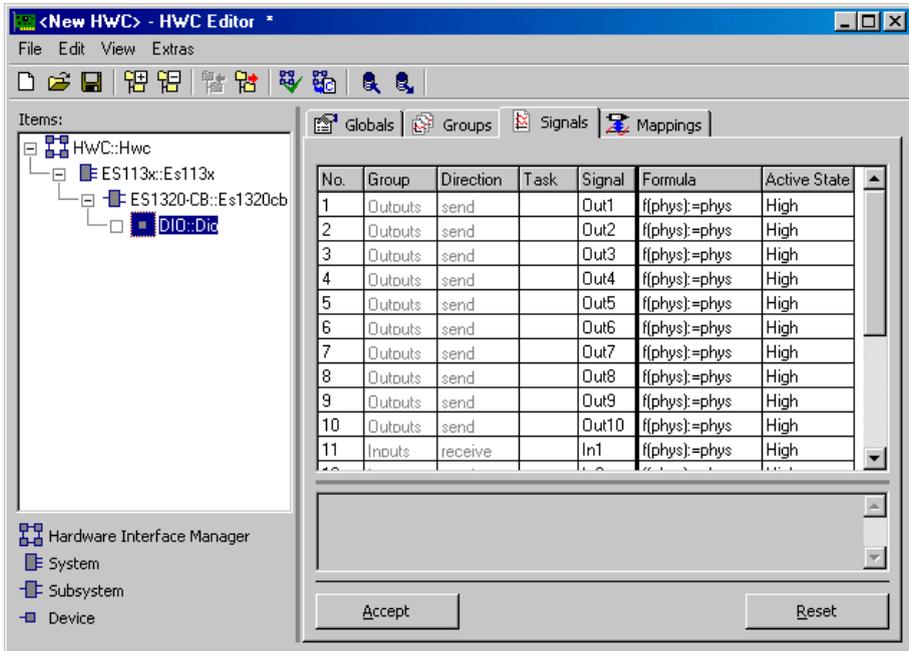


Abb. 10-52 Die Registerkarte „Signals“ des DIO Devices

Active State

In dieser Spalte kann festgelegt werden ob das jeweilige digitale Ein- oder Ausgangssignal durch die Treiberroutine invertiert wird (*Active Low*) oder nicht (*Active High*).

10.13.5 Mappings (DIO Device)

Die Einstellmöglichkeiten sind für alle Items gleich. Sie werden in Kapitel 7.3.5 auf Seite 120 beschrieben.

10.14 ES1325-DIO

Die digitale Ein-/Ausgangskarte ES1325 (DIO) kann auf jeweils sechzehn Kanälen digitale Signale erfassen und ausgeben. Zwei Triggereingänge ermöglichen eine Steuerung der Signalerfassung und der Signalausgabe. Mit einer

ES1130 oder einer ES1135 können gleichzeitig bis zu vier ES1325 Karten betrieben werden. Für jede ES1325 Karte sind maximal ein Input Device, maximal ein Output Device und maximal ein LED Device zulässig.

10.14.1 Globals (ES1325-DIO Subsystem)

In diesem Abschnitt werden die globalen Einstellungen des ES1325-DIO Subsystems beschrieben.

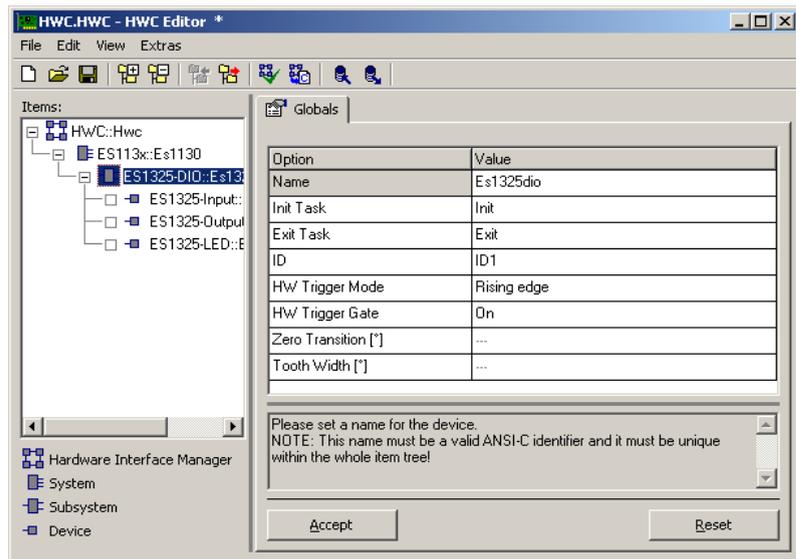


Abb. 10-53 Die Registerkarte „Globals“ des ES1325-DIO Subsystems
ID

Hier wird dem Board eine Kennung für die Boardnummer zugeordnet.

Hinweis

Bei der Karte ES1325 handelt es sich um eine sogenannte „Auto-ID“ Karte, die beim Einschalten des Systems vom Hardware-Manager automatisch eine ID-Nummer und einen freien Adressbereich zugewiesen bekommt. Karten vom selben Typ (z.B. zwei ES1325) werden von links nach rechts nummeriert, d.h. die linke Karte erhält die Nummer Eins, anschließend die Nummer Zwei usw. Es stehen 4 IDs zur Verfügung; somit können bis zu 4 Karten in einem Experimentalsystem betrieben werden.

HW Trigger Mode

In dieser Zeile wird der Triggermodus gesetzt, der den Start von getriggerten Operationen bestimmt. Diese Option ist nur verfügbar, wenn der Parameter „HW Trigger Gate“ auf „On“ gesetzt ist (siehe „HW Trigger Gate“ auf Seite 242).

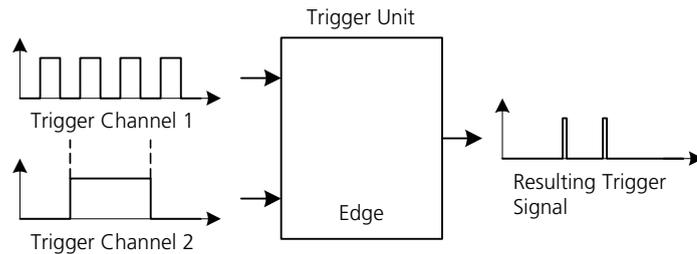
Folgende Modi stehen zur Verfügung:

- **Off**

Signale an den Triggereingängen werden nicht ausgewertet.

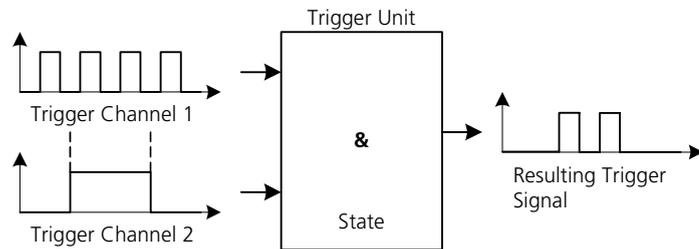
- **Rising edge**

Eine steigende Flanke am Triggereingang 1 startet nach ihrer Bewertung die Erfassung oder Generierung von Signalen an den digitalen Eingangs- und/oder den digitalen Ausgangskanälen.



- **Signal state**

Mit einer steigenden (fallenden) Flanke am Triggereingang 1 wird der Beginn (das Ende) einer Zeitspanne bestimmt. Während dieser Zeitspanne am Triggereingang 1 wird eine Operation an den digitalen Eingangskanälen ausgeführt.



Im Triggermodus „Signal State“ kann eine optionale Verzögerung des Requests (siehe „Signals“ auf Seite 249) genutzt werden. Diese Zeit muß kürzer als die Haltezeit des resultierenden Triggersignals sein, weil sonst die Datenerfassung nicht gestartet wird.

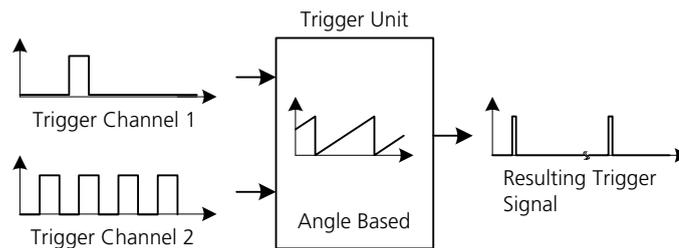
- **Angle based**

Triggereingang 1 und Triggereingang 2 werten beide kontinuierliche PWM Signale aus.

Für Kurbelwellensignale gilt:

- Triggereingang 1 = Nulldurchgang,
- Triggereingang 2 = Zahnbreite.

Eine einzelne PWM Periode am Triggereingang 2 entspricht einem Winkelsegment. Die Winkelposition wird durch Zählung der PWM Perioden ermittelt.



In diesem Betriebsmodus wird ein Zähler auf der Triggereinheit mit beiden Triggereingängen kontrolliert.

Eine steigende Flanke auf Triggerkanal 1 setzt den Zähler auf Null zurück und eine fallende Flanke auf Triggerkanal 2 erhöht den Zähler um 1. So kann zum Beispiel ein Kurbelwellenwinkel erfasst werden, wenn Triggerkanal 2 eine fallende Flanke bei konstanten Winkelsegmenten (z.B. 6° Zahnbreite) empfängt und Triggerkanal 1 eine steigende Flanke mit jeder vollen Umdrehung (im Nullwinkeldurchgang)

empfängt. Um den korrekten Betrieb zu gewährleisten, muss das Signal auf Triggerkanal 2 kontinuierliche und äquidistante Winkelsegmente darstellen. Das Signal auf Triggerkanal 1 muss den Nulldurchgang an der steigenden Flanke darstellen. Die fallende Flanke dieses Signals muss nach einer bestimmten Haltezeit folgen. Diese liegt zwischen der fallenden und der nächsten steigenden Flanke des Signals auf Triggerkanal 2 (Winkelsegmentensignal).

Bei den Triggermodi **Rising edge** und **Signal state** kann der Triggerkanal 2 optional als Triggergate definiert werden (siehe „HW Trigger Gate“ auf Seite 242).

HW Trigger Gate

In dieser Zeile wird der Modus gesetzt, der die Auswertung geeigneter Flanken an den Triggereingängen bestimmt. Der Parameter wird gesetzt, wenn der Triggerkanal 2 zur Freigabe der Triggererzeugung (TTL Hochpegel entspricht "trigger enabled") verwendet wird.

Folgende Modi stehen zur Verfügung:

- **Off**

Jede unter „HW Trigger Mode“ gewählte Flanke am Triggereingang 1 löst eine Operation an den digitalen Eingangs- und/oder Ausgangskanälen aus.

- **On**

Flanken am Triggereingang 1 lösen nur während eines Hochpegels am Triggereingang 2 eine Operation an den digitalen Eingangs- und/oder Ausgangskanälen aus.

Bei Auswahl des Trigger Modes „**Angle based**“ (siehe „HW Trigger Mode“ auf Seite 240) wird der Parameter „HW Trigger Gate“ automatisch auf „**On**“ gesetzt. Bei Auswahl des Trigger Modes „**Off**“ (siehe „HW Trigger Mode“ auf Seite 240) ist der Parameter „HW Trigger Gate“ nicht editierbar.

Zero Transition []*

In dieser Zeile wird der Nulldurchgang am Triggereingang 1 definiert.

Auflösung	Zahnbreite	
Wertebereich	Min.	2 x Zahnbreite
	Max.	32767 x Zahnbreite
Typischer Wert für ein Kurbelwellensignal	360° oder 720°	

Diese Option ist nur im Trigger Mode „**Angle based**“ verfügbar (siehe „HW Trigger Mode“ auf Seite 240). Sie ist innerhalb des Wertebereiches editierbar.

Tooth Width [*]

In dieser Zeile wird die Zahnbreite am Triggereingang 2 während einer PWM Periode definiert.

Auflösung		0,5°
Wertebereich	Min.	0,5°
	Max.	720°
Typischer Wert für ein Kurbelwellensignal		6°

Diese Option ist nur im Trigger Mode „**Angle based**“ verfügbar (siehe „HW Trigger Mode“ auf Seite 240). Sie ist innerhalb des Wertebereiches editierbar.

10.14.2 Globals (ES1325-Input Device)

In diesem Abschnitt werden die globalen Einstellungen des Input Devices beschrieben.

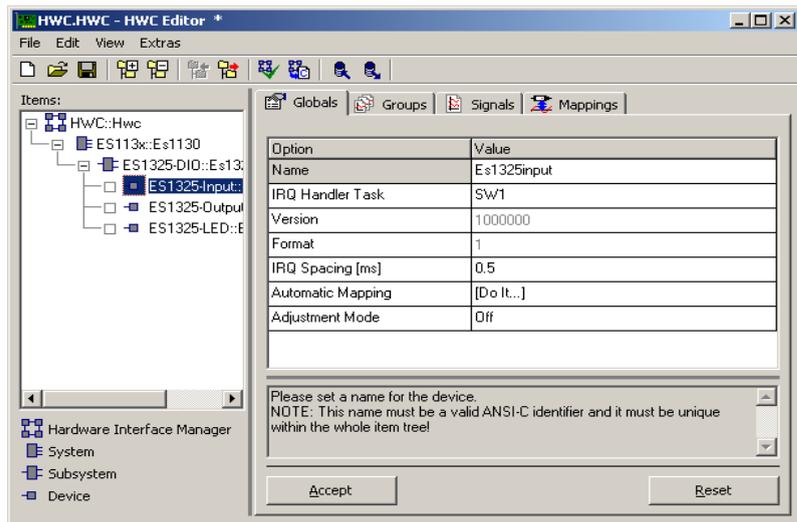


Abb. 10-54 Die Registerkarte „Globals“ des ES1325-DIO Input Devices

IRQ Handler Task

Die IRQ Handler-Task (Type: Software / Application Mode: active) wird benötigt, wenn Signalkanäle im Interrupt-Betrieb ausgewertet werden sollen.

Diese Task muss im ASCET-Projekteditor in der Task-Liste als „Software“-Task angelegt werden, wobei bei „Max. No. of Activations“ mindestens die Anzahl 2 (maximal 50) einzutragen ist.

Hinweis

Diese Task muss vom Typ „Software“ sein und **ausschließlich** für das Element bzw. den betreffenden Hardware-Treiber zur Verfügung stehen und darf von keinem anderen Element bzw. von Anwender-Prozessen mitbenutzt werden. Auch zwei gleichartige Elemente (z.B. zwei ES1325) dürfen sich **nicht** die gleiche Task teilen!

Diese Option ist gesperrt, wenn der Triggermodus **OFF** ausgewählt ist (siehe „HW Trigger Mode“ auf Seite 240).

IRQ Spacing [ms]

In dieser Zeile wird die minimale Zeit zwischen zwei von der ES1325 generierten Interruptanforderungen definiert.

Auflösung		0,1 ms
Wertebereich	Min.	0 ms
	Max.	25,5 ms

Diese Option ist nicht im Trigger Mode „**OFF**“ verfügbar (siehe „HW Trigger Mode“ auf Seite 240). Sie ist innerhalb des Wertebereiches editierbar.

Automatic Mapping

Diese Option ermöglicht die automatische Zuordnung zwischen Signalen und (ASCET) Messages.

Nach Drücken von [Do it . . .] öffnet sich folgendes Dialogfenster, welches das gegenwärtige Mapping aufzeigt:

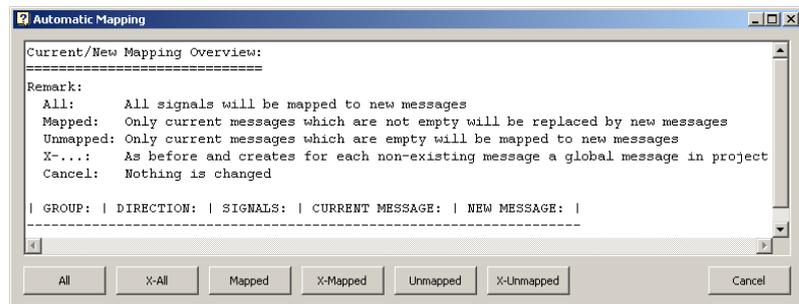


Abb. 10-55 Dialogfenster „Automatic Mapping“ mit Mappingstatus

Vom Fenster aus kann nun je nach Bedarf das weitere Vorgehen ausgewählt werden. Folgende Möglichkeiten stehen zur Verfügung:

- **All:**
Das vorhandene Mapping wird komplett ersetzt. Dabei wird so vorgegangen, dass für jedes Signal eine geeignete namensgleiche ASCET Message gesucht wird. Ist eine vorhanden, so wird sie eingetragen; falls nicht, so wird das Signal nicht abgebildet.
- **X-All:**
Funktioniert wie „All“, nur wird bei dieser erweiterten Funktionalität zusätzlich für jede fehlende ASCET Message eine globale Send-Receive Message angelegt. D.h. nach dieser Aktion sind grundsätzlich alle Signale abgebildet.
- **Mapped:**
Es werden nur die Signale neu abgebildet, die bereits abgebildet waren, d.h. für jedes abgebildete Signal wird eine namensgleiche ASCET Message gesucht. Wird keine gefunden, so wird das entsprechende Signal nicht mehr abgebildet (= leere ASCET Message).
- **X-Mapped:**
Funktioniert wie „Mapped“, nur werden bei dieser erweiterten Aktion ggf. neue globale Send-Receive Messages angelegt, falls keine namensgleichen ASCET Messages gefunden wurden.
- **Unmapped:**
Es werden nur die Signale neu abgebildet, die bisher nicht abgebildet waren (= leere ASCET Message), d.h. für jedes dieser Signale wird eine namensgleiche ASCET Message gesucht. Wird keine gefunden, so bleibt das entsprechende Signal nicht abgebildet.
- **X-Unmapped:**
Funktioniert wie „Unmapped“, nur werden mit dieser erweiterten Aktion ggf. neue globale Send-Receive-Messages angelegt, falls keine namensgleiche ASCET Message gefunden wurden.
- **Cancel:**
Bricht die automatische Zuordnung ab.

Ob die mit **X-*** generierten Messages im Projekt oder in einem der zum Projekt gehörigen Module generiert werden, legen Sie im Register „Mapping“ im Optionsfenster des HWC-Editors (s. Seite 108) fest.

Adjustment Mode

Der Adjustment Mode erlaubt bei Bedarf die Beeinflussung der Signalerfassung online, d.h. während der Laufzeit des ASCET-Modells.

Folgende Modi stehen zur Verfügung:

- **Off**

Der Adjustment Mode ist ausgeschaltet. Es wird für jeden Eingangskanal „ChXX“ eine Messgruppe „ChXXMr“ angezeigt.

- **On**

Der Adjustment Mode ist eingeschaltet. Für jeden Eingangskanal „ChXX“ wird zusätzlich zur Meßgruppe „ChXXMr“ eine Online-Justiergruppe „ChXXAdj“ eingeblendet. In dieser Gruppe „ChXXAdj“ sind zusätzlich online veränderbare Konfigurationsparameter in der Spalte „Signals“ im Register „Groups“ verfügbar.

Der Adjustment Mode kann nur ausgewählt werden, wenn der HW Trigger Mode auf „On“ gesetzt ist (siehe „HW Trigger Mode“ auf Seite 240).

10.14.3 Groups (ES1325-Input Device)

In diesem Abschnitt werden die signalgruppenspezifischen Einstellungen des Input Devices beschrieben.

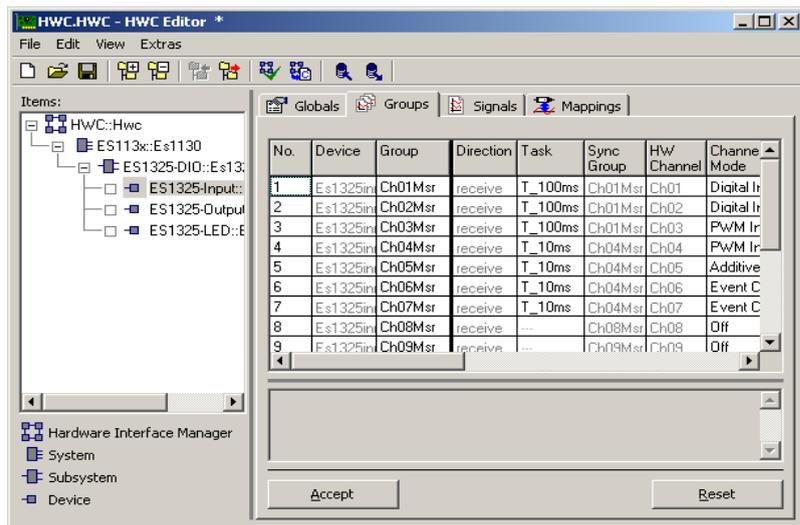


Abb. 10-56 Die Registerkarte „Groups“ des ES1325-DIO Input Devices

Sync Group

Signalgruppen, die in der gleichen Task (oder einer Gruppe von Tasks) synchron übertragen werden, werden zu einem Prozess kombiniert. Dieser Prozess wird nach dem in dieser Spalte angezeigten Gruppennamen benannt.

Bei einer Auswahl von mehreren Tasks können nur die Prozesse derjenigen Signalgruppen in einem gemeinsamen Prozeß zusammengefaßt werden, deren Auswahl im Feld „Task“ identisch ist.

Die Signalgruppen innerhalb dieses Prozesses werden synchron übertragen.

Hardware Channel

Eingangskanäle der ES1325 (Ch01 bis Ch16)

Channel Mode

In dieser Spalte wird die Art des auszuwertenden digitalen Eingangssignals für jeden Eingangskanal der ES1325.1 bestimmt:

- **Off**
Keine Auswertung des digitalen Eingangssignals.
- **Digital Input**
Auswertung des Zustands des digitalen Eingangssignales zum Zeitpunkt des Requests.
- **PWM Input**
Auswertung aller relevanten Informationen des letzten vollständigen PWM Signals vor dem Request.
- **Additive Time**
Addition der Dauer der aktiven Phasen eines Signals während eines definierten Zeitintervalls.
- **Event Counter**
Addition relevanter Signalfanken während eines definierten Zeitintervalls.

Die Festlegung, ob steigende, fallende oder beide Flanken ausgewertet werden sollen, erfolgt im Parameter „Significant Edge“.

Die Übertragung der Daten ist in den Kanälen nur dann aktiviert, wenn entweder eine Task ausgewählt ist oder der Parameter „IRQ“ auf „**Yes**“ gesetzt ist (siehe „IRQ“ auf Seite 248), sofern dieser verfügbar ist.

Use HW Trigger

Hier wird festgelegt, ob das digitale Eingangssignal synchron mit einem externen Triggersignal erfasst werden soll. Folgende Modi stehen zur Verfügung:

- **Yes**
Die Erfassung der digitalen Signale wird mit einem externen Trigger synchronisiert.

- **No**

Die Erfassung der digitalen Signale erfolgt im Polling Mode. Das digitale Eingangssignal wird dabei nicht mit einem externen Triggersignal synchronisiert.

Der Parameter „Trigger Mode“ wird für alle Hardwarekanäle gemeinsam in „Globals“ für das ES1325-DIO Subsystem festgelegt.

IRQ

In dieser Zeile wird festgelegt, ob nach der Signalerkennung ein Interrupt erzeugt wird.

- **Yes**

Eine IRQ Handler Task (siehe „IRQ Handler Task“ auf Seite 243) wird aktiviert. Nach der Datenerfassung wird eine Interruptanforderung zur Übernahme der Daten in das ASCET-Modell generiert.

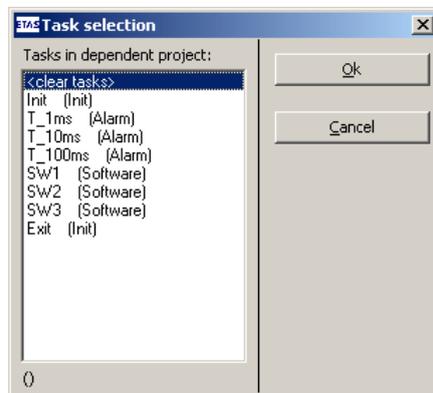
- **No**

Die erfassten Signale stehen in einem Pufferspeicher zur Verarbeitung zur Verfügung. Die Datenübernahme erfolgt ausgelöst durch das ASCET-Modell im Polling Mode innerhalb der im Feld „Task“ festgelegten Task.

Die Option „IRQ“ ist nur aktiv, wenn der Parameter „Use HW Trigger“ auf „Yes“ gesetzt ist (siehe „Use HW Trigger“ auf Seite 248).

Activated Task

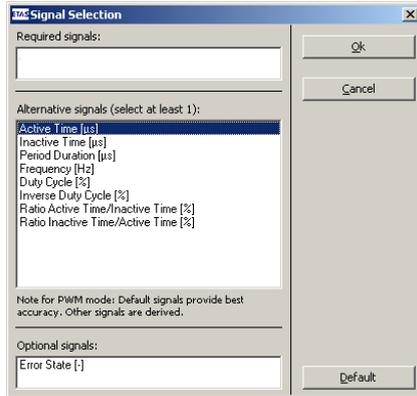
Klicken im Feld „Activated Task“ öffnet das Auswahlfenster „Task Selection“. Dort wird die Task kanalspezifisch ausgewählt und zugeordnet, die bei Empfang einer Signalgruppe aktiviert wird.



Diese Option ist nur aktiv, wenn der Parameter „IRQ“ auf „**yes**“ gesetzt ist (siehe „IRQ“ auf Seite 248).

Signals

Klicken der Schaltfläche [Select] im Feld „Signals“ öffnet das Auswahlfenster „Signal Selection“.



Jedem Signal werden Signalkomponenten zugeordnet.

Für die Messgruppen „ChXXMr“ stehen folgende Signale zur Verfügung:

- State
- Active Time [µs]
- Inactive Time [µs]
- Period Duration [µs]
- Frequency [Hz]
- Duty Cycle [%]
- Ratio Active Time/ Inactive Time [%]
- Ratio Inactive Time/ Active Time [%]
- Additive Active Time
- Counter Value

Für die Online-Justiergruppen „ChXXAdj“ (siehe „Adjustment Mode“ im Register „Globals“, Seite 246) stehen folgende Signale zur Verfügung:

- Time Delay [µs]
- Angle Delay [µs]
- Angle Interval [µs]

In Abhängigkeit von den Parametern „Channel Mode“, „Use HW Trigger“ und „HW Trigger Mode“ sind einige Signale obligatorisch oder es werden Standardwerte für die Signalgruppe selektiert.

Active State

Hier wird einem Pegel des Eingangssignals die Funktion des aktiven Status zugeordnet. Folgende Modi stehen zur Verfügung:

- High
Dem Hochpegel des Eingangssignals wird für die weitere Verarbeitung der Status „aktiv“ zugeordnet.
- Low
Dem Tiefpegel des Eingangssignals wird für die weitere Verarbeitung der Status „aktiv“ zugeordnet.

Significant Edge

Hier wird einer Flanke ein Ereignis (Channel Mode „**Event Counter**“) oder der Start einer PWM Periode (Channel Mode „**PWM Input**“) zugeordnet. Folgende Modi stehen zur Verfügung:

- Inactive-Active
Der Übergang des Eingangssignals vom inaktiven in den aktiven Zustand (siehe „Active State“ auf Seite 251) löst das Ereignis aus.
- Active-Inactive
Der Übergang des Eingangssignals vom aktiven in den inaktiven Zustand (siehe „Active State“ auf Seite 251) löst das Ereignis aus.
- Both
Übergänge des Eingangssignals vom inaktiven in den aktiven Zustand und vom aktiven in den inaktiven Zustand (siehe „Active State“ auf Seite 251) lösen das Ereignis aus.

Diese Einstellung ist für die Channel Modi „**Digital Input**“ und „**Additive Time**“ (siehe „Channel Mode“ auf Seite 247) nicht verfügbar.

Hysteresis

Jedem digitalen Eingangskanal muß eine Hystereseart zugeordnet werden. Vorhandene Optionen:

- TTL

- User defined

	TTL	User defined
Unterer Schwellwert	1,728 V	anwenderabhängig
Oberer Schwellwert	2,304 V	anwenderabhängig

Bei Auswahl der Option „**User defined**“ müssen der untere und der obere Schwellwert der Hysterese entsprechend der Definition festgelegt werden (siehe „Low Thresh. [V]“ auf Seite 252 und „High Thresh. [V]“ auf Seite 252).

Low Thresh. [V]

In dieser Zelle wird jedem digitalen Eingangskanal der untere Schwellwert für das Eingangssignal zugeordnet. Der untere Schwellwert muß mindestens einen Schritt unterhalb des oberen Schwellwerts definiert werden.

Auflösung		0,144 V
Wertebereich	Min.	0 V
	Max.	oberer Schwellwert - 0,144 V

Diese Option ist gesperrt, wenn der Hysterese Mode „**TTL**“ ausgewählt ist (siehe „Hysterese“ auf Seite 251).

High Thresh. [V]

In dieser Zelle wird jedem digitalen Eingangskanal der obere Schwellwert für das Eingangssignal zugeordnet. Der obere Schwellwert muß mindestens einen Schritt oberhalb des unteren Schwellwerts definiert werden.

Auflösung		0,144 V
Wertebereich	Min.	unterer Schwellwert + 0,144 V
	Max.	36 V

Diese Option ist gesperrt, wenn der Hysterese Mode „**TTL**“ ausgewählt ist (siehe „Hysterese“ auf Seite 251).

Timeout [ms]

In dieser Zelle wird jedem digitalen Eingangskanal ein Zeitraum zugeordnet, in dem das Eingangssignal mindestens einmal den Zustand wechseln muß. Erfolgt kein Zustandswechsel des Signals innerhalb dieses definierten Zeitraums, wird der zugehörige Fehlerstatus generiert.

Auflösung		2,5 ms
Wertebereich	Min.	0 ms (kein Timeout zugeordnet)
	Max.	163837,5 ms

10.14.4 Signals (ES1325-Input Device)

In diesem Abschnitt werden die signalspezifischen Einstellungen des Input Devices beschrieben.

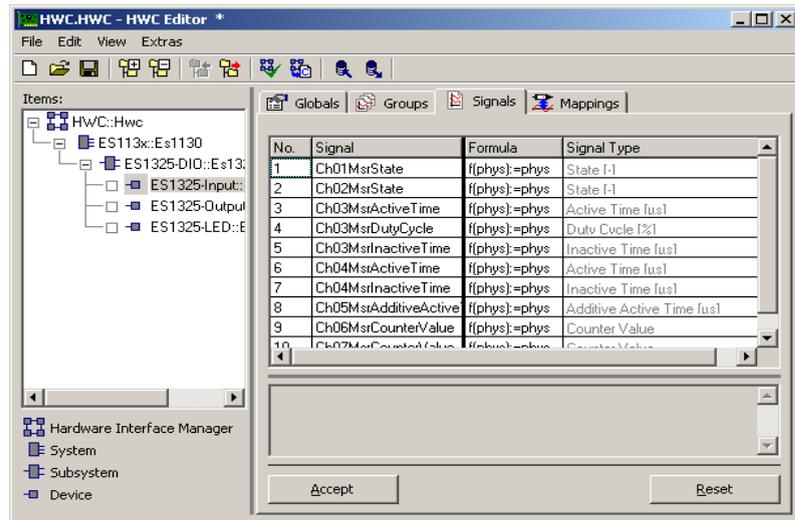


Abb. 10-57 Die Registerkarte „Signals“ des ES1325-DIO Input Devices
HW Channel

Anzeige der Eingangskanäle der ES1325.1 (Ch01 bis Ch16).

Signal Type

Hier wird der Signaltyp angezeigt, der diesem Signal zugeordnet ist. Im Register „Groups“, Parameter „Signals“ kann diese Zuordnung verändert werden.

10.14.5 Mappings (ES1325-Input Device)

Die Einstellmöglichkeiten werden in Kapitel 7.3.5 auf Seite 120 beschrieben.

10.14.6 Globals (ES1325-Output Device)

In diesem Abschnitt werden die globalen Einstellungen des Output Devices beschrieben.

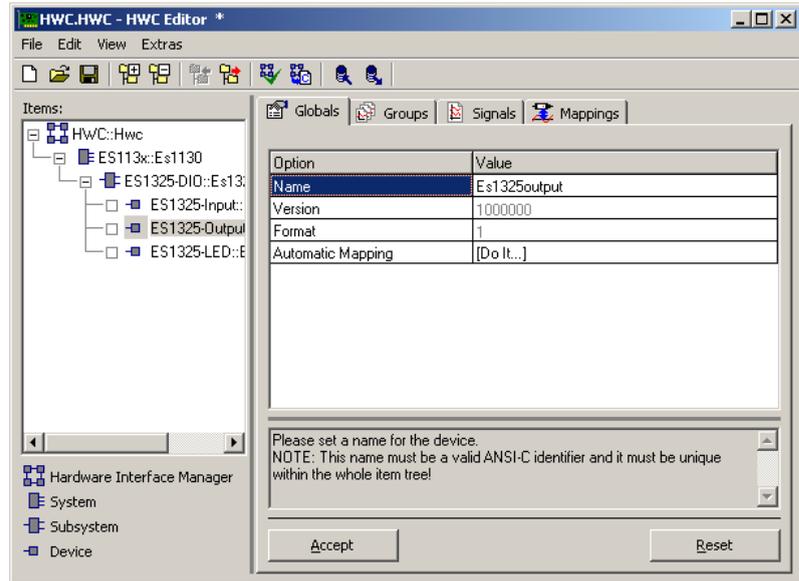


Abb. 10-58 Die Registerkarte „Globals“ des ES1325-DIO Output Devices
Automatic Mapping

Diese Option ermöglicht die automatische Zuordnung zwischen Signalen und (ASCET) Messages. Die Zuordnung erfolgt auf die gleiche Art wie beim Input Device (siehe Abschnitt „Automatic Mapping“ auf Seite 244).

10.14.7 Groups (ES1325-Output Device)

In diesem Abschnitt werden die signalgruppenspezifischen Einstellungen des Output Devices beschrieben.

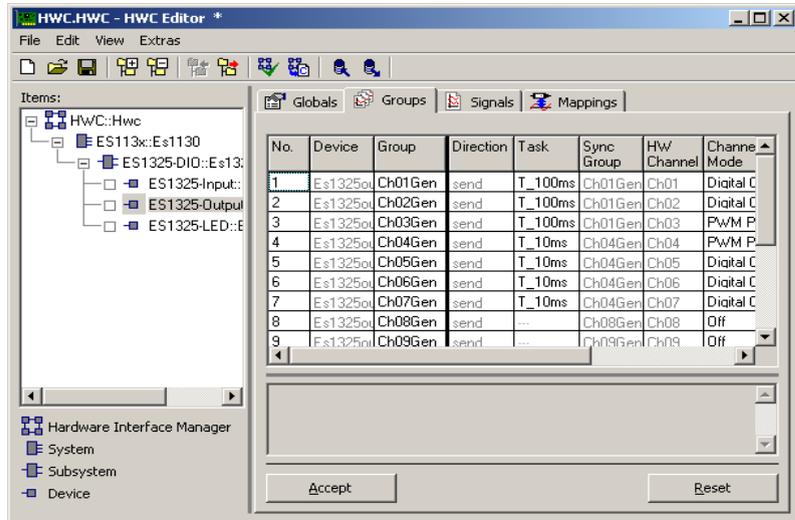


Abb. 10-59 Die Registerkarte „Groups“ des ES1325-DIO Output Devices
Sync Group

Signalgruppen, die in der gleichen Task (oder einer Gruppe von Tasks) synchron übertragen werden, werden zu einem Prozess kombiniert. Dieser Prozess wird nach dem in dieser Spalte angezeigten Gruppennamen benannt.

Bei einer Auswahl von mehreren Tasks können nur die Prozesse derjenigen Signalgruppen in einem gemeinsamen Prozeß zusammengefaßt werden, deren Auswahl im Feld „Task“ identisch ist.

Die Signalgruppen innerhalb dieses Prozesses werden synchron übertragen.

HW Channel

Ausgangskanäle der ES1325.1 (Ch01 bis Ch16).

Channel Mode

In dieser Spalte wird die Art des auszuwertenden digitalen Ausgangssignals für jeden Ausgangskanal der ES1325.1 bestimmt:

- Off
Keine Auswertung des digitalen Ausgangssignals.

- Digital Output

Bei Request kann der Zustand eines Ausgangskanals auf aktiv oder inaktiv gesetzt werden. Der Request kann vom Simulationsprozessor ausgelöst werden.

- PWM Periodic Output

Es werden PWM-Signale ohne Begrenzung der Ausgabedauer generiert.

Zur Beschreibung der generierten PWM-Signale sind genau zwei Signalkomponenten erforderlich (siehe „Signals“ auf Seite 256).

- PWM Interval Output

Die PWM-Funktionalität wird verwendet, um Einzelpulse oder Intervalle von PWM-Signalen zu generieren. Eine weitere Anforderung, die während der Abarbeitung der ersten Anforderung ankommt, wird verworfen.

Die Übertragung der Daten ist in den Kanälen nur dann aktiviert, wenn entweder eine Task ausgewählt ist oder der Parameter „IRQ“ auf „Yes“ gesetzt ist (siehe „IRQ“ auf Seite 248), sofern dieser verfügbar ist.

Use HW Trigger

Hier wird festgelegt, ob das digitale Ausgangssignal synchron mit einem externen Triggersignal ausgegeben werden soll. Folgende Modi stehen zur Verfügung:

- **Yes**

Die Ausgabe der digitalen Signale wird mit einem externen Trigger synchronisiert.

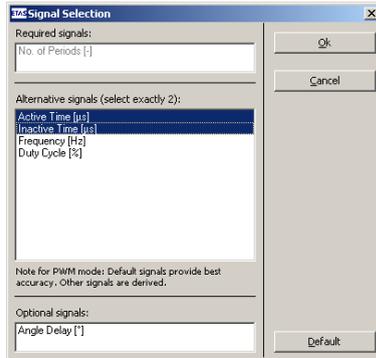
- **No**

Die Ausgabe der digitalen Signale erfolgt im Polling Mode. Das digitale Ausgangssignal wird dabei nicht mit einem externen Triggersignal synchronisiert.

Der Parameter „Trigger Mode“ wird für alle Hardwarekanäle gemeinsam in „Globals“ für das ES1325-DIO Subsystem festgelegt.

Signals

Klicken der Schaltfläche [Select] im Feld „Signals“ öffnet das Auswahlfenster „Signal Selection“. Die zu generierenden Signale können ausgewählt werden.



Jedem Signal werden Signalkomponenten zugeordnet.

- State [-]
- Active Time [µs]
- Inactive Time [µs]
- Frequency [Hz]
- Duty Cycle [%]
- Time Delay [µs]
- No. of Periods [-]
- Angle Delay [°]

In Abhängigkeit von den Parametern „Channel Mode“, „Use HW Trigger“ und „HW Trigger Mode“ sind einige Signale obligatorisch oder es werden Standardwerte für die Signalgruppe selektiert.

Active State

Hier wird einem Pegel des Ausgangssignals die Funktion des aktiven Status zugeordnet. Folgende Modi stehen zur Verfügung:

- High
Dem Hochpegel des Ausgangssignals wird für die weitere Verarbeitung der Status „aktiv“ zugeordnet.
- Low
Dem Tiefpegel des Ausgangssignals wird für die weitere Verarbeitung der Status „aktiv“ zugeordnet.

10.14.8 Signals (ES1325-Output Device)

In diesem Abschnitt werden die signalspezifischen Einstellungen des Output Devices beschrieben.

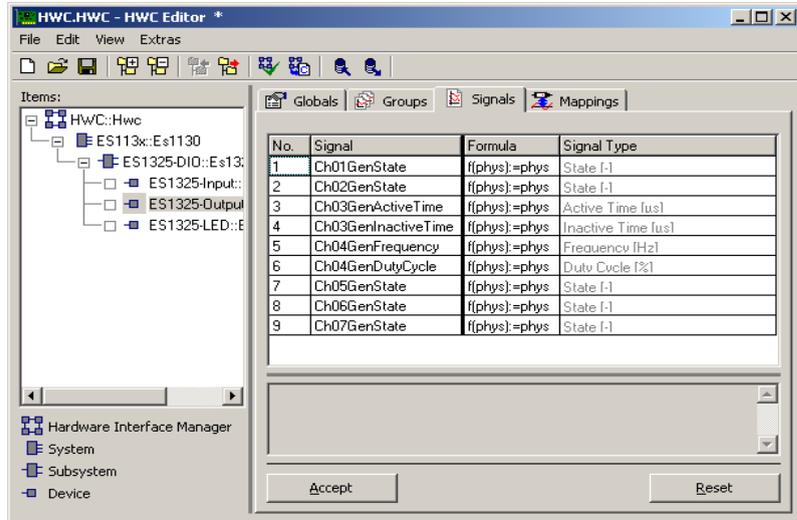


Abb. 10-60 Die Registerkarte „Signals“ des ES1325-DIO Output Devices
HW Channel

Ausgangskanäle der ES1325.1 (Ch01 bis Ch16).

Signal Type

Hier wird der Signaltyp angezeigt, der diesem Signal zugeordnet ist. Im Register „Groups“, Parameter „Signals“ kann diese Zuordnung verändert werden.

10.14.9 Mappings (ES1325-Output Device)

Die Einstellmöglichkeiten sind in Kapitel 7.3.5 auf Seite 120 beschrieben.

10.14.10 Globals (ES1325-LED Device)

In diesem Abschnitt werden die globalen Einstellungen des LED Devices beschrieben.

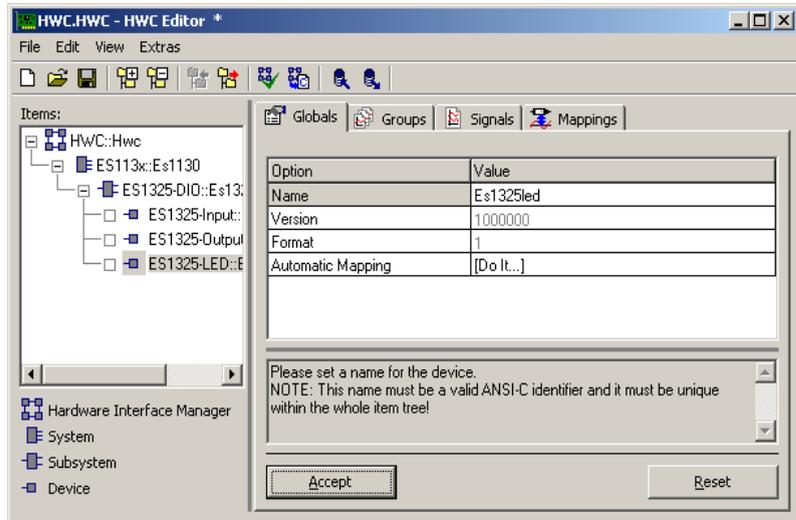


Abb. 10-61 Die Registerkarte „Globals“ des ES1325-DIO LED Devices
Automatic Mapping

Diese Option ermöglicht die automatische Zuordnung zwischen Signalen und (ASCET) Messages. Die Zuordnung erfolgt auf die gleiche Art wie beim Input Device (siehe Abschnitt „Automatic Mapping“ auf Seite 244).

10.14.11 Groups (ES1325-LED Device)

In diesem Abschnitt werden die signalgruppenspezifischen Einstellungen des LED Devices beschrieben.

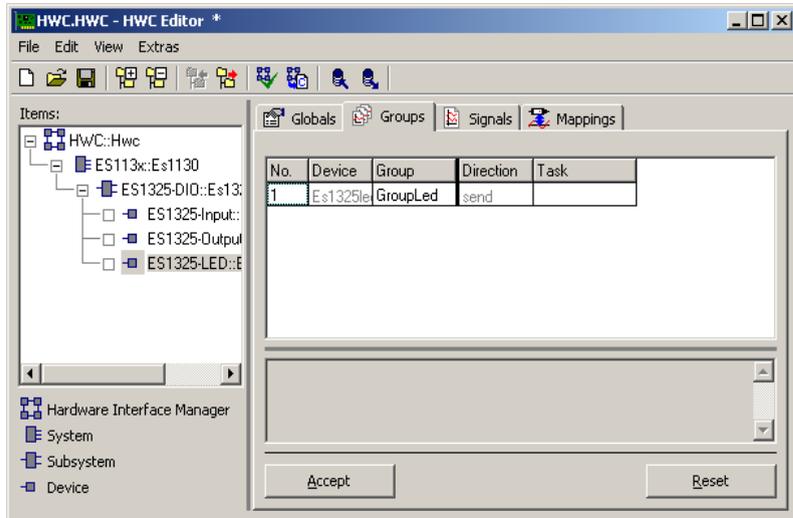


Abb. 10-62 Die Registerkarte „Groups“ des ES1325-DIO LED Devices

Für das ES1325-LED Device sind in der Registerkarte „Groups“ keine Item-spezifischen Spalten definiert. Die Einstellmöglichkeiten der Items werden in Kapitel 7.3.3 auf Seite 117 beschrieben.

Task

In diesem Feld muß die zugehörige Task ausgewählt werden.

10.14.12 Signals (ES1325-LED Device)

In diesem Abschnitt werden die signalspezifischen Einstellungen des LED Devices beschrieben.

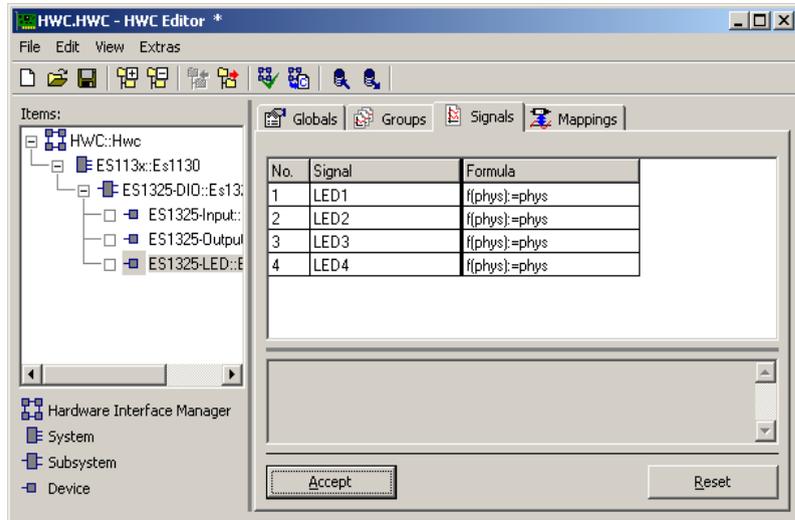


Abb. 10-63 Die Registerkarte „Signals“ des ES1325-DIO LED Devices

Für das ES1325-LED Device sind in der Registerkarte „Signals“ keine Item-spezifischen Spalten definiert. Die Einstellmöglichkeiten der Items werden in Kapitel 7.3.4 auf Seite 118 beschrieben.

10.14.13 Mappings (ES1325-LED Device)

Für das ES1325-LED Device sind in der Registerkarte „Mappings“ keine Item-spezifischen Spalten definiert. Die Einstellmöglichkeiten der Items werden in Kapitel 7.3.5 auf Seite 120 beschrieben.

10.15 ES1330-PWM

Die Zählerkarte ES1330 ermöglicht die Erfassung und Ausgabe von PWM-Signalen. Die ES1330 ist mit sechs Zählerbausteinen (Am9513A) bestückt. Jeder Zählerbaustein stellt fünf 16-Bit breite Zähler mit einer Zählfrequenz von 4 MHz zur Verfügung. Dreißig Zählereingänge und sechzehn Zählerausgänge sind auf der ES1330 in sechs interne Ports „Port 1 .. Port 6“ aufgeteilt. Jeder Zählerbaustein belegt einen dieser Ports. Der Zugriff auf die Ports geschieht

über den SUB-D Stecker der ES1330. In der Items-Liste des HWC-Editors wird die Zählerkarte ES1330 als Subsystem und der Zählerbaustein als Device eingefügt.

Hinweis

Die Tabelle im Kapitel „Steckverbinder X1 digitale Ein- und Ausgänge“ des Handbuchs zur Karte ES1330 (im Ordner ETASManuals\ASCET5.1\ES1000 Ihrer ASCET-Installation) enthält die möglichen Kombinationen von Ports und Ein-/Ausgängen der Zähler.

10.15.1 Globals (ES1330-PWM Subsystem)

In diesem Abschnitt werden die globalen Einstellungen des ES1330-PWM Subsystems beschrieben.

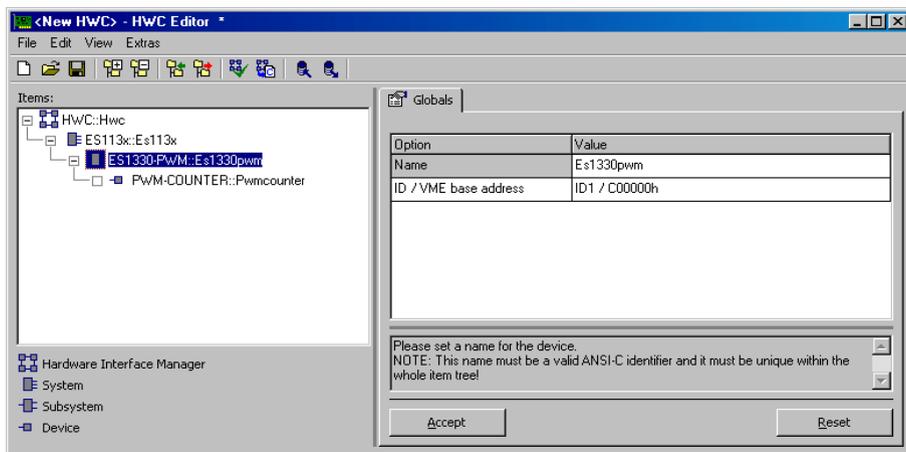


Abb. 10-64 Die Registerkarte „Globals“ des ES1330-PWM Subsystems
ID / VME base address

In dieser Zeile wird die Einstellung der VME-Basisadresse vorgenommen. Diese Einstellung muss mit den Jumper-Einstellungen der ES1330 übereinstimmen. Für die ES1330 können vier unterschiedliche VME-Basisadressen gewählt werden (ID1 / C00000h, ID2 / C00100h, ID3 / C00200h, ID4 / C00300h), dadurch können bis zu vier ES1330 in einem ETAS Experimentalsystem betrieben werden. Die ES1330 belegt einen Adressbereich von 256 Worten ab der Basisadresse.

10.15.2 Globals (PWM-COUNTER Device)

In diesem Abschnitt werden die globalen Einstellungen des PWM-COUNTER Devices beschrieben.

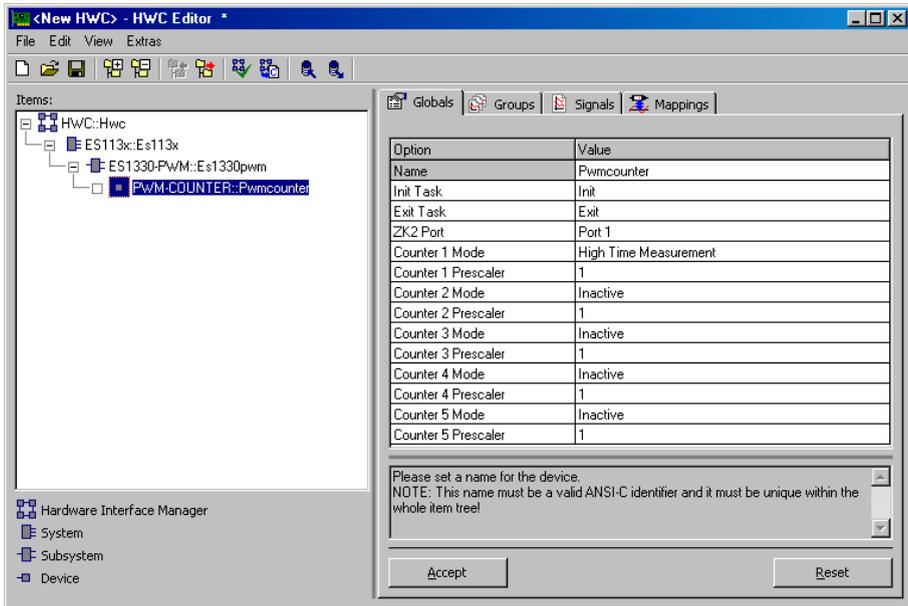


Abb. 10-65 Die Registerkarte „Globals“ des PWM-COUNTER Devices

Init Task

In dieser Zeile wird die Task zur Initialisierung der ES1330 zugeordnet (Type: Init / Application Mode: active).

Exit Task

In dieser Zeile wird die Task zugeordnet, die bei einem Stopp des Experiments mit der ES1330 ausgeführt werden soll (Type: Init / Application Mode: inactive).

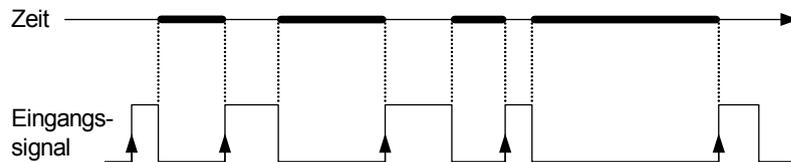
ZK2 Port

In dieser Zeile wird der Port (und damit der Zählerbaustein) zugeordnet. Jeder Zählerbaustein belegt einen eigenen Port.

Counter 1 Mode .. Counter 5 Mode

In diesen Zeilen wird der Zählermodus für jeden der fünf Zähler des Zählerbausteins gewählt. Die folgende Liste zeigt, welche Zählermodi ausgewählt werden können.

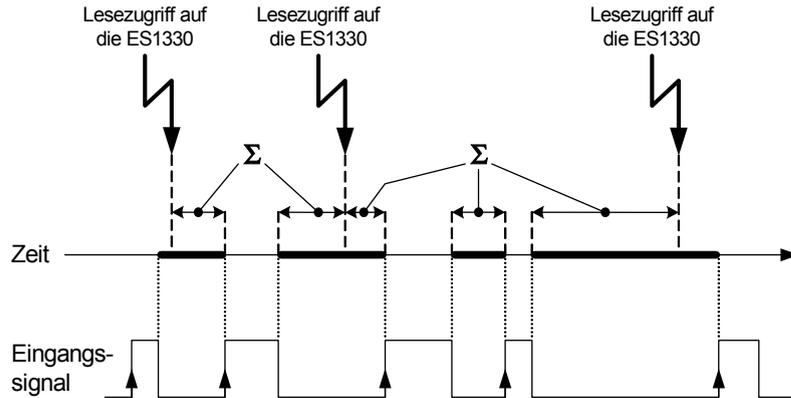
- Inactive
Der Zähler wird nicht verwendet.
- PWM-Generator (PWM_Gen)
Der Zähler wird genutzt, um PWM-Signale zu erzeugen.
- Period-Measurement (P_Meas)
Der Zähler wird genutzt um die Periode eines PWM-Signals zu messen.
- High Time Measurement
Bei diesem Modus wird die Dauer eines jeden Pulses mit aktivem Pegel (hier: high-aktiv) gemessen.
- Low Time Measurement
Bei diesem Modus wird die Dauer eines jeden Pulses mit aktivem Pegel (hier: low-aktiv) gemessen.



- Additive High Time Measurement
Der Simulationsprozessor erfasst Messwerte von der ES1330-Karte in der Regel in einem periodischen Raster. Bei den additiven Messungen wird die Zeit gemessen, in der das Signal zwischen zwei aufeinanderfolgenden Leseoperationen auf das Messregister aktiv (hier: high-aktiv) war.

- Additive Low Time Measurement

Der Simulationsprozessor erfasst Messwerte von der ES1330-Karte in der Regel in einem periodischen Raster. Bei den additiven Messungen wird die Zeit gemessen, in der das Signal zwischen zwei aufeinanderfolgenden Leseoperationen auf das Messregister aktiv (hier: low-aktiv) war.



Counter 1 Prescaler .. Counter 5 Prescaler

In diesen Zeilen kann ein Vorteiler für die Eingangsfrequenz von 4 MHz genutzt werden. Die geteilte Eingangsfrequenz dient dann als Taktsignal für die ersten vier Zähler. Der Vorteiler ermöglicht das Erzeugen und Auswerten von PWM-Signalen mit einer längeren Periodendauer, allerdings nimmt dadurch die Auflösung der PWM-Signale ab. Folgende Vorteiler können gewählt werden: 1, 10, 100, 1000 und 10000.

10.15.3 Groups (PWM-COUNTER Device)

In diesem Abschnitt werden die signalgruppenspezifischen Einstellungen des PWM-COUNTER Devices beschrieben.

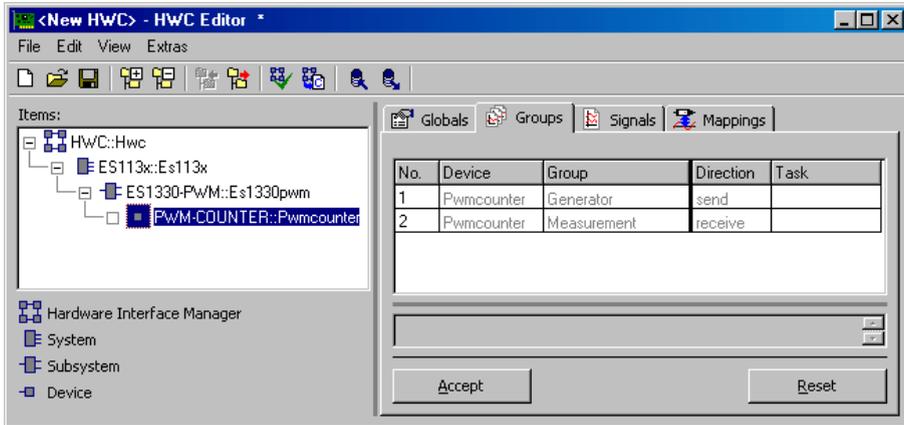


Abb. 10-66 Die Registerkarte „Groups“ des PWM-COUNTER Devices
Für das ES1330-PWM Subsystem sind in der Registerkarte „Groups“ keine Item-spezifischen Spalten definiert.

10.15.4 Signals (PWM-COUNTER Device)

In diesem Abschnitt werden die signalspezifischen Einstellungen des PWM-COUNTER Devices beschrieben.

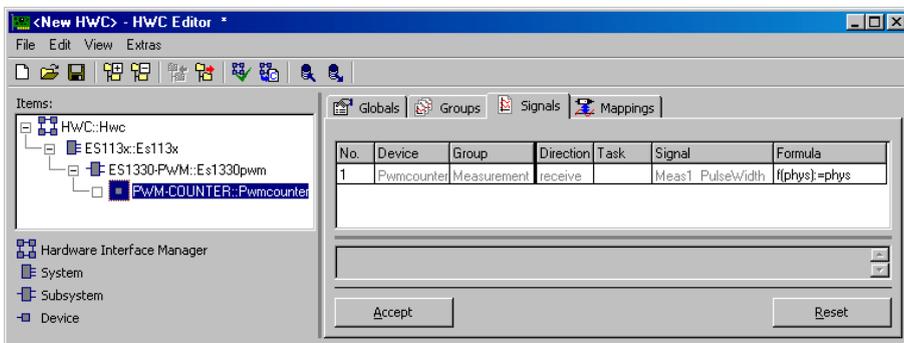


Abb. 10-67 Die Registerkarte „Signals“ des PWM-COUNTER Devices

No.

Die Anzahl und Bedeutung der Signale wird durch die „ZK2 Port und Counter X Mode“ Einstellungen aus der Registerkarte „Globals“ festgelegt.

Signal

In dieser Spalte sind Namen (ANSI-C) für die Signale fest vorgegeben. Es gibt folgende Namenskonvention (X = Nummer des Zählers):

- `GenX_Frequency`
Frequenz des PWM-Ausgabekanals in Hz.
- `GenX_DutyCycle`
Tastverhältnis des PWM-Ausgabekanals in %.
- `MeasX_Period`
Periodendauer des PWM-Auswertekanals in sec..
- `MeasX_PulseWidth`
Ergebnis der Pulsweitenmessung (Einzelpulse oder additiv).

10.15.5 Mappings (PWM-COUNTER Device)

Die Einstellmöglichkeiten sind in Kapitel 7.3.5 auf Seite 120 beschrieben.

Das Tutorial beschreibt anhand von mitgelieferten Beispielen ein Experiment mit INTECRIO (Kapitel 11.1) sowie die Einrichtung einer ES1222-Karte (Kapitel 11.2), einer ES1303-Karte (Kapitel 11.3) und einer ES1325-Karte (Kapitel 11.4 und 11.5).

ASCET-RP V5.4 enthält die Beispieldateien `RTIOTutorial.exp` und `INTECRIO_Tutorial.exp`. Bei der Installation werden diese Dateien im Unterverzeichnis `ASCET5.1\export` Ihrer ASCET-Installation abgelegt.

Sie können die Beispieldateien wahlweise in eine neue oder eine existierende Datenbank importieren.

Hinweis

*Dieses Tutorial geht davon aus, dass der ETAS Network Manager **nicht** verwendet wird. Wenn Sie den ETAS Network Manager verwenden, kann ein abweichendes Verhalten die Folge sein; siehe Kapitel 2.2 und Kapitel 4.*

Neue Datenbank anlegen:

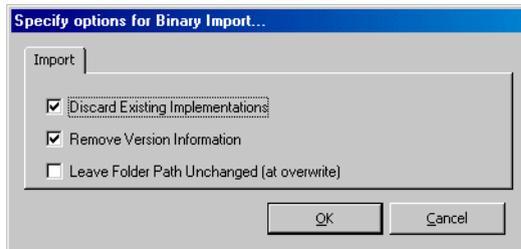
- Wählen Sie im Komponentenmanager die Menüfunktion **File** → **New Database**.
 - Geben Sie den Namen für die neue Datenbank ein.
 - Klicken Sie **OK**.
- Die Datenbank wird angelegt und geöffnet.

Importieren der Übungsbeispiele:

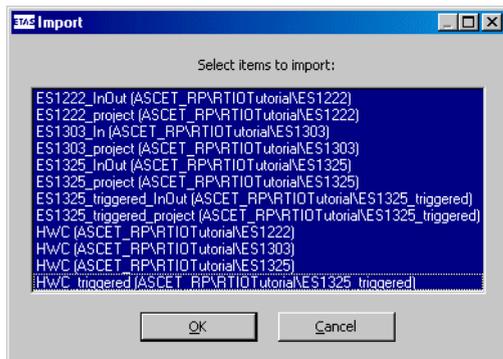
- Wählen Sie im Komponentenmanager **File** → **Import**.
- Wählen Sie im Dateiauswahlfenster das Unterverzeichnis `ASCET5.1\export`.
- Klicken Sie auf die Exportdatei `RTIOTutorial.exp`.

- Klicken Sie **Öffnen**.

Das Fenster „Specify options for Binary Import“ wird geöffnet.

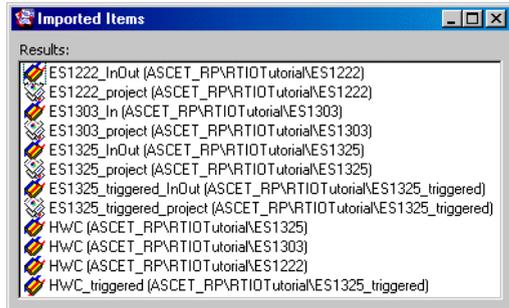


- Übernehmen Sie mit **OK**.
- Stellen Sie sicher, dass im Fenster „Import“ alle Komponenten für den Import angewählt sind.



- Klicken Sie **OK**.

Die Dateien werden importiert. Im Fenster „Imported Items“ wird eine Liste der importierten Datenbankobjekte angezeigt.



- Schließen Sie das Fenster „Imported Items“.
- Importieren Sie auf die gleiche Weise die Datei `INTECRIO_Tutorial.exp`.

Stellen Sie sicher, dass alle Komponenten für den Import gewählt sind.

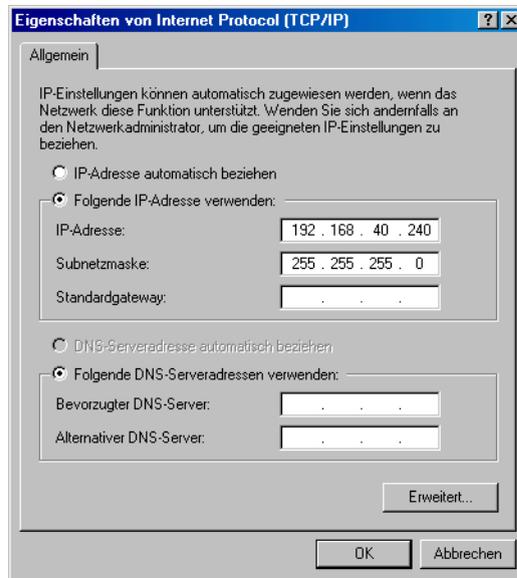


Konfiguration der TCP/IP-Protokolleinstellungen:

Für den Betrieb der ES1000.x muss das TCP/IP-Protokoll konfiguriert werden. Um Konflikte mit einer ggf. vorhandenen zweiten Netzwerkkarte für das lokale Netzwerk zu vermeiden, sind folgende TCP/IP-Einstellungen vorzunehmen:

- Deaktivieren Sie den DHCP-Dienst.
- Geben Sie als IP-Adresse `192.168.40.240` ein.

- Geben Sie als Subnetzmaske
255 . 255 . 255 . 0 ein.



(← Windows® 2000)

- Verwenden Sie für den DNS-Dienst die lokalen Einstellungen Ihres internen Netzwerks.
- Deaktivieren Sie den WINS-Dienst.
- Stellen Sie sicher, dass die Option „IP-Forwarding“ **nicht** aktiviert ist.

11.1 Tutorial – Experimentieren mit INTECRIO

Dieses Kapitel erklärt den Transfer von ASCET-Projekten nach INTECRIO sowie die Verwendung der Back-Animation (siehe Seite 64) beim Experimentieren mit INTECRIO. Die Erstellung eines Projekts in ASCET oder einer Arbeitsumgebung in INTECRIO ist nicht Teil dieses Kapitels; alle Dateien, die Sie benötigen, werden mitgeliefert.

- Die Exportdatei `INTECRIO_Tutorial.exp` enthält das ASCET-Projekt mit allen zugehörigen Komponenten.

Das ASCET-Projekt `P01_Project` besteht aus einem Datengenerator (Modul `M01_DataGenerator`), der als Zustandsautomat (`SM01_DataGenerator`) spezifiziert ist. Über den Parameter `PMODE`

bestimmen Sie, ob der Datengenerator als Sägezahn (1) oder als Dreieckssignal (2) läuft. Die generierten Daten stellen das Eingangssignal für einen Tiefpassfilter (Modul M01_LowPass) dar, der ebenfalls Teil des Projekts ist.

- Der Ordner `INTECRIO_Tutorial_WorkspaceES1130` enthält eine INTECRIO-Arbeitsumgebung, die für die Arbeit mit dem Simulationsrechner *ES1130* vorbereitet wurde.
- Der Ordner `INTECRIO_Tutorial_WorkspaceES1135` enthält eine INTECRIO-Arbeitsumgebung, die für die Arbeit mit dem Simulationsrechner *ES1135* vorbereitet wurde.

11.1.1 Vorbereitungen

Als erstes treffen Sie die nötigen Vorbereitungen.

INTECRIO-Arbeitsumgebung kopieren:

Die vorbereiteten INTECRIO-Arbeitsumgebungen finden Sie im Verzeichnis `ETAS\ASCET5.1\export` Ihrer ASCET-Installation.

- Wenn Sie mit der *ES1130* arbeiten, kopieren Sie das Verzeichnis `INTECRIO_Tutorial_WorkspaceES1130` auf Ihre Festplatte, z.B. nach `ETASData\INTECRIO1.0\INTECRIO_Tutorial_WorkspaceES1130`.
- Wenn Sie mit der *ES1135* arbeiten, kopieren Sie das Verzeichnis `INTECRIO_Tutorial_WorkspaceES1135` auf Ihre Festplatte, z.B. nach `ETASData\INTECRIO1.0\INTECRIO_Tutorial_WorkspaceES1135`.

Das Projekt `P01_Project` ist für die Arbeit mit einer *ES1130* eingerichtet. Wenn Sie mit diesem Simulationsrechner arbeiten, sind die Vorbereitungen hiermit abgeschlossen. Wenn Sie mit einer *ES1135* arbeiten, müssen Sie noch das entsprechende Target auswählen.

Target ES1135 auswählen:

- Öffnen Sie das Projekt `P01_Project` im Projekteditor.

- Wählen Sie in den Codegenerierungsoptionen das Target **ES1135** und den Compiler **GNU-C (PowerPC)**.

Die vordefinierten Betriebssystemeinstellungen für dieses Target werden geladen.

11.1.2 Transfer des Projekts

Als nächstes führen Sie den Transfer des Projekts nach INTECRIO durch.

Projekt nach INTECRIO transferieren:



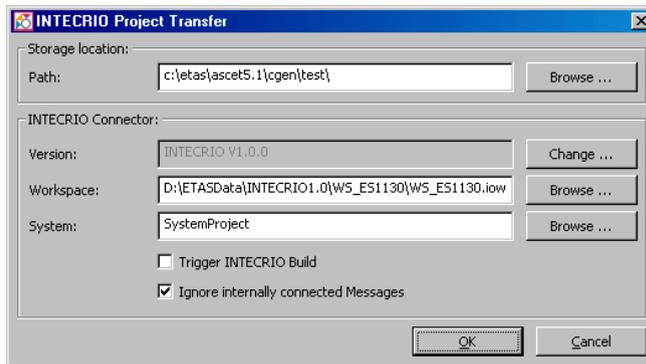
- Öffnen Sie das Projekt **P01_Project**.
- Wählen Sie aus dem Kombikästchen „Experiment Target“ den Eintrag **INTECRIO**.

Die Schaltflächen **Transfer Project to selected Experiment Target** und **Reconnect to Experiment of selected Experiment Target** sind nun verfügbar.



- Klicken Sie auf die Schaltfläche **Transfer Project to selected Experiment Target**.

Das Fenster „INTECRIO Project Transfer“ öffnet sich.



- Geben Sie im Feld „Path“ einen Pfad für die generierten Dateien an.
- Geben Sie über die Schaltfläche **Browse** neben dem Feld „Workspace“ diejenige mitgelieferte Arbeitsumgebung ein, mit der Sie arbeiten.

- Stellen Sie über die Schaltfläche **Browse** neben dem Feld „System“ das Systemprojekt `SystemProject` ein, das in der Arbeitsumgebung vorhanden ist.

Wenn INTECRIO noch nicht läuft, wird es jetzt gestartet.

- Klicken Sie **OK**, um den Transfer zu starten.
Der für die Arbeit mit INTECRIO erforderliche Code wird generiert und im angegebenen Verzeichnis abgelegt.
Das ASCET-Projekt wird in INTECRIO importiert und als Modul unter dem Namen `P01_Project` abgelegt. Es wird automatisch in das Systemprojekt `SystemProject` eingefügt.

11.1.3 Experimentieren in INTECRIO

Nun konfigurieren Sie in INTECRIO das Betriebssystem, starten den Build-Prozess und schließlich das INTECRIO-Experiment.

INTECRIO – Betriebssystem konfigurieren:

- Wechseln Sie in das INTECRIO-Fenster.
- Wählen Sie **System** → **OS Configuration**.
Der Betriebssystemeditor OSC öffnet sich. Da das Beispiel sehr einfach ist, können Sie die automatische Konfiguration verwenden.
- Wählen Sie **Edit** → **OS Auto mapping**.
Im Betriebsmodus `UserAppMode` wird die Task `auto_10msTask` angelegt. Die beiden Prozesse des ASCET-Projekts werden dieser Task zugewiesen.
Sie brauchen keine weiteren Einstellungen vorzunehmen.

INTECRIO – Build-Prozess starten:

- Wählen Sie im INTECRIO-Fenster **Integration** → **Build**.

Oder

- Wenn Sie den Build-Prozess zum wiederholten Mal starten, wählen Sie **Integration** → **Rebuild**.

Der Build-Prozess wird gestartet. Den Fortschritt können Sie im Feld „Log Window“ unten im INTECRIO-Fenster ablesen.

Nach erfolgreichem Build-Prozess erscheint in den letzten Zeilen folgende Meldung:

```
Action succeeded
```

```
The active system project has  
been set into the „Build“ mode.
```

INTECRIO – Experiment starten:

1. Experimentierumgebung öffnen

- Wählen Sie im INTECRIO-Fenster **Experiment** → **Open Experiment**.

Die Experimentierumgebung von INTECRIO öffnet sich in einem eigenen Fenster. Das Experiment wird in die Experimentierumgebung geladen.

2. Experiment starten

- Wählen Sie in der INTECRIO-Experimentierumgebung **Experiment** → **Download**.

Die ausführbare Datei (der Prototyp) wird auf die ES1000 geladen.

- Wählen Sie **Experiment** → **Start Experiment**.

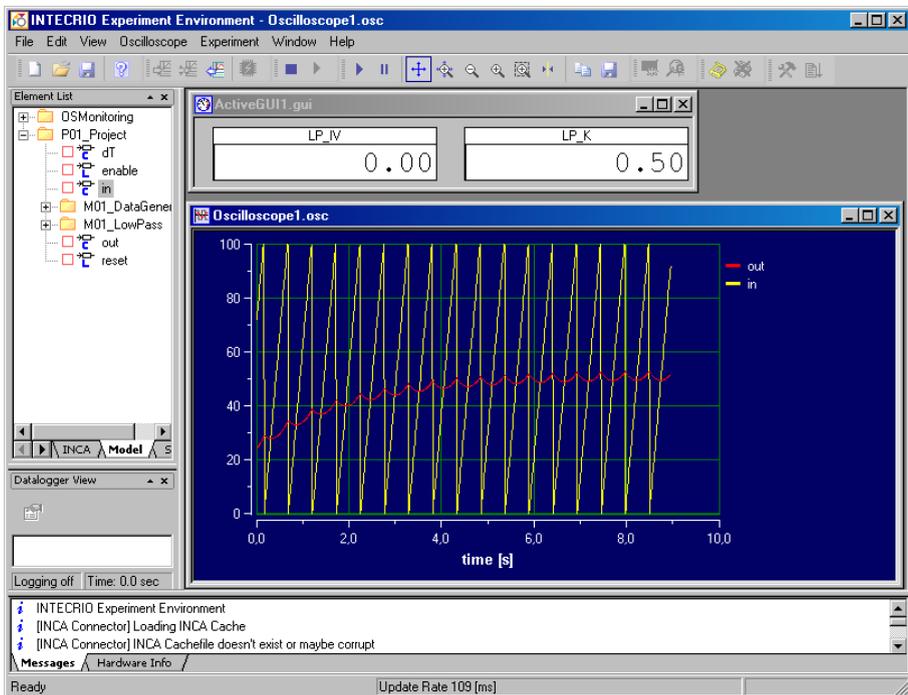
Die Simulation wird gestartet.

Um die Back-Animation zu verwenden, müssen Sie in INTECRIO keine Mess- und Verstellfenster öffnen. Da aber die Back-Animation mit ASCET kein Oszilloskop anbietet, enthält die INTECRIO-Arbeitsumgebung ein vordefiniertes Oszilloskop (`Oscilloscope1.osc`). Die Arbeitsumgebung enthält ebenfalls ein Active GUI-Fenster (`ActiveGUI1.gui`) mit zwei Verstellinstrumenten. Sie finden die beiden Dateien im Unterverzeichnis `EE_Dummy\System\GUI` Ihrer INTECRIO-Arbeitsumgebung.

Mess- und Verstelinstrumente laden:

- Wählen Sie in der INTECRIO-Experimentierumgebung **File** → **Open**.
Ein Dateiauswahlfenster öffnet sich.
- Stellen Sie die Anzeige *aller* Dateien ein.
- Wählen Sie das Unterverzeichnis `EE_Dummy\System\GUI` Ihrer INTECRIO-Arbeitsumgebung.
- Laden Sie nacheinander die Dateien `Oscilloscope1.osc` und `ActiveGUI1.gui`.
- Bringen Sie das Active GUI-Fenster mit **Edit** → **Run** in den Run-Modus.

Da Sie die Simulation bereits gestartet haben, werden sofort Werte angezeigt. Ihre INTECRIO-Experimentierumgebung sieht etwa so aus:



11.1.4 Back-Animation verwenden

Die Back-Animation starten Sie von ASCET aus. Das Experiment muss dafür in INTECRIO weiterlaufen.

Die Back-Animation starten:



Reconnect to Experiment of selected Experiment Target

- Klicken Sie im Projekteditor von ASCET auf die Schaltfläche **Reconnect to Experiment of Selected Experiment Target**.

Die Verbindung zu dem laufenden INTECRIO-Experiment wird hergestellt. Das Fenster „Physical Experiment ...“ öffnet sich.

- Wählen Sie im Fenster „Environment Browser“ die Umgebung **INTECRIO**.

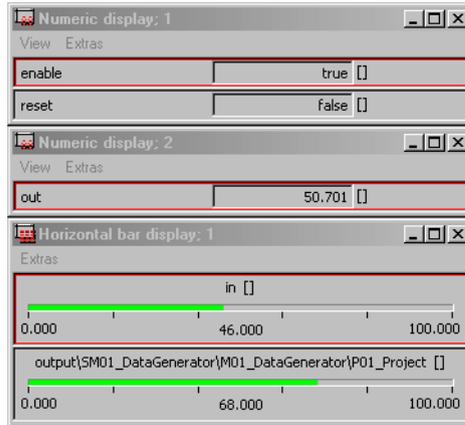
Die vordefinierte Anordnung aus Mess- und Verstellfenstern öffnet sich.

The screenshot shows the ASCET software interface for a physical experiment. The main window, titled "Physical Experiment for: P01_Project Target: >ES1130< Environment: >INTECRIO< 'INTECRIO Backanima...'", displays a block diagram with two main components: "M01_DataGenerator" and "M01_LowPass". The "M01_DataGenerator" block has inputs for "enable", "in", and "reset", and an output for "out". The "M01_LowPass" block has inputs for "enable", "in", and "reset", and an output for "out". The diagram shows connections between the "enable" and "in" ports of both blocks. Below the main window, several control windows are visible: "Numeric display: 1" for "enable" and "reset", "Numeric display: 2" for "out", "Horizontal bar display: 1" for "in" and "output", "Numeric Editor: 1" for "PMode|M01_DataGe" (value 1.000), "Logical Editor: 2" for "venable|M01_DataGenerator|P01_Project" (checked), and "Numeric Editor: 3" for "LP_IV|M01_LowPass" (value 0.000) and "LP_K|M01_LowPass" (value 0.500).



- Klicken Sie auf die Schaltfläche **Start Measurement**.

Die Messung im ASCET-Experiment wird gestartet; in den Messfenstern werden Werte angezeigt.



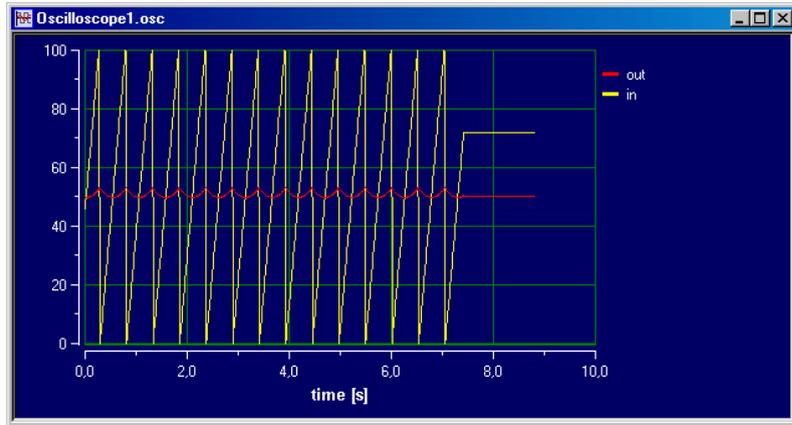
Sie können jetzt wahlweise im ASCET-Experiment oder im INTECRIO-Experiment Werte verstellen. Die geänderten Werte werden an das INTECRIO-Experiment weitergegeben und dort angezeigt und verwendet.

Werte verstellen:

- Geben Sie im ASCET-Experiment im Fenster „Numeric Editor; 3“ für die Größe `LP_IV` einen Wert ein.
Der Wert im linken Verstellinstrument des Fensters „ActiveGUI1.gui“ im INTECRIO-Experiment wird aktualisiert.
- Geben Sie im INTECRIO-Experiment im Fenster „ActiveGUI1.gui“ für die Größe `LP_IV` einen anderen Wert ein.
Der Wert im Fenster „Numeric Editor; 3“ des ASCET-Experiments wird aktualisiert.

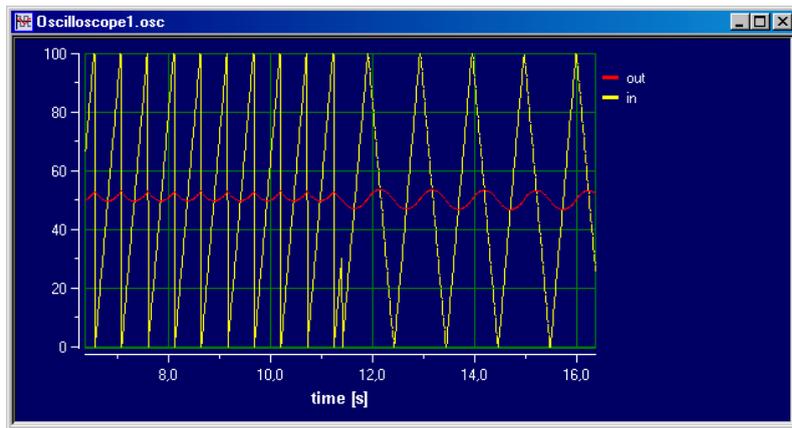
- Setzen Sie im Fenster „Logical Editor; 2“ den Parameter `venable` auf `false`.

Der Wert des Signalgenerators bleibt auf dem letzten Wert, der Tiefpassfilter wird auf den Initialisierungswert `LP_IV` gesetzt.



- Setzen Sie `venable` wieder auf `true` und dann `PMode` auf 2, um den anderen Signalgenerator auszuwählen.

Die Anzeige im Oszilloskop von INTECRIO ändert sich entsprechend.



Komponenten des ASCET-Projekts betrachten:

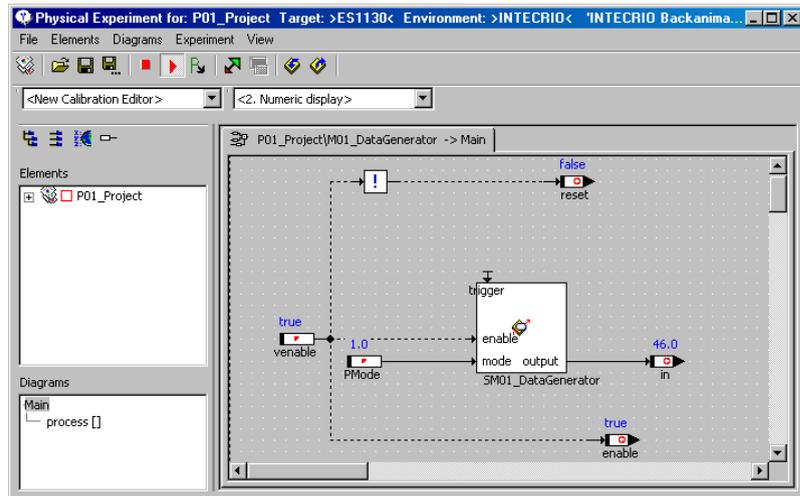
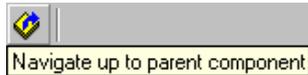
- Doppelklicken Sie im ASCET-Fenster „Physical Experiment ...“ im Register „Graphics“ auf eine Komponente, um sich diese im Detail anzusehen.

Die Komponente wird im Fenster „Physical Experiment ...“ angezeigt.

So können Sie durch die gesamte Hierarchie navigieren; die Schaltfläche **Navigate up to parent component** oder ein Doppelklick auf die leere Fläche bringt Sie zurück auf die nächsthöhere Ebene.

- Wählen Sie **View** → **Monitor All**.

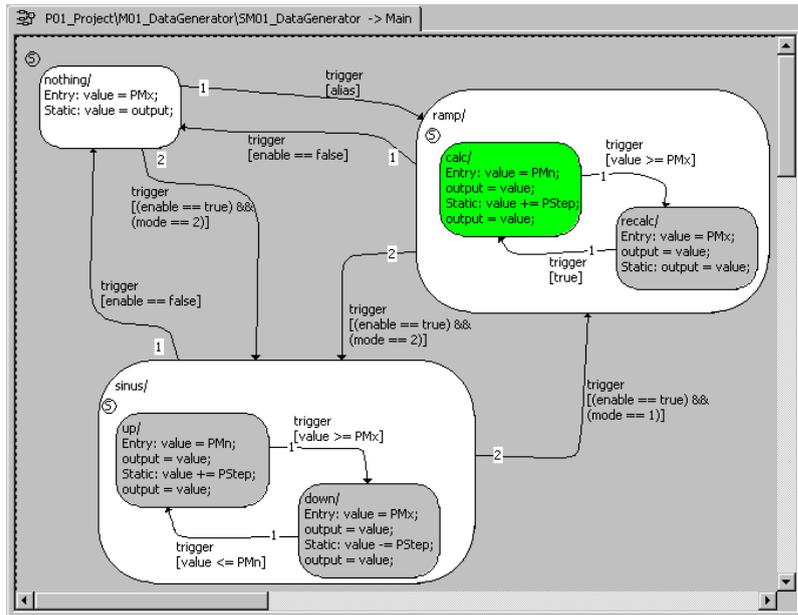
Über den Elementen werden deren aktuelle Werte angezeigt.



- Navigieren Sie durch die Modell-Spezifikation bis zum Zustandsautomaten **SM01_DataGenerator**.

- Klicken Sie mit der rechten Maustaste auf einen der Zustände und wählen Sie aus dem Kontextmenü **Animate States**.

Der aktuelle Zustand wird im Zustandsdiagramm farblich hervorgehoben.



11.2 Tutorial – ES1222 (CAN-IO)

Das Beispiel zur ES1222 enthält drei Variablen, die vom CAN-Kanal 1 gesendet und vom CAN-Kanal 2 empfangen werden. Wenn Sie die beiden Kanäle **zusammen mit einem Abschlusswiderstand** verbinden (siehe Abb. 11-3 auf Seite 308), können Sie ein Experiment durchführen.

Das Modell ist vorhanden und die für die RTIO-Einbindung der ES1222 notwendigen Vorbereitungen (Tasks, Messages und HWC-Modul) sind bereits getroffen; Sie legen die Hardware-Konfiguration an und erzeugen den Code.

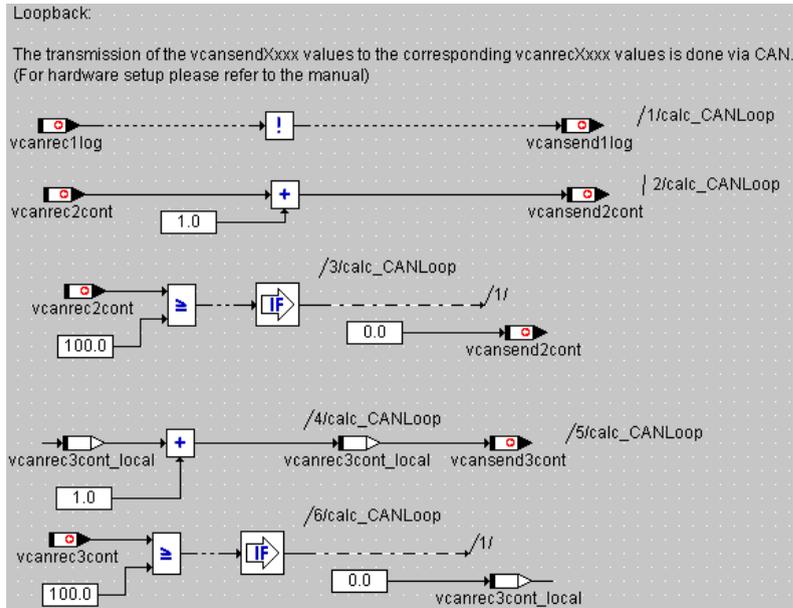


Abb. 11-1 ES1222 - Modul ES1222_InOut

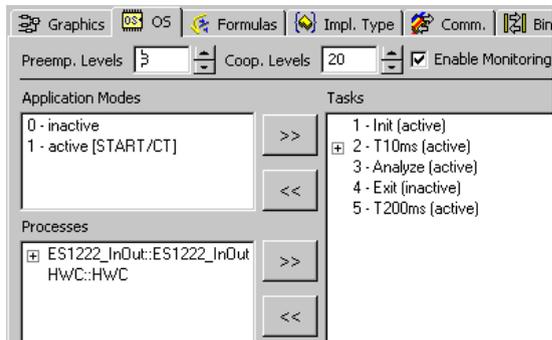
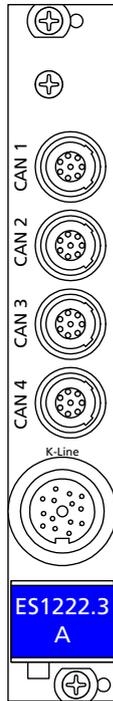


Abb. 11-2 ES1222 – Voreinstellungen im OS-Editor

11.2.1 Die ES1222 Karte

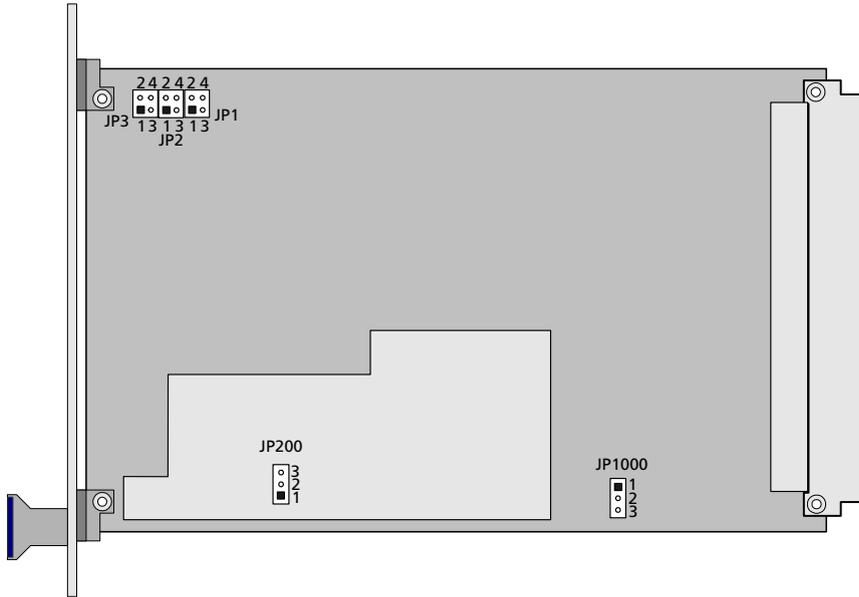
Die folgende Abbildung zeigt die Frontplatte der CAN-Karte ES1222:



Die Eingänge CAN1 bis CAN4 sind unabhängige galvanisch getrennte CAN-Schnittstellen; der Eingang K-Line ist eine serielle K- und L-Leitungsschnittstelle für die Verbindung von VMEbus-Systemen mit externen Geräten.

Steckbrücken der ES1222

Damit das Experiment funktioniert, achten Sie bitte auf die richtige Konfiguration der Steckbrücken.



Steckbrücke	PIN	Bedeutung
JP1	offen	CAN1 und CAN2 sind unabhängig
JP2	offen	
JP3	offen	
JP1000	1-2 geschlossen	
JP200 (nur ES1222.3) beibehalten		K-Line wird im Beispiel nicht verwendet

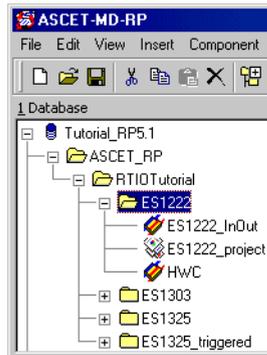
Betrieb mehrerer ES1222 Karten

In einer ES1000 können bis zu 4 ES1222 Karten betrieben werden.

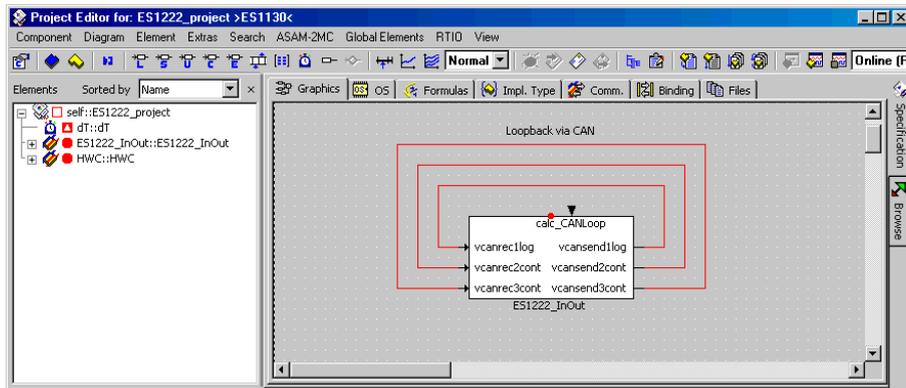
11.2.2 Beispielprojekt

Öffnen des Übungsbeispiels:

- Wählen Sie im Komponentenmanager den Ordner `ASCET_RP\RTIOTutorial\ES1222`.

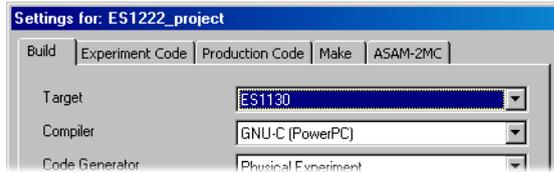


- Wählen Sie das Projekt `ES1222_project`.
- Öffnen Sie das Projekt.



- Klicken Sie auf die Schaltfläche **Specify Code Generation Options**.

Das Fenster „Settings for: ES1222_project“ öffnet sich.



- Wählen Sie im Register „Build“ die Optionen
 Target: >ES1130< oder >ES1135<,
 Compiler: GNU-C (PowerPC).

Hinweis

Nur Messages, die als „Exported“ deklariert sind, stehen für die RTIO-Kommunikation zur Verfügung.

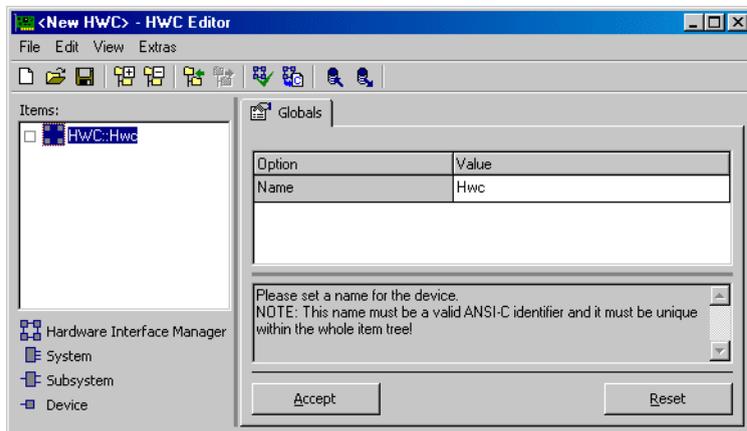
11.2.3 Erstellen der Hardware-Konfiguration

Hinweis

Normalerweise müssen Sie das C-Code-Modul HWC anlegen und in das Projekt einbinden, bevor Sie die Hardware-Konfiguration bearbeiten. Im Tutorial ist Ihnen dieser Schritt jedoch abgenommen.

Öffnen des HWC-Editors

- Wählen Sie im Projekteditor RTIO → **Open Editor**.
 Der HWC-Editor wird geöffnet.



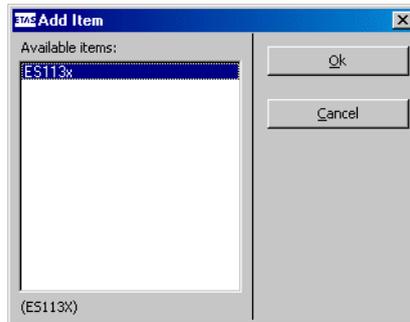
Hardware-Konfiguration (HWC) erstellen:

Die Hardware muss in der Items-Liste als baumartige Struktur beschrieben werden. Das Item **HWC** ist immer vorhanden und bildet die Wurzel des Baumes.

- Wählen Sie im HWC-Editor **Edit** → **Add Item** oder



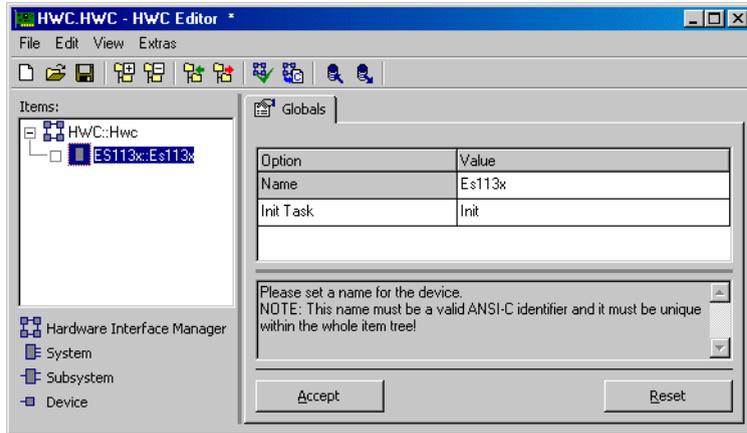
- klicken Sie auf die Schaltfläche **Add Item**.
Das Fenster „Add Item“ wird angezeigt.



Über **Add Item** wird immer die Liste der verfügbaren Items der nächsten Hierarchiestufe geöffnet.

- Wählen Sie den Eintrag **ES113x** aus.
Dieser Eintrag dient der Beschreibung des ES1000.x-Systems mit integriertem ES1130 oder ES1135 PowerPC-Rechenknoten.

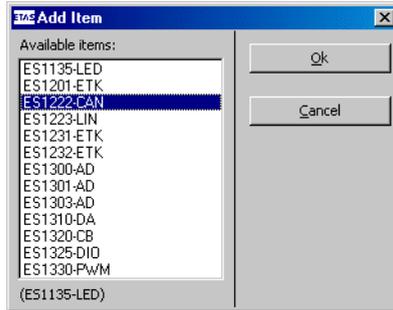
- Klicken Sie **OK**.
In der Liste „Items“ wird das Element `ES113x` hinzugefügt.



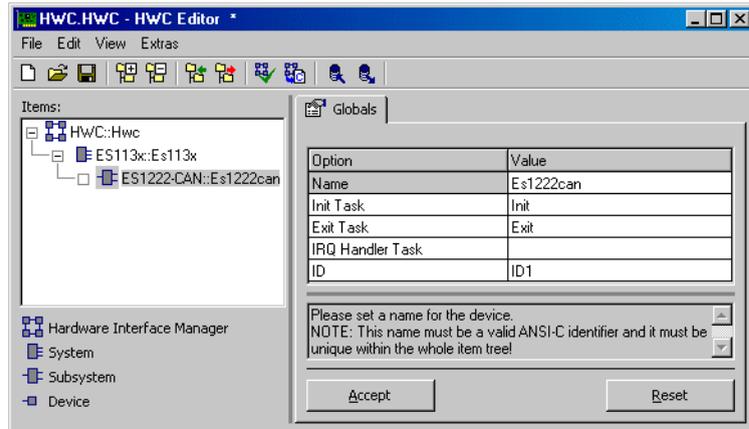
- Wählen Sie in der Liste „Items“ das Element `ES113x`.
Im Register „Globals“ wird für die Option `Init Task` der Task-Name `Init` vorgegeben. Im OS-Editor des Beispielprojekts gibt es eine gleichnamige `Init-Task`, infolgedessen muss hier keine andere `Init-Task` ausgewählt werden.

ES1222 einbinden und einrichten:

- Öffnen Sie über **Edit** → **Add Item** die Liste der verfügbaren Elemente der nächsten Hierarchiestufe.



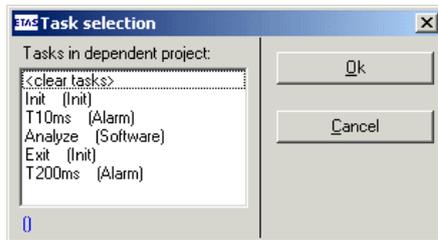
- Wählen Sie den Eintrag **ES1222-CAN** aus. Dieser Eintrag dient der Beschreibung der ES1222 CAN und K-Line-Schnittstellen.
- Klicken Sie **OK**.
In der Liste „Items“ wird das Element **ES1222-CAN** hinzugefügt.



- Wählen Sie in der Liste „Items“ das Element **ES1222-CAN**.

Im Register „Globals“ werden für die Optionen `Init Task` und `Exit Task` die Task-Namen `Init` und `Exit` vorgegeben. Im OS-Editor des Beispielprojekts gibt es gleichnamige Init-Tasks, infolgedessen müssen hier keine anderen Tasks ausgewählt werden. Für die Option `IRQ Handler Task` hingegen gibt es keinen vordefinierten Namen.

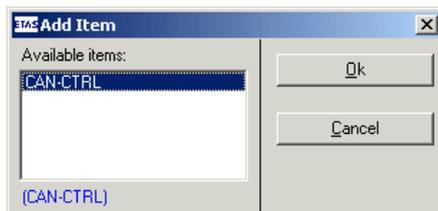
- Doppelklicken Sie in das leere Feld neben der Option `IRQ Handler Task`.
Das Fenster „Task selection“ öffnet sich.



- Wählen Sie die Software-Task `Analyze` und klicken Sie **OK**.
Die Nummer der Karte wird in der Option `ID` automatisch eingestellt. Da das Element `ES1222-CAN` das erste dieses Typs ist, ist der Wert „ID1“ voreingestellt.
- Übernehmen Sie die Änderung mit **Accept**.

CAN-Kanäle einbinden:

- Wählen Sie im HWC-Editor das Item `ES1222-CAN`.
- Öffnen Sie über **Edit** → **Add Item** die Liste der verfügbaren Elemente der nächsten Hierarchiestufe.



Baud Rate [kBaud]	500
Identifizier	1000 500 250 125 100 50 20 10 <Special Timing>

- Wählen Sie den Eintrag **CAN-CTRL** aus und klicken Sie **OK**.

Dieser Eintrag dient der Beschreibung des CAN-Controllers.

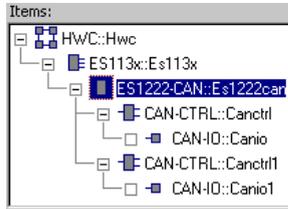
- Klicken Sie im Register „Globals“ neben der Option „Baud Rate [kBaud]“ auf den Wert 1000.

Das Feld wird zu einem Kombikästchen, in dem die verfügbaren Baud-Raten aufgelistet sind.

- Wählen Sie aus dem Kombikästchen eine Rate von 500 kBaud.
- Sichern Sie die Einstellung mit **Accept**.
- Wählen Sie in der Liste „Items“ das Element **CAN-CTRL**.
- Öffnen Sie über **Edit** → **Add Item** die Liste der verfügbaren Elemente der nächsten Hierarchiestufe.

- Wählen Sie den Eintrag **CAN-IO** aus und klicken Sie **OK**.
- Fügen Sie einen zweiten CAN-Controller mit **CAN-IO Device** für den zweiten CAN-Kanal hinzu.

Nun ist der Item-Baum für die Beschreibung des Beispielsystems komplett spezifiziert.



11.2.4 HWC-Einstellungen für die ES1222 (CAN-IO) vornehmen

Nun müssen die beiden CAN-Kanäle konfiguriert werden. Der erste Kanal (Element `CAN-IO::Canio` im HWC-Editor) soll die Signale senden, die der zweite Kanal (`CAN-IO::Canio1`) empfängt.

Im Register „Globals“ des Items `CAN-IO` brauchen Sie für das Beispiel keine Einstellungen vorzunehmen.

Kanal 1

Register „Groups“: In diesem Register werden die CAN-Botschaften in Form von Signalgruppen spezifiziert.

Einstellungen für Gruppe 1 vornehmen:

- Wählen Sie in der Liste „Items“ das Element `CAN-IO::Canio`.
- Wählen Sie das Register „Groups“.

No.	Group	Direction	Task	IRQ	Identifier dec	Identifier hex	Length [Byte]
1	Group1	send		---	0	0	8

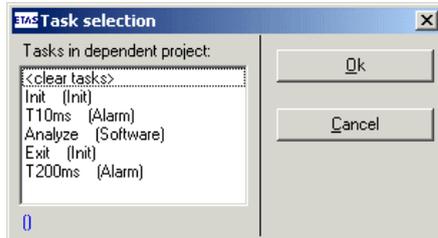
Die Gruppe `Group1` ist standardmäßig vorhanden.

- Doppelklicken Sie in die Zeile 1, Spalte „Group“.

Das Feld der Tabelle wird zu einem Eingabefeld.



- Nennen Sie die Gruppe MSG_100.
- Doppelklicken Sie in die Zeile 1, Spalte „Task“.
Das Fenster „Task Selection“ öffnet sich.



- Wählen Sie die Alarm-Task T10ms aus und klicken Sie auf **OK**.
- Klicken Sie in die Zeile 1, Spalte „Identifier dec“.
- Geben Sie für den Identifier den Wert 100 ein.
In der Spalte „Identifier hex“ wird automatisch der Wert 64 eingetragen.
- Sichern Sie die Einstellungen mit **Accept**.

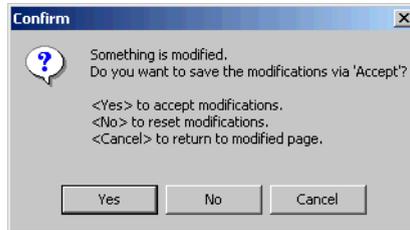


Gruppe hinzufügen:

Als nächstes fügen Sie eine zweite Gruppe hinzu, die in einer anderen Task übertragen wird.

- Klicken Sie in Zeile 1, Spalte „No.“, um die vorhandene Gruppe auszuwählen.
Im Menü **Edit** werden Funktionen zum Hinzufügen von Signalgruppen aktiviert.
- Wählen Sie **Edit** → **Add Row After**
oder
- wählen Sie im Register „Groups“ **Add Row After** aus dem Kontextmenü.

- Wenn Sie die Änderungen an der ersten Gruppe nicht gespeichert haben, werden Sie aufgefordert, das jetzt zu tun.



- Bestätigen Sie den Hinweis mit **OK**.
- Rufen Sie erneut **Add Row After** auf.

Eine neue Zeile wird nach der ersten eingefügt.

No.	Group	Direction	Task	IRQ	Identifier dec	Identifier hex	Length [Byte]
1	MSG_100	send	T10ms	---	100	64	8
2	Group2	send		---	0	0	8

Einstellungen für Gruppe 2 vornehmen:

- Nennen Sie die Gruppe MSG_101.
- Wählen Sie in der Spalte „Task“ die Alarm-Task T200ms aus.
- Tragen Sie in der Spalte „Identifier dec“ den Wert 101 ein.
In der Spalte „Identifier hex“ wird automatisch der Wert 65 eingetragen.
- Sichern Sie die Einstellungen mit **Accept**.

Register „Signals“: In diesem Register werden die CAN-Signale spezifiziert. Ein Signal ist standardmäßig vorgegeben. Das Beispiel arbeitet mit zwei numerischen und einem logischen Signal; Sie fügen als erstes zwei Signale hinzu.

- Doppelklicken Sie in Zeile 2, Spalte „1 0“.
Das Feld wird mit einer 1 markiert.
- Doppelklicken Sie in Zeile 2, Spalte „1 7“.
Das Feld wird mit einer 1 markiert.
- Drücken Sie die <SHIFT>-Taste und klicken Sie noch einmal in Zeile 2, Spalte „1 7“.
Alle Felder in den dazwischenliegenden Spalten werden mit einer 1 markiert.
Damit ist festgelegt, dass das Signal im Byte 1, Bits 0 – 7, der ersten CAN-Botschaft übertragen wird.

Zweites numerisches Signal einrichten:

- Nennen Sie das Signal `vcansend3IRQ`.
- Wählen Sie in Zeile 3, Spalte „Group“, die Gruppe `MSG_101`.
- Wählen Sie in Zeile 3, Spalte „Signal-Type“, den Typ `uint`.
- Markieren Sie in Zeile 3 wie oben beschrieben die Spalten „0 0“ bis „0 7“.

Damit ist festgelegt, dass das Signal im Byte 0, Bits 0 - 7, der zweiten CAN-Botschaft übertragen wird.

- Übernehmen Sie Ihre Eingaben mit **Accept**.

Wenn Sie alle drei Signale eingerichtet haben, sollte die Tabelle so aussehen:

No.	Group	Signal	Formula	Signal-Type	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0		
1	MSG_100	vcansend1log	!(phys)=phys	bool																				
2	MSG_100	vcansend2cont	!(phys)=phys	uint																				
3	MSG_101	vcansend3IRQ	!(phys)=phys	int																				

Register „Mappings“: In diesem Register werden die CAN-Signale und die ASCET Messages aus dem Projekt einander zugeordnet. Durch Anklicken der gewünschten Zelle in der Spalte „ASCET Message“ kann ein Auswahldialog geöffnet werden. Dieser enthält nur Messages, die das Attribut *Exported* besitzen und der Übertragungsrichtung der Signalgruppe (*send* oder *receive*) entsprechen; also:

- Direction = receive → Receive-Messages,

- Direction = send → Send-Messages

ASCET Message manuell zuordnen:

- Wählen Sie das Register „Mappings“.

No.	Device	Group	Direction	Task	Signal	ASCET Message Data	
1	Cario	MSG 100	send	T10ms	vcansend1loa		0.000
2	Cario	MSG 100	send	T10ms	vcansend2cont		0.000
3	Cario	MSG 101	send	T200ms	vcansend3IRQ		0.000

- Doppelklicken Sie in die Zeile 3, Spalte „ASCET Message“.
Das Fenster „Message Selection“ öffnet sich.



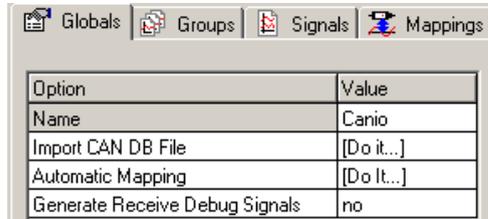
Es enthält alle Send-Messages aus dem ASCET Projekt.

- Wählen Sie die Message `vcansend3cont` aus und klicken Sie auf **OK**.
- Sichern Sie Ihre Eingabe mit **Accept**.

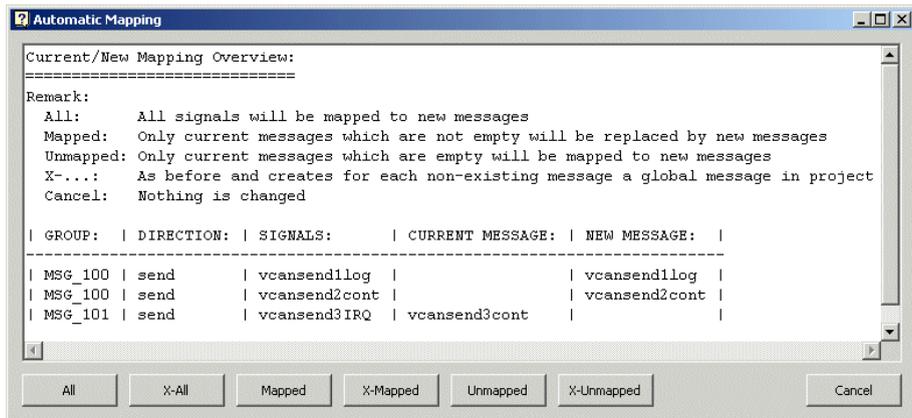
Die Signale in den Zeilen 1 und 2 haben die gleichen Namen wie zwei Send-Messages im Projekt. Sie können die Messages auf die eben beschriebene Weise manuell zuordnen, Sie können sie aber auch automatisch zuordnen.

ASCET Message automatisch zuordnen:

- Wählen Sie das Register „Globals“.



- Doppelklicken Sie in die Zeile „Automatic Mapping“, Spalte „Value“.
Das Fenster „Automatic Mapping“ öffnet sich.



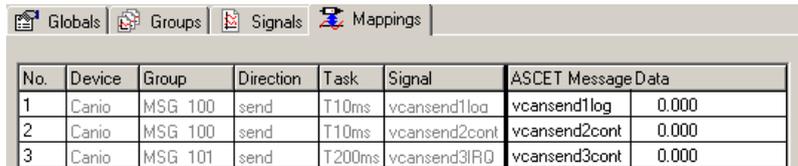
Es enthält eine Kurzbeschreibung der Schaltflächen sowie eine Auflistung der gegenwärtigen Zuordnungen.

Eine ausführliche Beschreibung finden Sie im Abschnitt „Automatic Mapping“ auf Seite 172.

- Klicken Sie auf die Schaltfläche **Unmapped**.
Den Signalen `vcansend1log` und `vcansend2cont` werden die gleichnamigen ASCET Messages zugeordnet.

Dem Signal `vcansend3IRQ` wurde bereits manuell eine Message zugeordnet, es bleibt daher unbeeinflusst.

Im Register „Mappings“ ist nun zu sehen, dass jedem Signal eine Message zugeordnet ist.



No.	Device	Group	Direction	Task	Signal	ASCET Message Data	
1	Canio	MSG_100	send	T10ms	vcansend1log	vcansend1log	0.000
2	Canio	MSG_100	send	T10ms	vcansend2cont	vcansend2cont	0.000
3	Canio	MSG_101	send	T200ms	vcansend3IRQ	vcansend3cont	0.000

Damit ist der erste CAN-Kanal komplett konfiguriert.

Kanal 2

Register „Groups“: In diesem Register werden die CAN-Botschaften in Form von Signalgruppen spezifiziert.

Einstellungen für Gruppe 1 vornehmen:

- Wählen Sie in der Liste „Items“ das Element `CAN-IO::Canio1`.
- Wählen Sie das Register „Groups“.
Der standardmäßig erzeugte Inhalt des Registers ist derselbe wie beim ersten Kanal.
- Nennen Sie die erste Gruppe `MSG_100` (s. Seite 293).

Da der zweite Kanal die Signale *empfängt*, müssen Sie die Übertragungsrichtung der Gruppe ändern.



- Doppelklicken Sie in Zeile 1, Spalte „Direction“.
Das Feld der Tabelle wird zu einem Kombikästchen, in dem die verfügbaren Richtungen aufgelistet sind.
- Wählen Sie die Richtung `receive`.
- Wählen Sie in Zeile 1, Spalte „Task“, die Alarm-Task `T10ms` aus (s. Seite 293).

- Geben Sie in die Zeile 1, Spalte „Identifizier dec“ den Wert 100 ein.
In der Spalte „Identifizier hex“ wird automatisch der Wert 64 eingetragen.
- Sichern Sie die Einstellungen mit **Accept**.

Als nächstes fügen Sie wie auf Seite 294 beschrieben eine zweite Gruppe hinzu, die im Interrupt-Betrieb übertragen wird.

Einstellungen für Gruppe 2 vornehmen:

- Nennen Sie die zweite Gruppe MSG_101.
- Wählen Sie in der Spalte „Direction“ die Richtung receive.
- Doppelklicken Sie in Zeile 2, Spalte „IRQ“.



- Das Feld der Tabelle wird zu einem Kombikästchen, in dem die verfügbaren Optionen aufgelistet sind.
- Wählen Sie die Option yes.

Die Spalte „Task“ wird automatisch zurückgesetzt und gesperrt. In die Spalte „Prescaler“ wird automatisch der Wert 1 eingetragen.

No.	Device	Group	Direction	Task	IRQ	Identifizier dec	Identifizier hex	Length [Byte]	Activated Task	Pre-scaler
1	Canio1	MSG_100	receive	T10ms	no	100	64	8		---
2	Canio1	MSG_101	receive	---	yes	0	0	8		1

- Tragen Sie in der Spalte „Identifizier dec“ den Wert 101 ein.
In der Spalte „Identifizier hex“ wird automatisch der Wert 65 eingetragen.
- Sichern Sie die Einstellungen mit **Accept**.

Register „Signals“: In diesem Register werden die CAN-Signale spezifiziert. Ein Signal ist standardmäßig vorgegeben. Sie fügen als erstes zwei Signale hinzu, wie auf Seite 296 beschrieben.

Das logische und ein numerisches Signal werden in der ersten Gruppe übertragen, das andere numerische Signal in der zweiten Gruppe.

Logisches Signal einrichten:

- Nennen Sie das erste Signal vcanrec1log.

- Wählen Sie in Zeile 1, Spalte „Signal-Typ“, den Typ `bool`.
- Markieren Sie die Spalte „0 0“ in Zeile 1 (s. Seite 297).
Damit ist festgelegt, dass das Signal im Byte 0, Bit 0, der ersten CAN-Botschaft übertragen wird.

Erstes numerisches Signal einrichten:

- Nennen Sie das Signal `vcanrec2cont`.
- Wählen Sie in Zeile 2, Spalte „Signal-Typ“, den Typ `uint`.
- Markieren Sie die Spalten „1 0“ bis „1 7“ in Zeile 2 (s. Seite 297).
Damit ist festgelegt, dass das Signal im Byte 1, Bits 0 - 7, der ersten CAN-Botschaft übertragen wird.

Zweites numerisches Signal einrichten:

- Nennen Sie das Signal `vcanrec3cont`.
- Wählen Sie in Zeile 3, Spalte „Group“, die Gruppe `MSG_101`.
- Wählen Sie in Zeile 3, Spalte „Signal-Typ“ den Typ `uint`.
- Markieren Sie die Spalten „0 0“ bis „0 7“ in Zeile 3.
Damit ist festgelegt, dass das Signal im Byte 0, Bits 0 - 7, der zweiten CAN-Botschaft übertragen wird.
- Übernehmen Sie Ihre Änderungen mit **Accept**.

Wenn Sie alle drei Signale eingerichtet haben, sollte die Tabelle so aussehen:

No.	Group	Signal	Formula	Signal-Type	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
1	MSG_100	vcanrec1log	f[phys]=phys	bool																	1
2	MSG_100	vcanrec2cont	f[phys]=phys	uint																	
3	MSG_101	vcanrec3cont	f[phys]=phys	uint																	

Zu allen drei Signalen gibt es gleichnamige Receive-Messages im Projekt. Sie können daher alle Messages automatisch zuordnen.

ASCET Messages automatisch zuordnen:

- Wählen Sie das Register „Globals“.
- Klicken Sie in die Zeile „Automatic Mapping“, Spalte „Value“.

Das Fenster „Automatic Mapping“ öffnet sich.

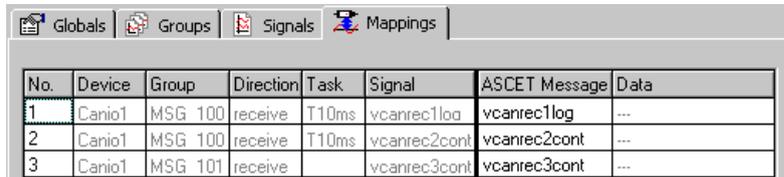
Eine ausführliche Beschreibung finden Sie in Abschnitt „Automatic Mapping“ auf Seite 172.

- Klicken Sie auf die Schaltfläche **All**.

Den Signalen werden die gleichnamigen ASCET Messages zugeordnet.

Sie können die Messages natürlich auch im Register „Mappings“ manuell zuordnen, wie auf Seite 299 beschrieben.

Im Register „Mappings“ ist nun jedem Signal die gleichnamige Message zugeordnet.



No.	Device	Group	Direction	Task	Signal	ASCET Message	Data
1	Canio1	MSG 100	receive	T10ms	vcanrec1log	vcanrec1log	---
2	Canio1	MSG 100	receive	T10ms	vcanrec2cont	vcanrec2cont	---
3	Canio1	MSG 101	receive		vcanrec3cont	vcanrec3cont	---

Damit ist auch der zweite CAN-Kanal komplett konfiguriert.

11.2.5 Hardware-Konfiguration speichern

Die erstellte Hardware-Konfiguration kann im File-Container des Projekts oder als DOS-Datei (Erweiterung *.HWX) gespeichert werden.

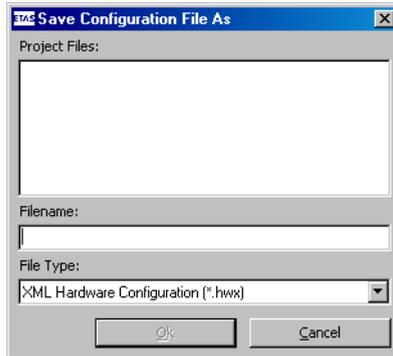
Speichern der Hardware-Konfiguration im File-Container:



- Klicken Sie im HWC-Editor auf die Schaltfläche **Save**

oder

- wählen Sie die Menüfunktion **File** → **Save**, um die Hardware-Konfiguration zu speichern. Das Fenster „Save Configuration File“ öffnet sich, wenn Sie die Konfiguration das erste Mal speichern.



- Wählen Sie im Kombikästchen „File Type“ den Eintrag XML Hardware Configuration (* .hwx).
- Geben Sie einen Namen für die Konfiguration ein.

Hinweis

Die Lösung des Beispiels ist unter dem Namen ES1222 .hwx abgelegt. Achten Sie darauf, diese **nicht zu überschreiben**.

- Klicken Sie **OK**. Die Hardware-Konfiguration wird im File-Container des Projekts gespeichert. Sie taucht dort im Register „Files“ auf.

Speichern der Hardware-Konfiguration als DOS-Datei:

- Wählen Sie im HWC-Editor **File** → **Export**.
- Geben Sie im Fenster „Export Hardware Configuration“ Dateityp, Pfad und Dateinamen für die Hardware-Konfiguration an.

Die Hardware-Konfiguration wird in die angegebene Datei gesichert. Sie kann später mit **File** → **Import** wieder in den HWC-Editor eingelesen werden.

11.2.6 Code für HWC-Modul erzeugen

Anschließend muss die Generierungssequenz für das HWC-Modul gestartet werden.

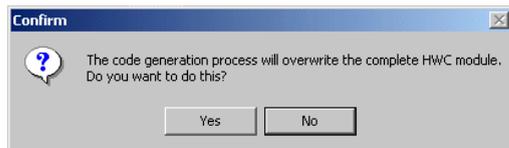
Erzeugen von Code für das HWC-Modul

- Wählen Sie im HWC-Editor **Extras** → **Generate Code** → **For Current Experiment**

oder



- klicken Sie auf die Schaltfläche **Generate Code for Current Experiment**.
- Bestätigen sie das Überschreiben des HWC-Moduls.



Der C-Code für das HWC-Modul wird für das derzeit in den Codegenerierungsoptionen des Projekts ausgewählte Experiment generiert.

Sie können auch Code für andere Experimente generieren:

- Wählen Sie im HWC-Editor **Extras** → **Generate Code** → **For Phys. and Quant. Experiment**, um Code für das Physikalische und Quantisierungs-Experiment zu generieren.

- Wählen Sie im HWC-Editor **Extras** → **Generate Code** → **For Phys., Quant. and Impl. Experiment**, um Code für das Physikalische, Quantisierungs- und Implementierungs-Experiment zu generieren.

11.2.7 Experimentieren mit dem Beispielprojekt

Alle RTIO-spezifischen Aktionen sind nun beendet, somit kann der HWC-Editor geschlossen werden. Als weiterer Arbeitsgang schließt sich nun die normale Codegenerierung für das Experimentaltarget an, die aus dem Projekteditor gestartet wird.

Hinweis

Es empfiehlt sich, das HWC-Modul nicht in die grafische Darstellung des Projekteditors mit aufzunehmen, da dieses Modul durch jeden RTIO-Generierungsprozess verändert wird. Man erhält eine ungünstige grafische Darstellung des HWC-Moduls.

Codegenerierung für das Experimentaltarget:

- Wählen Sie im Projekteditor **Component** → **Build**, um den Code für das gesamte Projekt zu erzeugen.
- Wählen Sie **Component** → **View Generated Code**, um den generierten Code zu betrachten.

Online experimentieren:

- Verbinden Sie die Eingänge CAN 1 und CAN 2 der ES1222 mit den passenden Kabeln und einem Abschlusswiderstand.

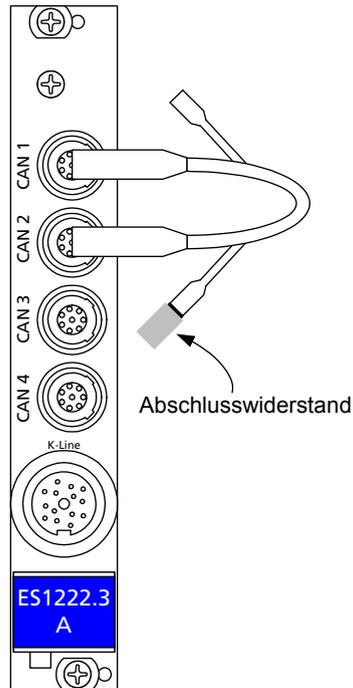


Abb. 11-3 ES1222 mit Kabel und Abschlusswiderstand

Hinweis

Wenn Sie den Abschlusswiderstand vergessen, findet keine Kommunikation statt.

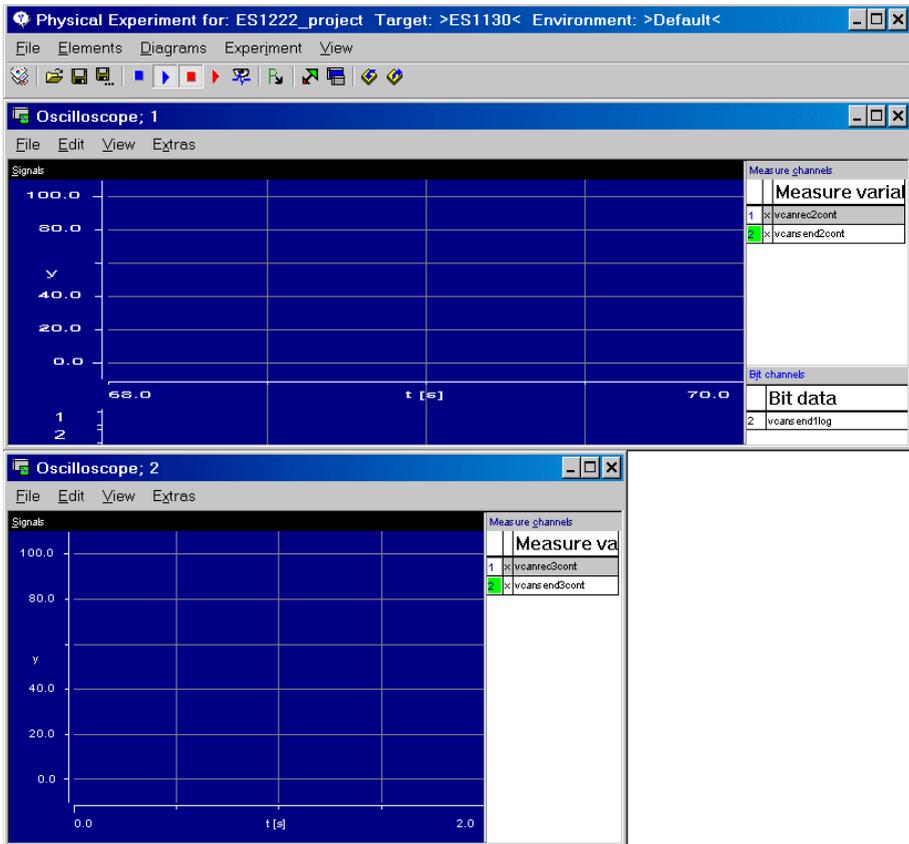


- Wählen Sie im Projekteditor aus dem Kombikästchen „Experiment Target“ den Eintrag `Online (RP)`.

Der Eintrag `Offline (RP)` ist für Offline-Experimente auf dem Target gedacht.

- Wählen Sie **Component** → **Open Experiment**.

Das Fenster „Physical Experiment“ sowie die vordefinierte Experimentierumgebung bestehend aus zwei Oszilloskopen öffnen sich.



- Wählen Sie im Fenster „Physical Experiment“ **Experiment** → **Start ERCOS**

oder



- klicken Sie auf die Schaltfläche **Start ERCOS**. Das Betriebssystem wird gestartet, das Modell wird ausgeführt.

- Wählen Sie im Fenster „Physical Experiment“ **Experiment → Start Measurement**

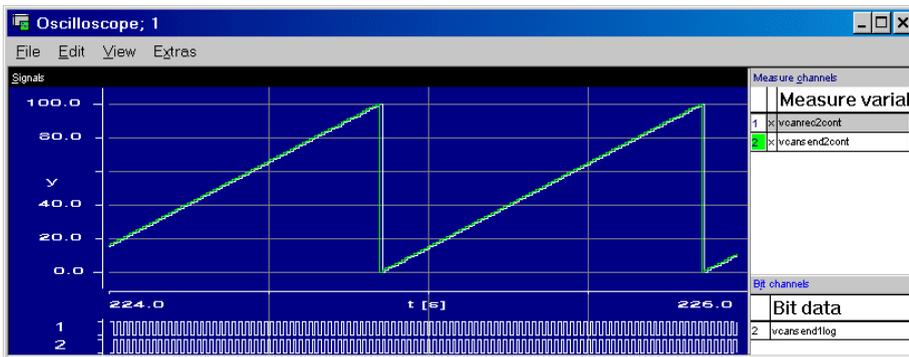
oder



- klicken Sie auf die Schaltfläche **Start Measurement**.

Die Werte der ASCET-Messages werden in den beiden Oszilloskopen angezeigt.

Das obere Oszilloskop zeigt in der Fläche „Signals“ die Send-Message `vcansend2cont` (grüne linke Kurve) und die Receive-Message `vcanrec2cont` (weiße rechte Kurve). In der Bitanzeige werden die Receive-Message `vcanrec1log` (obere Kurve) und die Send-Message `vcansend1log` (untere Kurve) angezeigt.



Verlauf und Versatz der Kurven kommen folgendermaßen zustande:

- Die Werte für die Send-Messages werden im 10 ms-Raster gerechnet (Prozess `calcCANLoop` in der Task `T10ms`) und gesendet (Prozess `CanioMSG100_T10ms_HWCL` in der Task `T10ms`).
- Ebenfalls im 10 ms-Raster, aber eine Zeiteinheit später, wird der gerechnete Wert über die Receive-Message wieder eingelesen (Prozess `Canio1MSG100_T10ms_HWCF` in der Task `T10ms`).

Das untere Oszilloskop zeigt die Send-Message `vcansend3cont` (grüne Kurve mit kleinen Stufen) und die Receive-Message `vcanrec3cont` (weiße Kurve mit großen Stufen).



Die Send-Message `vcansend3cont` wird im 10 ms-Raster gerechnet (Prozess `calcCANLoop` in der Task `T10ms`), deswegen verläuft die grüne Kurve ebenso glatt wie die Kurve von `vcansend2cont`. Der Wert wird im 200 ms-Raster gesendet (Prozess `CanioMSG101_T200ms_HWCL` in der Task `T200ms`) und im Interruptbetrieb, getriggert durch die Task `Analyse` über die Receive-Message wieder empfangen. Entsprechend stufig verläuft die Kurve für `vcanrec3cont`.

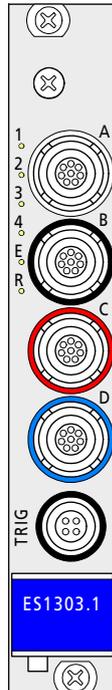
11.3 Tutorial – ES1303

In diesem Abschnitt wird eine ES1303 über das RTIO-Package in ein ASCET-RP Projekt eingebunden. Voraussetzung für eine Einbindung sind die Schritte, die in Kapitel 6 „Vorbereitende Maßnahmen“ beschrieben werden. Insbesondere soll noch einmal darauf hingewiesen werden, dass die Einordnung (Mapping) zwischen RTIO-Kanal und Modell nur über Messages erfolgen kann, die im übrigen Modell als *Exported* deklariert sind.

Da mit einer ES1303-Karte nur Werte empfangen, nicht aber gesendet werden können, wird in diesem Beispiel auf eine anschließende experimentelle Veranschaulichung verzichtet.

11.3.1 Die ES1303 Karte

Die folgende Abbildung zeigt die Frontplatte der Analog/Digital Wandlerkarte ES1303:



Die Eingänge A bis D fassen je vier Eingangskanäle zusammen; der Eingang TRIG ist für die Triggerkanäle gedacht.

Eingangsspannungsbereiche der ES1303

Der Eingangsspannungsbereich der ES1303 Karte kann im HWC-Editor entweder auf ± 10 V oder auf ± 60 V eingestellt werden.

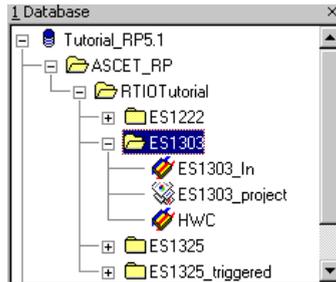
Betrieb mehrerer ES1303 Karten

In einer ES1000 können bis zu vier ES1303 Karten betrieben werden.

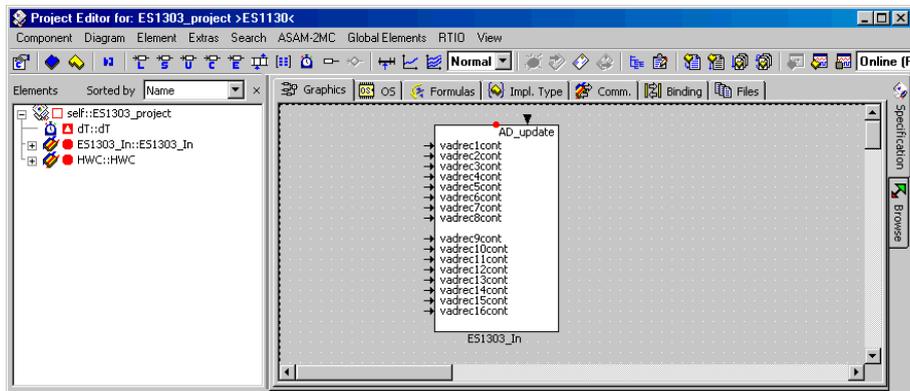
11.3.2 Beispielprojekt

Öffnen des Übungsbeispiels:

- Wählen Sie im Komponentenmanager den Ordner `ASCET_RP\RTIOTutorial\ES1303`.



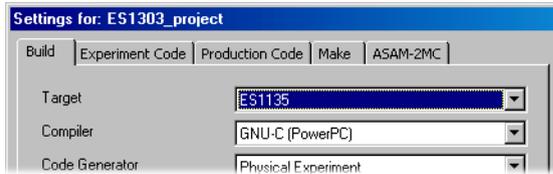
- Öffnen Sie das Projekt `ES1303_project`.





- Klicken Sie auf die Schaltfläche **Specify Code Generation Options**.

Das Fenster „Settings for: ES1303_project“ öffnet sich.



- Wählen Sie im Register „Build“ die Optionen
Target: >ES1130< oder >ES1135<,
Compiler: GNU-C (PowerPC).

Hinweis

Nur Messages, die als „Exported“ deklariert sind, stehen für die RTIO-Kommunikation zur Verfügung.

Im Beispielprojekt sind die für die RTIO-Einbindung der ES1303 notwendigen Vorbereitungen (Tasks, Messages und HWC-Modul) bereits getroffen.

11.3.3 Erstellen der Hardware-Konfiguration

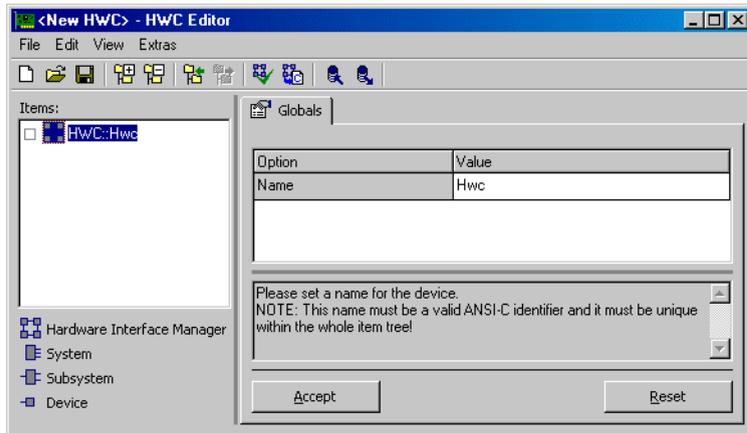
Hinweis

Normalerweise müssen Sie das C-Code-Modul HWC anlegen und in das Projekt einbinden, bevor Sie die Hardware-Konfiguration bearbeiten. Im Tutorial ist Ihnen dieser Schritt jedoch abgenommen.

Öffnen des HWC-Editors

- Wählen Sie im Projekteditor **RTIO** → **Open Editor**.

Der HWC-Editor wird geöffnet.



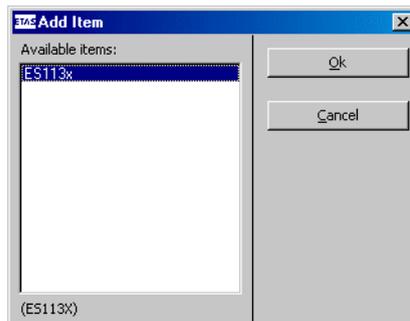
Hardware-Konfiguration (HWC) erstellen:

Die Hardware muss in der Items-Liste als baumartige Struktur beschrieben werden. Das Item HWC ist immer vorhanden und bildet die Wurzel des Baumes.

- Wählen Sie im HWC-Editor **Edit** → **Add Item** oder

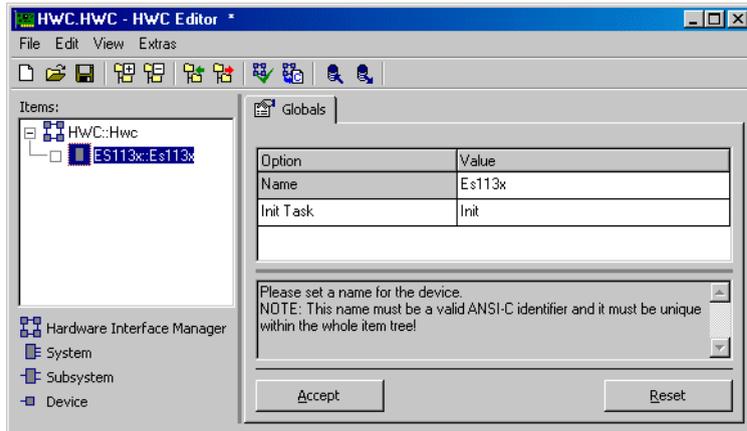


- klicken Sie auf die Schaltfläche **Add Item**. Das Fenster „Add Item“ wird angezeigt.



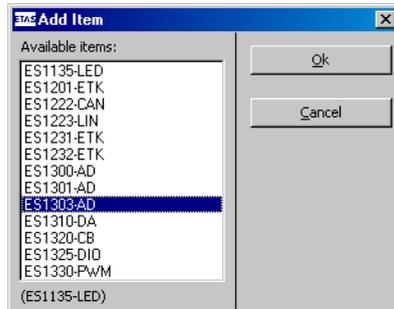
Über **Add Hardware Item...** wird immer die Liste der verfügbaren Items der nächsten Hierarchiestufe geöffnet.

- Wählen Sie den Eintrag ES113x aus.
Dieser Eintrag dient der Beschreibung des ES1000.x-Systems mit integriertem ES1130 oder ES1135 PowerPC-Rechenknoten.
- Klicken Sie **OK**.
In der Liste „Items“ wird das Element ES113x hinzugefügt.



- Wählen Sie in der Liste „Items“ das Element ES113x.
Im Register „Globals“ wird für die Option `Init Task` der Task-Name `Init` vorgegeben. Im OS-Editor des Beispielprojekts gibt es eine gleichnamige `Init-Task`, infolgedessen muss hier keine andere `Init-Task` ausgewählt werden.

- Öffnen Sie nun über **Edit** → **Add Item** die Liste der verfügbaren Elemente der nächsten Hierarchiestufe.



- Wählen Sie den Eintrag **ES1303-AD** aus, welcher der Beschreibung der ES1303 Analog nach Digital Schnittstelle dient.
- Klicken Sie **OK**.

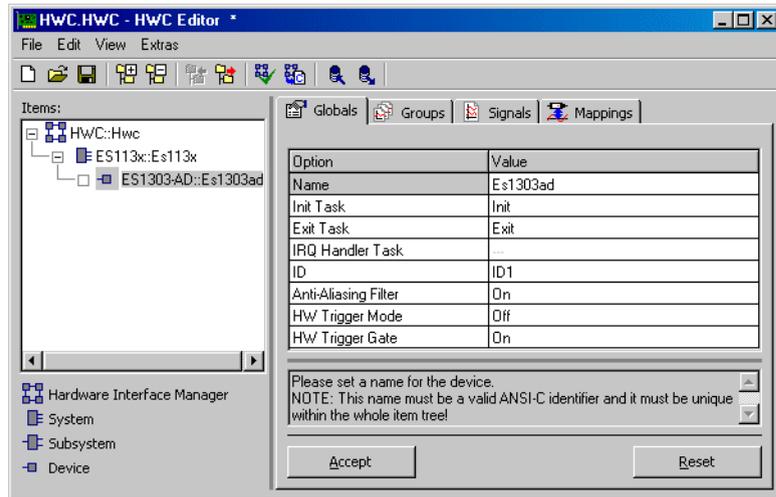
Nun ist der Item-Baum für die Beschreibung des Beispielsystems komplett spezifiziert.

11.3.4 HWC-Einstellungen für die ES1303 vornehmen

Im Register „Globals“ müssen nun Task-Einstellungen vorgenommen werden.

Einstellungen im Register „Globals“ vornehmen:

- Wählen Sie in der Liste „Items“ das Element **ES1303-AD**.
- Wählen Sie das Register „Globals“.



Im Register „Globals“ sind für die Optionen `Init Task` und `Exit Task` die Task-Namen `Init` und `Exit` vorgegeben. Im OS-Editor des Beispielprojekts gibt es die gleichlautenden Tasks, infolgedessen müssen hier keine anderen Tasks ausgewählt werden.

Für die `IRQ Handler Task` ist kein Standardname vorgegeben. Wenn Sie den Hardwaretrigger (`HW Trigger Mode`) verwenden, können Sie eine Task auswählen.

Die Nummer der Karte wird in der Option `ID` automatisch eingestellt. Da das Element `ES1303-AD` das erste dieses Typs ist, ist der Wert „ID1“ voreingestellt.

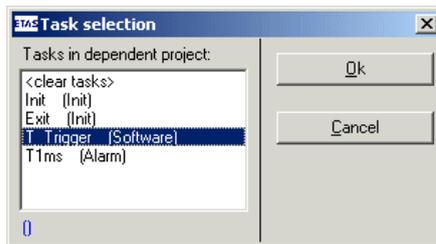
- Wenn Sie die Bandbreite des Eingangssignals begrenzen wollen, wählen Sie für die Option `Anti-Aliasing Filter` den Wert `On`.

- Wählen Sie in der Option HW Trigger Mode die Flanke des ersten Triggersignals, mit der Sie die Messung starten wollen (z.B. Falling Edge).
- Legen Sie in der Option HW Trigger Gate fest, ob das zweite Triggersignal berücksichtigt wird (On) oder nicht (Off).

Hinweis

Für den Polling-Modus (Register „Groups“, IRQ=No) sind die Optionen HW Trigger Mode und HW Trigger Gate nicht von Bedeutung.

- Doppelklicken Sie in das leere Feld neben der Option IRQ Handler Task.
Das Fenster „Task selection“ öffnet sich.



- Wählen Sie die Software-Task **T_Trigger**.
Die Tabelle zeigt eine mögliche Konfiguration der Karte.

Option	Value
Name	ES1303ad
Init Task	Init
Exit Task	Exit
IRQ Handler Task	T_Trigger
ID	ID1
Anti-Aliasing Filter	On
HW Trigger Mode	Falling Edge
HW Trigger Gate	On

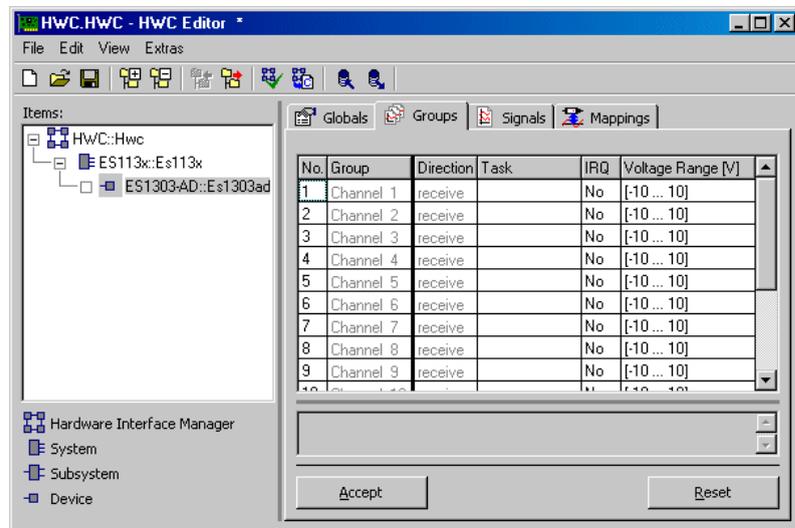
- Wenn Sie alle nötigen Einstellungen vorgenommen haben, klicken Sie im HWC-Editor die Schaltfläche **Accept**, um die Einstellungen zu sichern.

Im Register „Groups“ werden die signalgruppenspezifischen Einstellungen vorgenommen.

Die ES1303 hat für jedes Signal eine fest vorgegebene Signalgruppe (Input). Sie wählen hier für jede Gruppe die Taskzuordnung, den Eingangsspannungsbereich und die Empfangsart (interruptgesteuert oder Polling) aus.

Einstellungen im Register „Groups“ vornehmen:

- Wählen Sie die Registerkarte „Groups“.

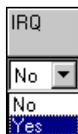


Nehmen Sie die folgenden Schritte für jedes Signal vor, das Sie verwenden wollen.

1. Empfangsart auswählen

- Wenn Sie ein Signal im Interrupt-Betrieb empfangen wollen, doppelklicken Sie in der entsprechenden Zeile in das Feld „IRQ“.

Ein Kombikästchen erscheint.

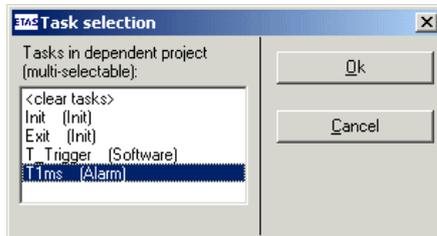


- Wählen Sie aus dem Kombikästchen den Wert Yes.

Die geänderte Auswahl wird im Feld „IRQ“ angezeigt. Gleichzeitig wird das Feld „Task“ zurückgesetzt und gesperrt.

No.	Group	Direction	Task	IRQ	Voltage Range [V]
1	Channel 1	receive	---	Yes	[-10 ... 10]
2	Channel 2	receive		No	[-10 ... 10]

- Wenn Sie ein Signal im Polling-Betrieb empfangen wollen, verwenden Sie die Standard-einstellung No in der Spalte „IRQ“.
2. Zuordnen einer Task (nur für Empfangsart Polling)
- Doppelklicken Sie in der gewünschten Zeile in das Feld „Task“, um das Auswahlfenster „Task selection“ zu öffnen.
 - Wählen Sie in der Task-Auswahlliste die Task T1ms. In dieser Task soll die Datenübertragung erfolgen.

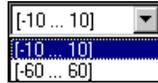


In der Task-Auswahlliste können auch mehrere Tasks ausgewählt werden. Für die ES1303 sind dies üblicherweise Alarm-Tasks.

- Klicken Sie **OK**.
Die ausgewählte(n) Task(s) werden in der Spalte „Task“ angezeigt.

3. Spannungsbereich auswählen

- Um den Eingangsspannungsbereich eines Signals festzulegen, doppelklicken Sie in der entsprechenden Zeile in das Feld „Voltage Range“.



Ein Kombikästchen erscheint.

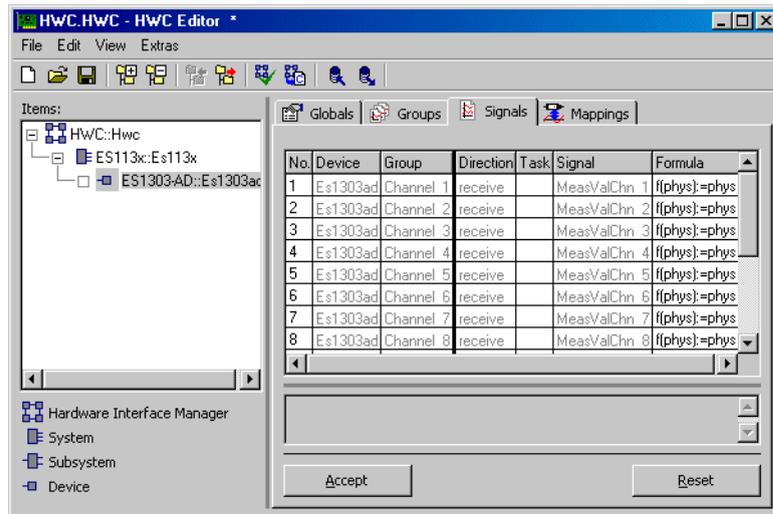
- Wählen Sie aus dem Kombikästchen einen Wert für den Spannungsbereich.

Die Auswahl wird im Feld „Voltage Range“ angezeigt.

No.	Group	Direction	Task	IRQ	Voltage Range [V]
1	Channel 1	receive	---	Yes	[-10 ... 10]
2	Channel 2	receive		No	[-60 ... 60]

- Wenn Sie alle nötigen Einstellungen vorgenommen haben, klicken Sie im HWC-Editor auf **Accept**, um die Einstellungen zu sichern.

Hinter der Registerkarte „Signals“ verbergen sich signalspezifische Einstellungen. Da die ES1303-Karte hier keine besonderen Optionen besitzt, muss hier keine Einstellung vorgenommen werden. Sie können das Register aktivieren, um die aktuellen Einstellungen zu sehen.



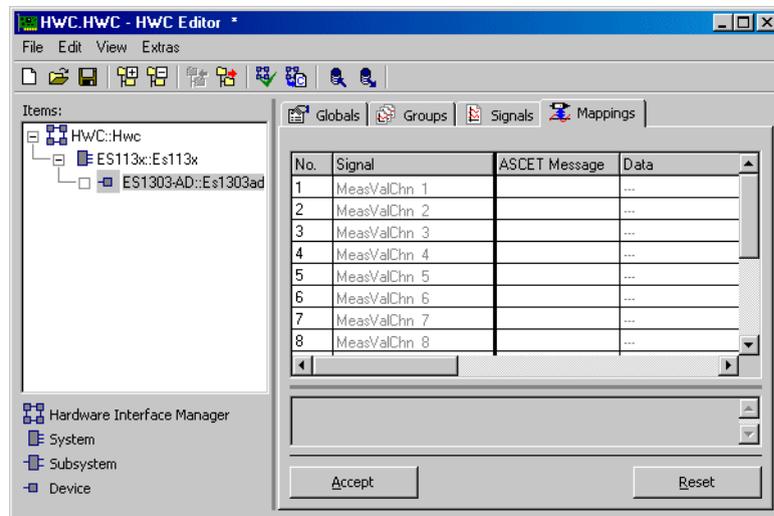
In der Registerkarte „Mappings“ werden die Beziehungen zwischen Signal und Message definiert. Durch Anklicken der gewünschten Zelle in der Spalte „ASCET Message“ kann ein Auswahldialog auf die gefundenen Messages des Restmodells geöffnet werden. Im Auswahldialog werden nur Messages aufgelistet, die das Attribut *Exported* besitzen und der Übertragungsrichtung der Signalgruppe (*send* oder *receive*) entsprechen; also:

Direction = receive → Receive-Messages

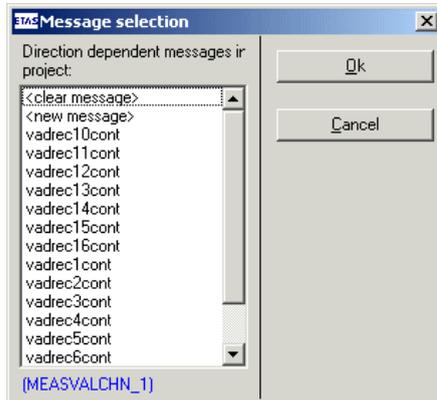
Direction = send → Send-Messages (für die ES1303 ohne Belang)

Einstellungen im Register „Mappings“ vornehmen:

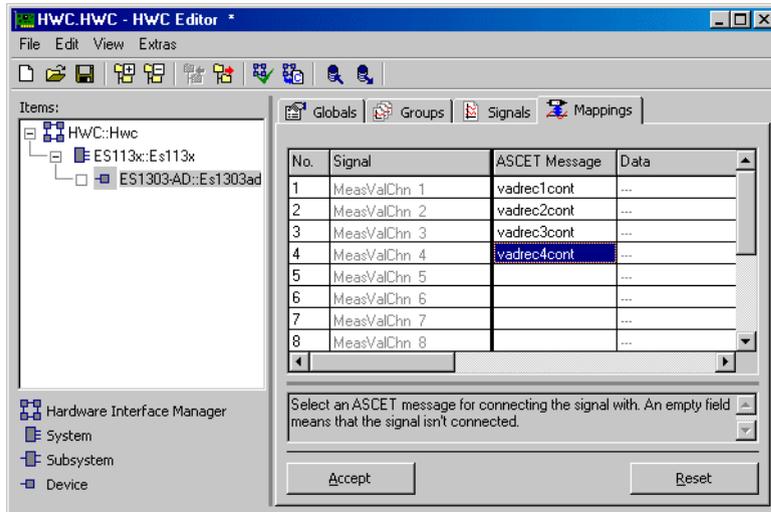
- Wählen Sie die Registerkarte „Mappings“.



- Doppelklicken Sie in der gewünschten Zeile in das Feld „ASCET Message“, um das Auswahl-fenster „Message selection“ zu öffnen.



- Ordnen Sie mit der Message-Auswahl-liste nacheinander die sechzehn Receive-Messages den jeweiligen Signalen zu.



- Klicken Sie im HWC-Editor auf **Accept**.

11.3.5 Hardware-Konfiguration speichern

Die erstellte Hardware-Konfiguration kann im File-Container des Projekts oder als DOS-Datei (Erweiterung * .Hwx) gespeichert werden. Wie das geht, ist in Kapitel 11.2.5 auf Seite 304 beschrieben.

Hinweis

Die Lösung des Beispiels ist unter dem Namen ES1303.hwx abgelegt. Achten Sie darauf, diese nicht zu überschreiben.

11.3.6 Code für HWC-Modul erzeugen

Anschließend muss die Generierungssequenz für das HWC-Modul gestartet werden. Wie das geht, ist in Kapitel 11.2.6 auf Seite 306 beschrieben.

11.3.7 Abschließende Aktionen

Alle RTIO-spezifischen Aktionen sind nun beendet, somit kann der HWC-Editor geschlossen werden. Als weiterer Arbeitsgang schließt sich nun noch die normale Codegenerierung für das Experimentaltarget an, die aus dem Projekteditor gestartet wird.

Hinweis

Es empfiehlt sich, das HWC-Modul nicht in die grafische Darstellung des Projekteditors mit aufzunehmen, da dieses Modul durch jeden RTIO-Generierungsprozess verändert wird. Man erhält eine ungünstige grafische Darstellung des HWC-Moduls.

Codegenerierung für das Experimentaltarget:

- Wählen Sie im Projekteditor **Component** → **Build**, um den Code für das gesamte Projekt zu erzeugen.
- Wählen Sie **Component** → **View Generated Code**, um den generierten Code zu betrachten.

Online experimentieren:



- Wählen Sie im Projekteditor aus dem Kombikästchen „Experiment Target“ den Eintrag **Online (RP)**.

Der Eintrag **Offline (RP)** ist für Offline-Experimente auf dem Target gedacht.

- Wählen Sie **Component** → **Open Experiment**.
- Wählen Sie im Environment-Browser das Environment **Tutorial**.
- Wählen Sie im Fenster „Physical Experiment“ **Experiment** → **Start ERCOS**.
- Wählen Sie im Fenster „Physical Experiment“ **Experiment** → **Start Measurement**.

Die analogen Messwerte der ES1303 werden in einem ASCET-Oszilloskop aufgenommen.

11.4 Tutorial – ES1325 (ohne Trigger)

Das Beispiel zur ES1325 ohne Verwendung des Triggers enthält Modelle zu allen wichtigen Anwendungen der Karte. Damit das Blockdiagramm übersichtlich bleibt, sind die Modelle in grafischen Hierarchien zusammengefasst.

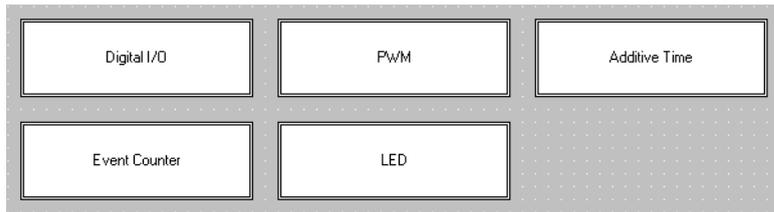


Abb. 11-4 ES1325 – Modellübersicht

Digital I/O: Der Eingangskanal 3 wird eingelesen und an den Ausgangskanal 3 weitergegeben. Das Signal wird invertiert, indem beim Einlesen dem Tiefpegel der Status „aktiv“ zugeordnet ist, während beim Senden dem Hochpegel der Status „aktiv“ zugeordnet ist (siehe „Active State“ auf Seite 251).

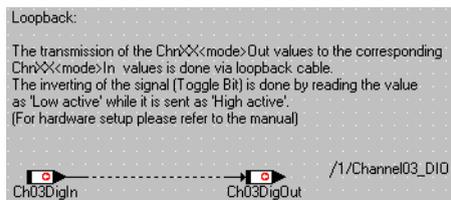


Abb. 11-5 ES1325 – Digital I/O

PWM: Kanal 4 beinhaltet ein statisches PWM-Signal mit festen aktiven und inaktiven Zeiten. Kanal 5 beinhaltet ein dynamisches PWM-Signal, dessen Tastverhältnis von einem Sägezahnsignal bestimmt wird.

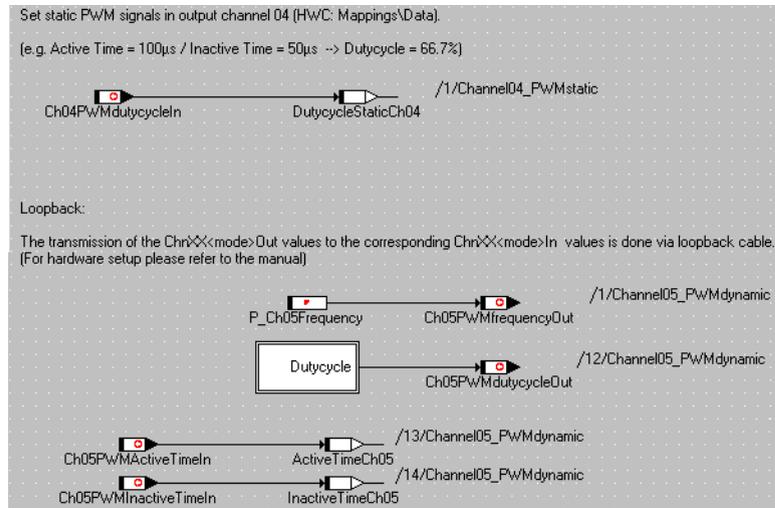


Abb. 11-6 ES1325 – PWM

Additive Time: Das PWM-Ausgangssignal von Kanal 5 wird auch vom Ausgangskanal 6 übertragen. Hier dient es der additiven Zeitmessung, das heißt, der Aufsummierung der aktiven Phasen des Signals während der Tasklaufzeit.

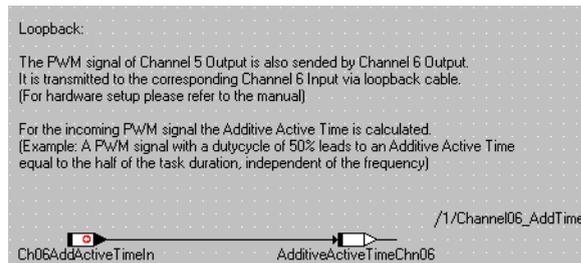


Abb. 11-7 ES1325 – Additive Time

Event Counter: Kanal 7 zählt jede steigende Flanke eines Ereignisses, Kanal 8 zählt jede Flanke eines Ereignisses. Beide Kanäle werden über den Parameter P_Ch0708DigOut gesteuert.

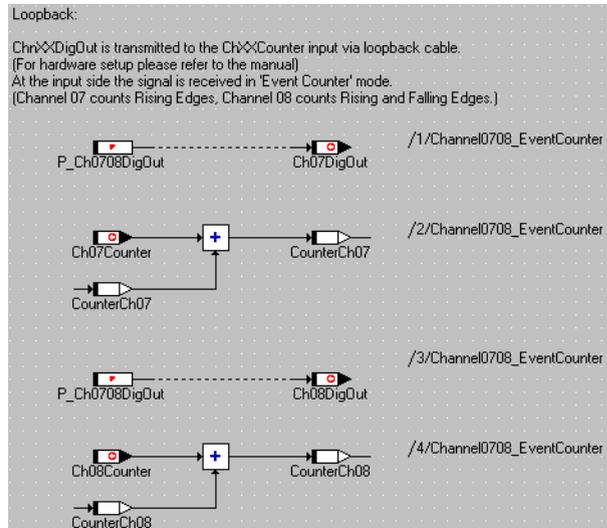


Abb. 11-8 ES1325 – Event Counter

LED: Hier werden die Eingangssignale für die LEDs 1 bis 3 gesetzt. LED1 wird über einen Parameter vom Typ `log` gesteuert, LED2 über einen Parameter vom Typ `sdisc`, und LED3 über einen Parameter vom Typ `cont`.

Hinweis

Die Verwendung von `cont`- und `sdisc`-Werten zur Steuerung der LEDs ist zwar möglich, wird aber ausdrücklich **nicht empfohlen**. Im Sinne eines guten Modellier-/Programmierstils ist eine Abbildung auf eine logische Größe, z.B. durch Fallunterscheidung) vorzuziehen.

LED4 wird über die Message Ch03DigOut gesteuert.

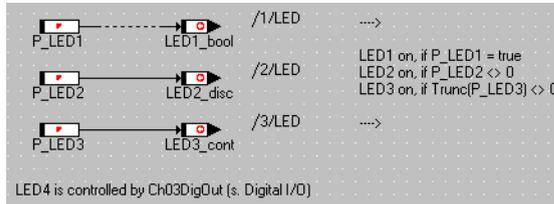
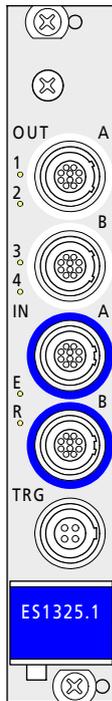


Abb. 11-9 ES1325 – PWM

11.4.1 Die ES1325 Karte

Die folgende Abbildung zeigt die Frontplatte der Karte ES1325:



Die Buchsen Out-A und Out-B enthalten jeweils 8 Ausgangskanäle, die Buchsen In-A und In-B enthalten jeweils 8 Eingangskanäle. TRG enthält 2 Trigger-eingänge. Die LEDs 1 – 4 stehen für Anzeigezwecke zur Verfügung.

Anschlüsse

Damit das Experiment funktioniert, achten Sie auf die richtige Verkabelung. Sie verwenden im Tutorial die Buchsen Out-A, IN-A und (in Kapitel 11.5) TRG; die hier angegebene Verkabelung gilt für beide Aufgaben zur ES1325.

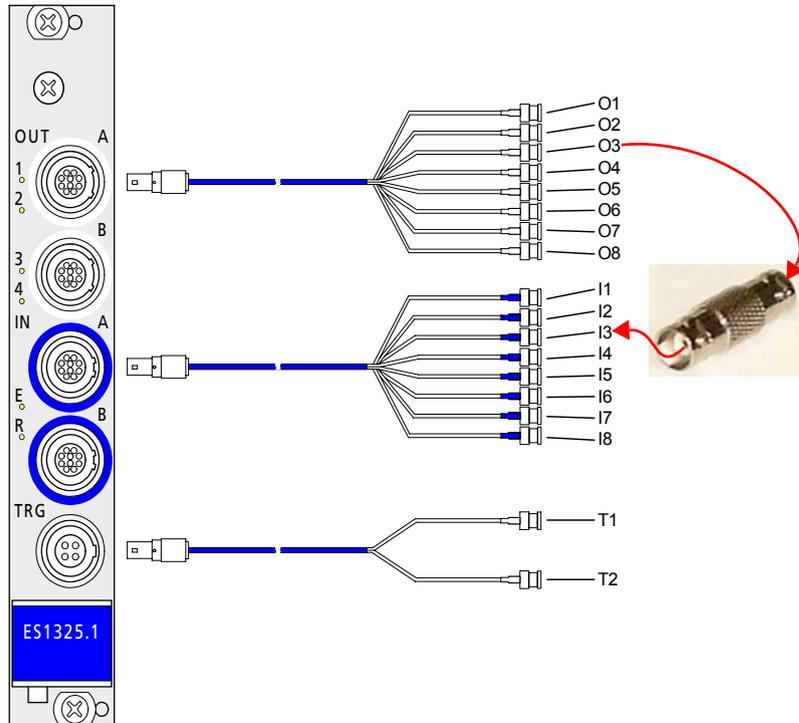


Abb. 11-10 Verkabelung der ES1325

Die nachstehende Tabelle gibt an, welche Ausgänge mit welchen Eingängen verbunden werden. Die Eingänge I1 und I2 bleiben leer.

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

Hinweis

Die Verbindungsstücke für die Ein- und Ausgänge müssen vom Anwender beschafft werden..

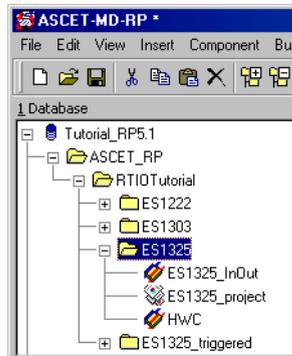
Betrieb mehrerer ES1325-Karten

In einer ES1000 können bis zu 4 ES1325-Karten betrieben werden.

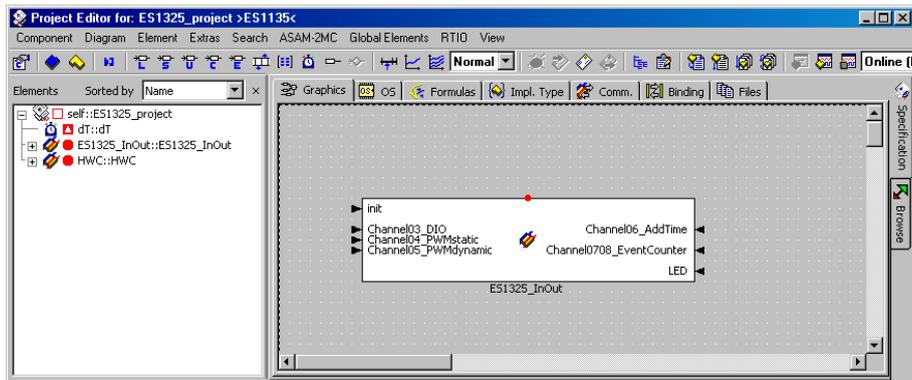
11.4.2 Beispielprojekt

Öffnen des Übungsbeispiels:

- Wählen Sie im Komponentenmanager den Ordner `ASCET_RP\RTIOTutorial\ES1325`.



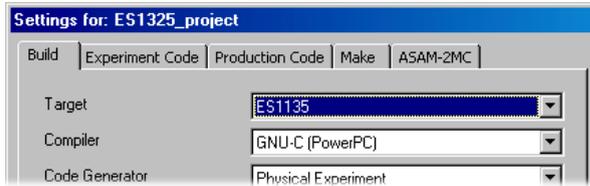
- Wählen Sie das Projekt `ES1325_project`.
- Öffnen Sie das Projekt.





- Klicken Sie auf die Schaltfläche **Specify Code Generation Options**.

Das Fenster „Settings for: ES1325_project“ öffnet sich.



- Wählen Sie im Register „Build“ die Optionen
Target: >ES1130< oder >ES1135<,
Compiler: GNU-C (PowerPC).

Hinweis

Nur Messages, die als „Exported“ deklariert sind, stehen für die RTIO-Kommunikation zur Verfügung.

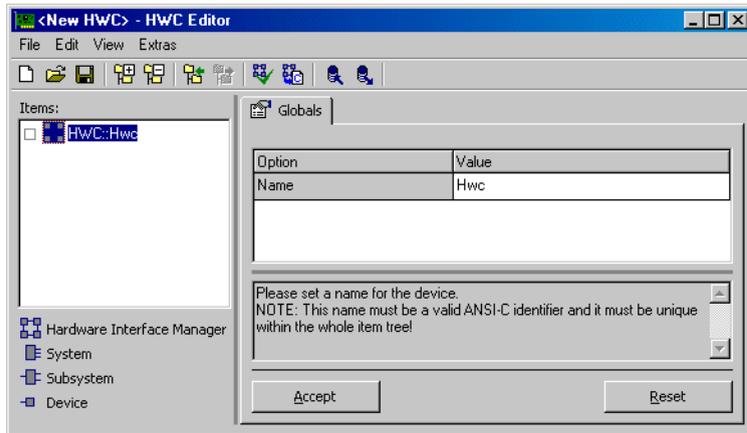
11.4.3 Erstellen der Hardware-Konfiguration

Hinweis

Normalerweise müssen Sie das C-Code-Modul HWC anlegen und in das Projekt einbinden, bevor Sie die Hardware-Konfiguration bearbeiten. Im Tutorial ist Ihnen dieser Schritt jedoch abgenommen.

Öffnen des HWC-Editors

- Wählen Sie im Projekteditor **RTIO** → **Open Editor**.
Der HWC-Editor wird geöffnet.



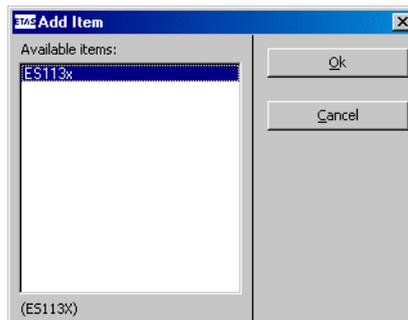
Hardware-Konfiguration (HWC) erstellen:

Die Hardware muss in der Items-Liste als baumartige Struktur beschrieben werden. Das Item HWC ist immer vorhanden und bildet die Wurzel des Baumes.

- Wählen Sie im HWC-Editor **Edit** → **Add Item** oder

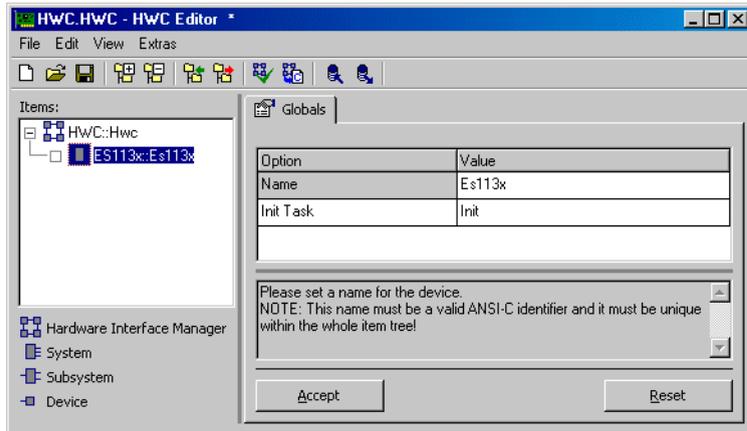


- klicken Sie auf die Schaltfläche **Add Item**. Das Fenster „Add Item“ wird angezeigt.



Über **Add Item** wird immer die Liste der verfügbaren Items der nächsten Hierarchiestufe geöffnet.

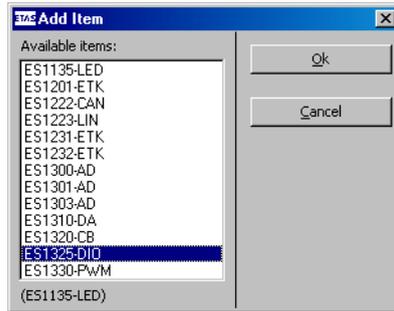
- Wählen Sie den Eintrag ES113x aus.
Dieser Eintrag dient der Beschreibung des ES1000.x-Systems mit integriertem ES1130 oder ES1135 PowerPC-Rechenknoten.
- Klicken Sie **OK**.
In der Liste „Items“ wird das Element ES113x hinzugefügt.



- Wählen Sie in der Liste „Items“ das Element ES113x.
Im Register „Globals“ wird für die Option `Init Task` der Task-Name `Init` vorgegeben. Im OS-Editor des Beispielprojekts gibt es eine gleichnamige `Init-Task`, infolgedessen muss hier keine andere `Init-Task` ausgewählt werden.

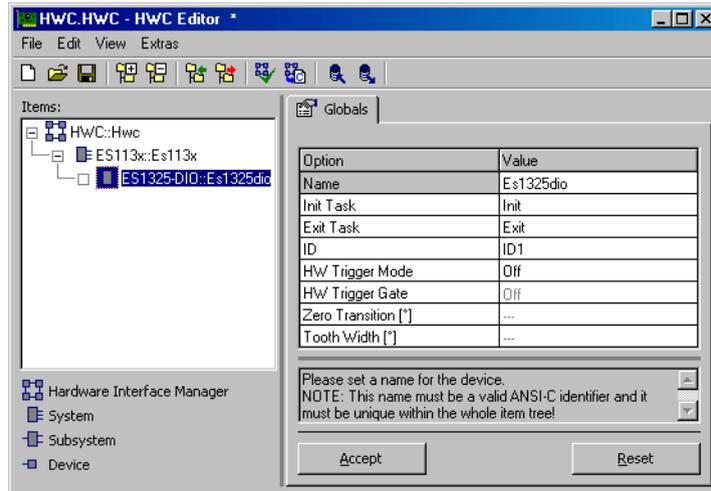
ES1325 einbinden und einrichten:

- Öffnen Sie über **Edit** → **Add Item** die Liste der verfügbaren Elemente der nächsten Hierarchiestufe.



- Wählen Sie den Eintrag **ES1325-DIO** aus. Dieser Eintrag dient der Beschreibung der ES1325-Schnittstellen.
- Klicken Sie **OK**.

In der Liste „Items“ wird das Element **ES1325-DIO** hinzugefügt.



- Wählen Sie in der Liste „Items“ das Element ES1325-DIO.

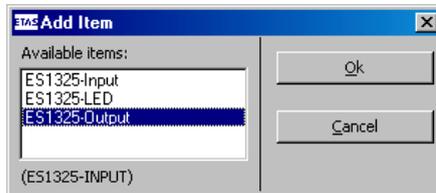
Im Register „Globals“ werden für die Optionen `Init Task` und `Exit Task` die Task-Namen `Init` und `Exit` vorgegeben. Im OS-Editor des Beispielprojekts gibt es gleichnamige Init-Tasks, infolgedessen müssen hier keine anderen Tasks ausgewählt werden.

Die Nummer der Karte wird in der Option `ID` automatisch eingestellt. Da das Element ES1325-DIO das erste dieses Typs ist, ist der Wert „ID1“ voreingestellt.

- Übernehmen Sie die Einstellung mit **Accept**.

Geräte anlegen:

- Wählen Sie in der Liste „Items“ das Element ES1325-DIO.
- Öffnen Sie über **Edit** → **Add Item** die Liste der verfügbaren Elemente der nächsten Hierarchiestufe.



- Wählen Sie den Eintrag `ES1325-Input` und klicken Sie auf **OK**.

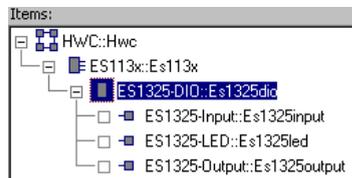
Dieser Eintrag dient der Beschreibung der Eingangskanäle der ES1325.

- Fügen Sie auch die Geräte `ES1325-Output` und `ES1325-LED` hinzu.

`ES1325-Output` dient der Beschreibung der Ausgangskanäle der ES1325, und `ES1325-LED` dient der Beschreibung der LEDs.

Für dieses Beispiel verwenden Sie für alle drei Geräte die Standardeinstellungen im Register „Globals“.

Nun ist der Item-Baum für die Beschreibung des Beispielsystems komplett spezifiziert.



11.4.4 HWC-Einstellungen für die ES1325 vornehmen

Nun müssen die Aus- und Eingänge sowie die LEDs konfiguriert werden. Die Ausgänge liefern die Signale, die die Eingänge empfangen. Die LEDs dienen der Anzeige.

Register „Globals“: Im Register „Globals“ brauchen Sie für keins der drei Geräte Einstellungen vorzunehmen.

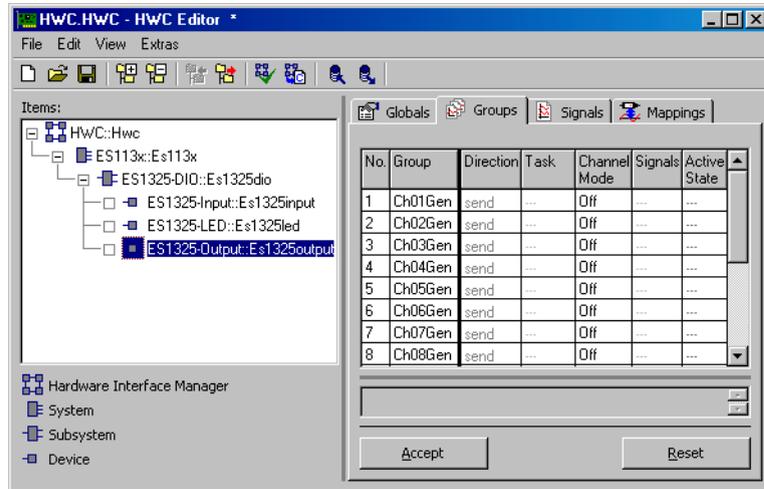
Ausgänge – Gerät `ES1325-Output`

Register „Groups“: Im Register „Groups“ werden die signalgruppenspezifischen Einstellungen vorgenommen.

Das Device `ES1325-Output` hat für jedes Ausgangssignal eine vorgegebene Signalgruppe (`Ch<n>Gen`, `<n> = 01 – 16`). Sie stellen hier für jede verwendete Gruppe den Modus, die Taskzuordnung, die generierten Signale sowie die verwendeten Pegel und Flanken ein.

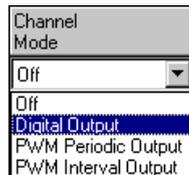
Einstellungen im Register „Groups“ vornehmen:

- Wählen Sie das Register „Groups“.



Nehmen Sie die folgenden Schritte für die Signalgruppen 3 bis 8 vor.

1. Kanalmodus auswählen



- Wählen Sie in der Spalte „Channel Mode“ für die verschiedenen Gruppen folgende Modi:

Gruppe

Channel Mode

Ch03Gen, Ch07Gen, Digital Output
Ch08Gen

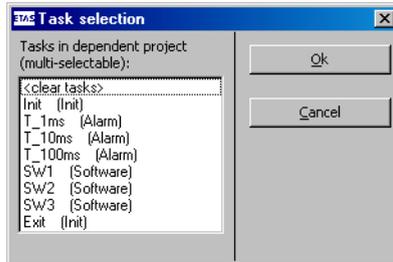
Ch04Gen, Ch05Gen, PWM Periodic Output
Ch06Gen

In der Spalte „Active State“ wird der Standardwert für den jeweiligen Modus eingetragen.

In der Spalte „Task“ können Sie den verwendeten Gruppen nun eine Task zuordnen, in der die Signalübertragung stattfinden soll.

2. Zuordnen einer Task

- Doppelklicken Sie in die gewünschte Zeile, Spalte „Task“, um das Auswahlfenster „Task selection“ zu öffnen.



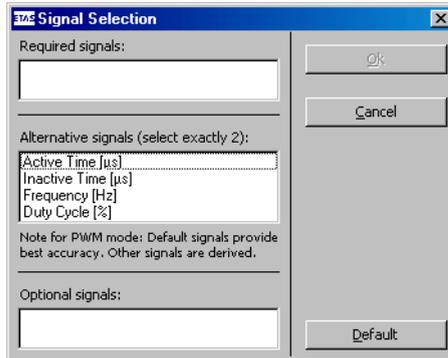
- Wählen Sie in der Task-Auswahlliste die folgenden Tasks für die Datenübertragung.

Gruppe	Task
Ch03Gen, Ch04Gen	T_100ms
Ch05Gen, Ch06Gen, Ch07Gen, Ch08Gen	T_10ms

- Klicken Sie **OK**.
Die ausgewählten Tasks werden in der Spalte „Task“ angezeigt.

3. Signale auswählen

- Doppelklicken Sie in der gewünschten Zeile; Spalte „Signals“, um das Fenster „Signal Selection“ zu öffnen.



Eine Beschreibung der Signale finden Sie im Abschnitt „Signals“ auf Seite 249.

- Wählen Sie folgende Signale aus.

Gruppe	Signals
Ch03Gen, Ch07Gen, Ch08Gen	Standardeinstellung übernehmen
Ch04Gen	Active Time [µs], Inactive Time [µs]
Ch05Gen, Ch06Gen	Frequency [Hz], Duty Cycle [%]

- Bestätigen Sie die Auswahl mit **OK**.

4. Pegel auswählen

Da alle Gruppen die Standardeinstellung **High** verwenden, brauchen Sie in der Spalte „Active State“ keine Änderungen vornehmen.

- Wenn Sie alle nötigen Einstellungen vorgenommen haben, klicken Sie im HWC-Editor auf **Accept**, um die Einstellungen zu sichern.

Wenn Sie die Signalgruppen eingerichtet haben, sollte das Register so aussehen:

No.	Group	Direction	Task	Channel Mode	Signals	Active State
1	Ch01Gen	send	---	Off	---	---
2	Ch02Gen	send	---	Off	---	---
3	Ch03Gen	send	T_100ms	Digital Output	[Select]	High
4	Ch04Gen	send	T_100ms	PWM Periodic Output	[Select]	High
5	Ch05Gen	send	T_10ms	PWM Periodic Output	[Select]	High
6	Ch06Gen	send	T_10ms	PWM Periodic Output	[Select]	High
7	Ch07Gen	send	T_10ms	Digital Output	[Select]	High
8	Ch08Gen	send	T_10ms	Digital Output	[Select]	High

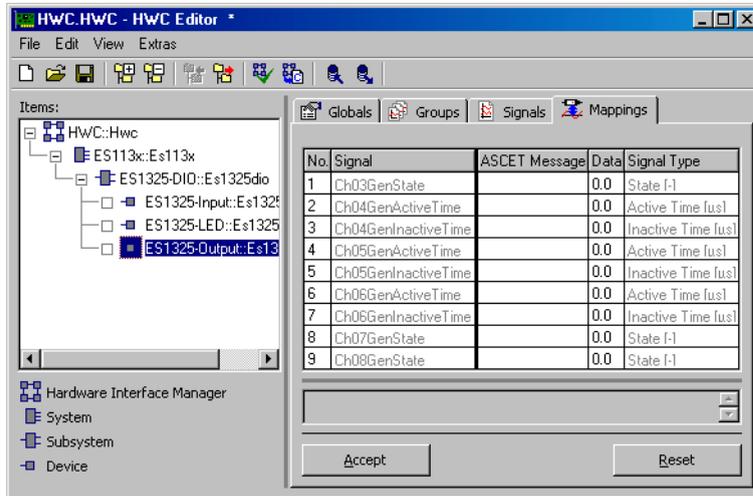
Register „Signals“: In diesem Register sind die Signale enthalten, die Sie im Register „Groups“ generiert haben.

No.	Signal	Formula	Signal Type
1	Ch03GenState	f(phys)=phys	State I-1
2	Ch04GenActiveTime	f(phys)=phys	Active Time [us]
3	Ch04GenInactiveTime	f(phys)=phys	Inactive Time [us]
4	Ch05GenFrequency	f(phys)=phys	Frequency [Hz]
5	Ch05GenDutyCycle	f(phys)=phys	Duty Cycle [%]
6	Ch06GenFrequency	f(phys)=phys	Frequency [Hz]
7	Ch06GenDutyCycle	f(phys)=phys	Duty Cycle [%]
8	Ch07GenState	f(phys)=phys	State I-1
9	Ch08GenState	f(phys)=phys	State I-1

Sie können in diesem Register die Namen und die Formeln der Signale bearbeiten. Für das Beispiel sind jedoch *keine* Änderungen erforderlich.

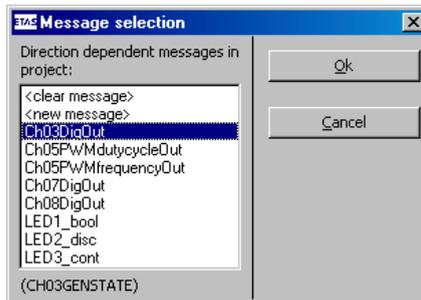
Register „Mappings“: In diesem Register werden die Signale und die ASCET-Messages aus dem Projekt einander zugeordnet. Durch Anklicken der gewünschten Zelle in der Spalte „ASCET Message“ wird ein Auswahldialog geöffnet. Dieser enthält nur Messages, die das Attribut *Exported* besitzen und der Übertragungsrichtung der Signalgruppe (hier: *send*) entsprechen; also:

- Direction = send → Send-Messages



ASCET-Message manuell zuordnen:

- Wählen Sie das Register „Mappings“.
 - Doppelklicken Sie in die gewünschte Zeile, Spalte „ASCET Message“.
- Das Fenster „Message selection“ öffnet sich.



Es enthält alle Send-Messages aus dem ASCET-Projekt.

- Wählen Sie folgende ASCET-Messages für die Signale aus.

Signal	ASCET Message
Ch03GenState	Ch03DigOut
Ch05GenFrequency, Ch06GenFrequency	Ch05PWMfrequencyOut
Ch05GenDutyCycle, Ch06GenDutyCycle	Ch05PWMdutyCycleOut
Ch07GenState	Ch07DigOut
Ch08GenState	Ch08DigOut

Den Signalen Ch04GenActiveTime und Ch04GenInactiveTime wird keine Message zugeordnet, sondern manuell ein fester Wert zugewiesen.

- Klicken Sie auf **OK**.
- Klicken Sie abschließend im HWC-Editor auf **Accept**, um die Einstellungen zu sichern.

Einen Wert manuell eingeben:

Wenn einem Signal keine ASCET-Message zugewiesen wurde, können Sie in der Spalte „Data“ einen festen Wert eingeben.

- Doppelklicken Sie in die gewünschte Zeile, Spalte „Data“.



Das Feld wird zum Eingabefeld.

- Geben Sie die folgenden Werte ein.

Signal	Data
Ch04GenActiveTime	100.0
Ch04GenInactiveTime	50.0

Wenn Sie allen Signalen Messages oder Werte zugewiesen haben, sollte das Register so aussehen:

No.	Signal	ASCET Message	Data	Signal Type
1	Ch03GenState	Ch03DigOut	...	State [-1]
2	Ch04GenActiveTime		100.0	Active Time [us]
3	Ch04GenInactiveTime		50.0	Inactive Time [us]
4	Ch05GenFrequency	Ch05PwMfrequencyOut	...	Frequency [Hz]
5	Ch05GenDutyCycle	Ch05PwMdutyCycleOut	...	Duty Cycle [%]
6	Ch06GenFrequency	Ch05PwMfrequencyOut	...	Frequency [Hz]
7	Ch06GenDutyCycle	Ch05PwMdutyCycleOut	...	Duty Cycle [%]
8	Ch07GenState	Ch07DigOut	...	State [-1]
9	Ch08GenState	Ch08DigOut	...	State [-1]

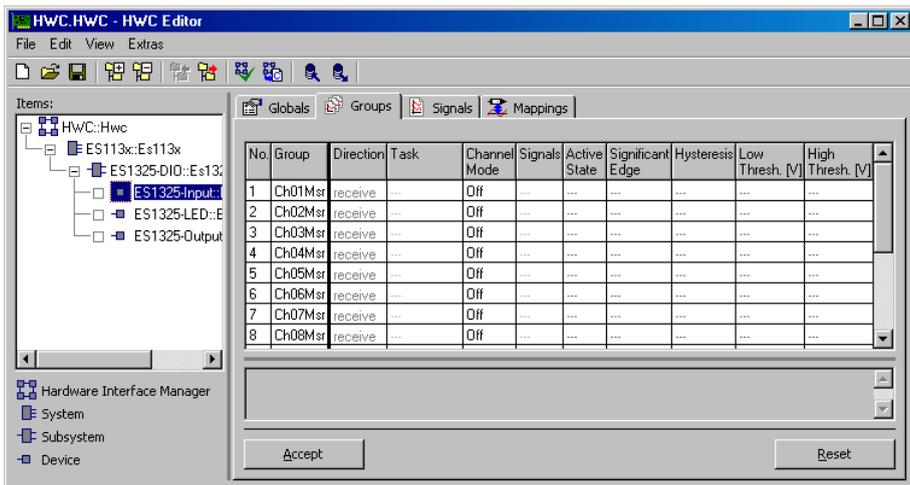
Eingänge – Gerät ES1325 – Input

Register „Groups“: Im Register „Groups“ werden die signalgruppenspezifischen Einstellungen vorgenommen.

Das Device ES1325-Input hat für jedes Eingangssignal eine vorgegebene Signalgruppe (Ch<n>Msr, <n> = 01 – 16). Sie stellen hier für jede verwendete Gruppe den Modus, die Taskzuordnung, die generierten Signale sowie die verwendeten Pegel und Flanken ein.

Einstellungen im Register „Groups“ vornehmen:

- Wählen Sie das Register „Groups“.



Nehmen Sie die folgenden Schritte für die Signalgruppen 3 bis 8 vor.

1. Kanalmodus auswählen

- Doppelklicken Sie in der gewünschten Zeile in die Spalte „Channel Mode“.
Ein Kombikästchen öffnet sich.
- Wählen Sie für die verschiedenen Gruppen folgende Modi:



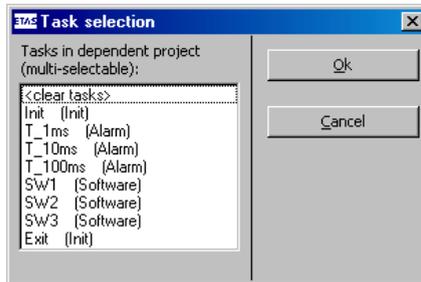
Gruppe	Channel Mode
Ch03Msr	Digital Input
Ch04Msr, Ch05Msr	PWM Input
Ch06Msr	Additive Time
Ch07Msr, Ch08Msr	Event Counter

In den Spalten „Active State“, „Significant Edge“, „Hysteresis“, „Low Thresh. [V]“ und „High Thresh. [V]“ werden die Standardwerte für die jeweiligen Modi eingetragen.

In der Spalte „Task“ können Sie den verwendeten Gruppen nun eine Task zuordnen, in der die Signalübertragung stattfinden soll.

2. Zuordnen einer Task

- Doppelklicken Sie in die gewünschte Zeile, Spalte „Task“, um das Auswahlfenster „Task selection“ zu öffnen.



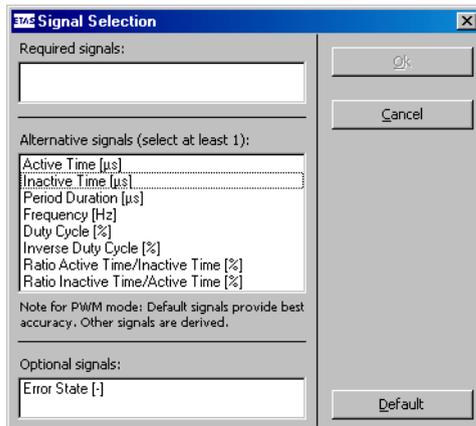
- Wählen Sie in der Task-Auswahlliste die folgenden Tasks, in denen die Datenübertragung erfolgen soll.

Gruppe	Task
Ch03Msr, Ch04Msr	T_100ms
Ch05Msr, Ch06Msr, Ch07Msr, Ch08Msr	T_10ms

- Klicken Sie **OK**.
Die ausgewählten Tasks werden in der Spalte „Task“ angezeigt.

3. Signale auswählen

- Doppelklicken Sie in der gewünschten Zeile; Spalte „Signals“, um das Fenster „Signal Selection“ zu öffnen.



In diesem Fenster legen Sie fest, welche Signale für die Gruppe generiert werden. Eine Beschreibung der Signale finden Sie im Abschnitt „Signals“ auf Seite 249.

- Wählen Sie folgende Signale aus.

Gruppe	Signals
Ch04Msr	Duty Cycle [%]
Ch05Msr	Active Time [μ s], Inactive Time [μ s]
Ch03Msr, Ch06Msr, Ch07Msr, Ch08Msr	Standardeinstellung übernehmen

4. Pegel auswählen

- Bestätigen Sie die Auswahl mit **OK**.
- Doppelklicken Sie in der gewünschten Zeile, Spalte „Active State“ (siehe Seite 251), um einem Pegel des Eingangssignals den aktiven Status zuzuordnen.
Ein Kombikästchen öffnet sich.
- Wählen Sie die folgenden Zustände.



Gruppe	Active State
Ch03Msr	Low
Ch04Msr – Ch08Msr	High

Die Auswahl wird im Feld „Active State“ angezeigt.

5. Flanke auswählen (nur Modi PWM Input und Event Counter)

- Doppelklicken Sie in der gewünschten Zeile, Spalte „Significant Edge“ (siehe Seite 251), um einer Flanke ein Ereignis zuzuordnen.
Ein Kombikästchen öffnet sich.
- Wählen Sie die folgenden Optionen.



Gruppe	Significant Edge
Ch04Msr, Ch05Msr, Ch07Msr	Inactive-Active
Ch08Msr	Both

Die Auswahl wird im Feld „Significant Edge“ angezeigt.

- Wenn Sie alle nötigen Einstellungen vorgenommen haben, klicken Sie im HWC-Editor auf **Accept**, um die Einstellungen zu sichern.

Wenn Sie die Signalgruppen eingerichtet haben, sollte das Register so aussehen:

No.	Group	Direction	Task	Channel Mode	Signals	Active State	Significant Edge	Hysteresis	Low Thresh. [V]	High Thresh. [V]
1	Ch01Msr	receive	---	Off	---	---	---	---	---	---
2	Ch02Msr	receive	---	Off	---	---	---	---	---	---
3	Ch03Msr	receive	T_100ms	Digital Input	[Select]	Low	---	TTL	1.728	2.304
4	Ch04Msr	receive	T_100ms	PwM Input	[Select]	High	Inactive-Active	TTL	1.728	2.304
5	Ch05Msr	receive	T_10ms	PwM Input	[Select]	High	Inactive-Active	TTL	1.728	2.304
6	Ch06Msr	receive	T_10ms	Additive Time	[Select]	High	---	TTL	1.728	2.304
7	Ch07Msr	receive	T_10ms	Event Counter	[Select]	High	Inactive-Active	TTL	1.728	2.304
8	Ch08Msr	receive	T_10ms	Event Counter	[Select]	High	Both	TTL	1.728	2.304

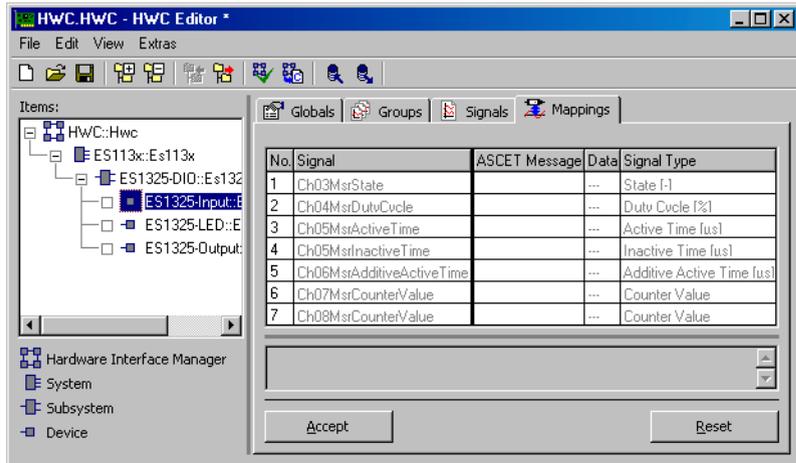
Register „Signals“: In diesem Register sind die Signale enthalten, die Sie im Register „Groups“ generiert haben.

No.	Signal	Formula	Signal Type
1	Ch03MsrState	f(phys)=phys	State [-1]
2	Ch04MsrDutyCycle	f(phys)=phys	Duty Cycle [%]
3	Ch05MsrActiveTime	f(phys)=phys	Active Time [us]
4	Ch05MsrInactiveTime	f(phys)=phys	Inactive Time [us]
5	Ch06MsrAdditiveActiveTime	f(phys)=phys	Additive Active Time [us]
6	Ch07MsrCounterValue	f(phys)=phys	Counter Value
7	Ch08MsrCounterValue	f(phys)=phys	Counter Value

Sie können in diesem Register die Namen und die Formeln der Signale bearbeiten. Für das Beispiel sind jedoch *keine* Änderungen erforderlich.

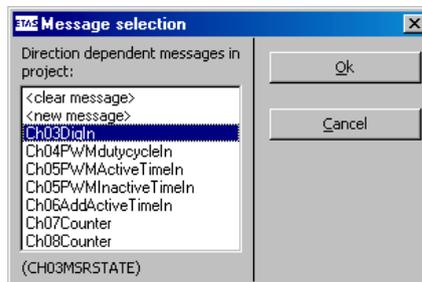
Register „Mappings“: In diesem Register werden die Signale und die ASCET-Messages aus dem Projekt einander zugeordnet. Durch Anklicken der gewünschten Zelle in der Spalte „ASCET Message“ wird ein Auswahldialog geöffnet. Dieser enthält nur Messages, die das Attribut *Exported* besitzen und der Übertragungsrichtung der Signalgruppe (hier: *receive*) entsprechen; also:

- Direction = receive → Receive-Messages



ASCET-Message manuell zuordnen:

- Wählen Sie das Register „Mappings“.
 - Doppelklicken Sie in die gewünschte Zeile, Spalte „ASCET Message“.
- Das Fenster „Message selection“ öffnet sich.



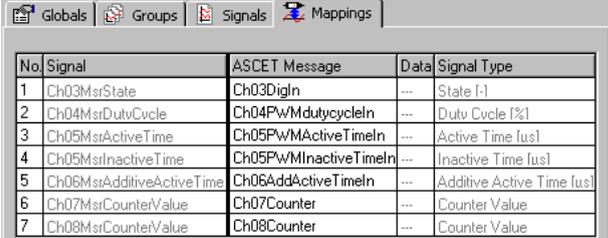
Es enthält alle Receive-Messages aus dem ASCET Projekt.

- Wählen Sie folgende ASCET-Messages für die Signale aus.

Signal	ASCET Message
Ch03MsrState	Ch03DigIn
Ch04MsrDutyCycle	Ch04PWMdutycycleIn
Ch05MsrActiveTime	Ch05PWMActiveTimeIn
Ch05MsrInactiveTime	Ch05PWMInactiveTimeIn
Ch06MsrAdditiveActiveTime	Ch06AddActiveTimeIn
Ch07MsrCounterValue	Ch07Counter
Ch08MsrCounterValue	Ch08Counter

- Klicken Sie auf **OK**.
- Klicken Sie abschließend im HWC-Editor auf **Accept**, um die Einstellungen zu sichern.

Wenn Sie allen Signalen Messages zugewiesen haben, sollte das Register so aussehen:



No.	Signal	ASCET Message	Data	Signal Type
1	Ch03MsrState	Ch03DigIn	---	State [1]
2	Ch04MsrDutyCycle	Ch04PWMdutycycleIn	---	Duty Cycle [%]
3	Ch05MsrActiveTime	Ch05PWMActiveTimeIn	---	Active Time [us]
4	Ch05MsrInactiveTime	Ch05PWMInactiveTimeIn	---	Inactive Time [us]
5	Ch06MsrAdditiveActiveTime	Ch06AddActiveTimeIn	---	Additive Active Time [us]
6	Ch07MsrCounterValue	Ch07Counter	---	Counter Value
7	Ch08MsrCounterValue	Ch08Counter	---	Counter Value

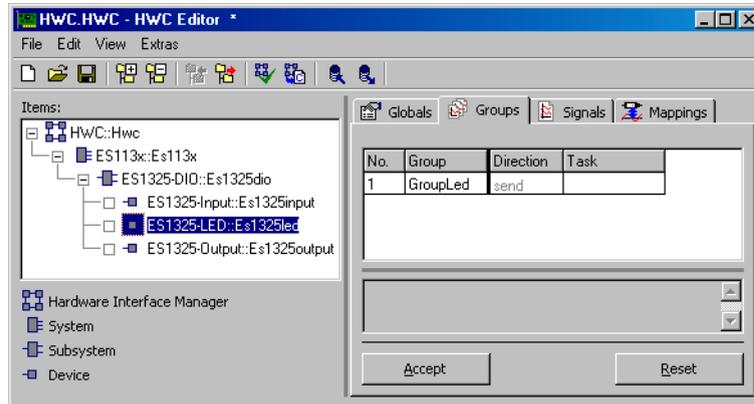
LEDs – Gerät ES1325-LED

Register „Groups“: Im Register „Groups“ werden die signalgruppenspezifischen Einstellungen vorgenommen.

Das Device ES1325-LED hat eine vorgegebene Signalgruppe (GroupLED). Sie stellen hier für diese Gruppe die Taskzuordnung ein.

Einstellungen im Register „Groups“ vornehmen:

- Wählen Sie das Register „Groups“.

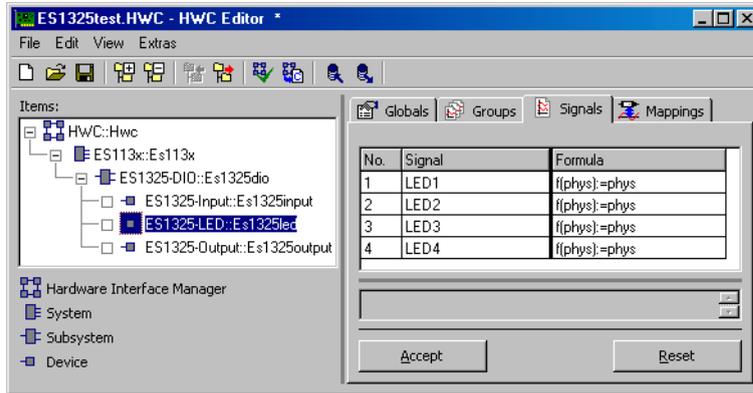


- Doppelklicken Sie in der gewünschten Zeile, Spalte „Task“, um das Auswahlfenster „Task selection“ zu öffnen.
- Wählen Sie in der Task-Auswahlliste die Task T_1ms (Alarm).
- Klicken Sie **OK**.
Die Task wird in der Spalte „Task“ angezeigt.
- Klicken Sie im HWC-Editor auf **Accept**, um die Einstellungen zu sichern.

Das Register sollte nun so aussehen:



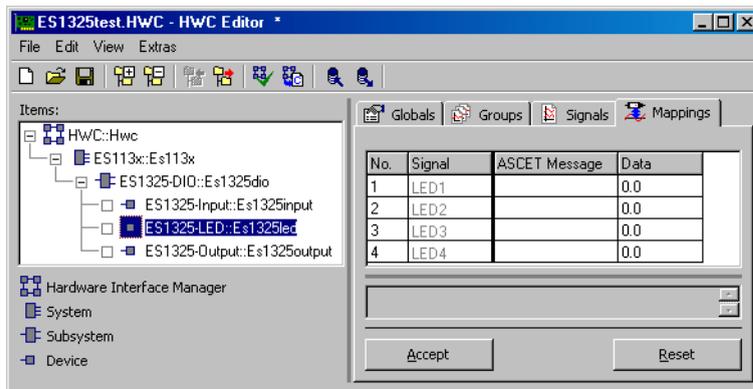
Register „Signals“: In diesem Register sind die Signale enthalten, die zu der Signalgruppe `GROUPLED` gehören. Für jede LED auf der Karte gibt es ein Signal.



Sie können in diesem Register die Namen und die Formeln der Signale bearbeiten. Für das Beispiel sind jedoch *keine* Änderungen erforderlich.

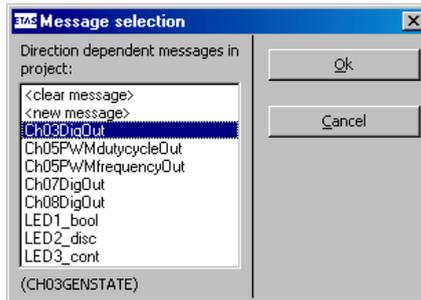
Register „Mappings“: In diesem Register werden die Signale und die ASCET-Messages aus dem Projekt einander zugeordnet. Durch Anklicken der gewünschten Zelle in der Spalte „ASCET Message“ wird ein Auswahldialog geöffnet. Dieser enthält nur Messages, die das Attribut *Exported* besitzen und der Übertragungsrichtung der Signalgruppe (hier: *send*) entsprechen; also:

- Direction = send → Send-Messages



ASCET-Message manuell zuordnen:

- Wählen Sie das Register „Mappings“.
 - Doppelklicken Sie in die gewünschte Zeile, Spalte „ASCET Message“.
- Das Fenster „Message selection“ öffnet sich.



Es enthält alle Send-Messages aus dem ASCET-Projekt.

- Wählen Sie folgende ASCET-Messages für die Signale aus.

Signal ASCET Message

LED1	LED1_bool
LED2	LED2_disc
LED3	LED3_cont
LED4	Ch03DigOut

- Klicken Sie auf **OK**.
- Klicken Sie abschließend im HWC-Editor auf **Accept**, um die Einstellungen zu sichern.

Das Register sollte nun so aussehen:

No.	Signal	ASCET Message	Data
1	LED1	LED1_bool	---
2	LED2	LED2_disc	---
3	LED3	LED3_cont	---
4	LED4	Ch03DigOut	---

11.4.5 Hardware-Konfiguration speichern

Die erstellte Hardware-Konfiguration kann im File-Container des Projekts oder als DOS-Datei (Erweiterung * .Hwx) gespeichert werden. Wie das geht, ist in Kapitel 11.2.5 auf Seite 304 beschrieben.

Hinweis

Die Lösung des Beispiels ist unter dem Namen ES1325.hwx abgelegt. Achten Sie darauf, diese nicht zu überschreiben.

11.4.6 Code für HWC-Modul erzeugen

Anschließend muss die Generierungssequenz für das HWC-Modul gestartet werden. Wie das geht, ist in Kapitel 11.2.6 auf Seite 306 beschrieben.

11.4.7 Experimentieren mit dem Beispielprojekt

Alle RTIO-spezifischen Aktionen sind nun beendet, somit kann der HWC-Editor geschlossen werden. Als weiterer Arbeitsgang schließt sich nun noch die normale Codegenerierung für das Experimentaltarget an, die aus dem Projekteditor gestartet wird.

Hinweis

Es empfiehlt sich, das HWC-Modul nicht in die grafische Darstellung des Projekteditors mit aufzunehmen, da dieses Modul durch jeden RTIO-Generierungsprozess verändert wird. Man erhält eine ungünstige grafische Darstellung des HWC-Moduls.

Codegenerierung für das Experimentaltarget:

- Wählen Sie im Projekteditor **Component** → **Build**, um den Code für das gesamte Projekt zu erzeugen.
- Wählen Sie **Component** → **View Generated Code**, um den generierten Code zu betrachten.

Online experimentieren:

- Falls noch nicht geschehen, verbinden Sie die Ein- und Ausgänge der ES1325 gemäß Abschnitt „Anschlüsse“ auf Seite 330 und schalten Sie die Stromversorgung der ES1000.x ein.

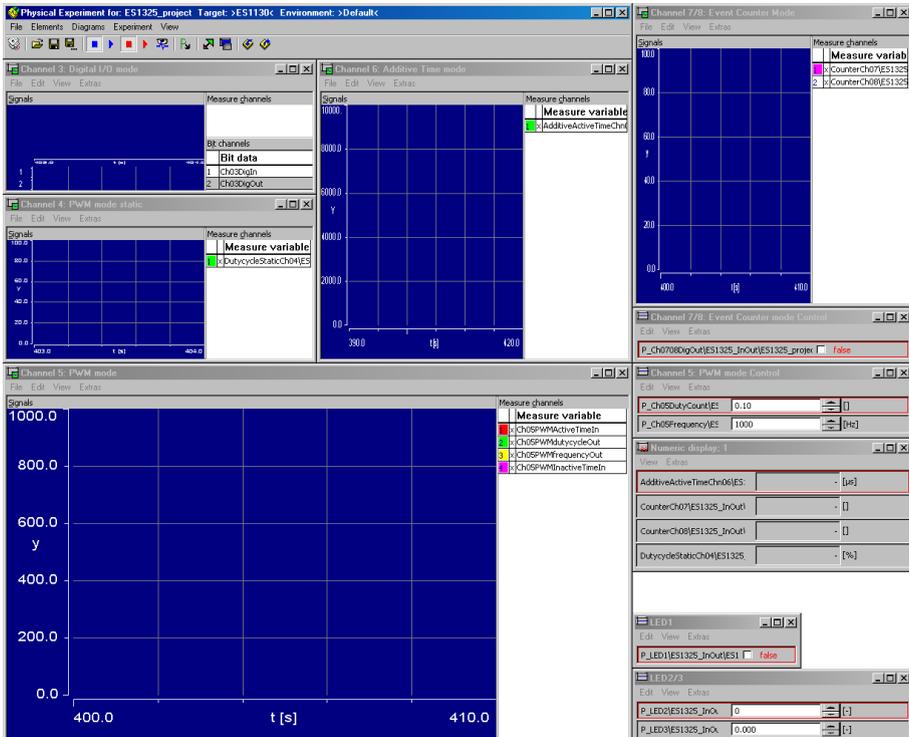


- Wählen Sie im Projekteditor aus dem Kombikästchen „Experiment Target“ den Eintrag `Online (RP)`.

Der Eintrag `Offline (RP)` ist für Offline-Experimente auf dem Target gedacht.

- Wählen Sie **Component** → **Open Experiment**.

Das Fenster „Physical Experiment“ sowie die vordefinierte Experimentierumgebung bestehend aus fünf Oszilloskopen, einer numerischen Anzeige und vier Verstellfenstern öffnen sich.



- Wählen Sie im Fenster „Physical Experiment“ **Experiment** → **Start ERCOS**

oder



- klicken Sie auf die Schaltfläche **Start ERCOS**. Das Betriebssystem wird gestartet, das Modell wird ausgeführt.
- Wählen Sie im Fenster „Physical Experiment“ **Experiment** → **Start Measurement**

oder

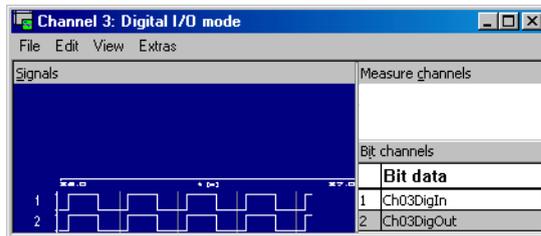


- klicken Sie auf die Schaltfläche **Start Measurement**.

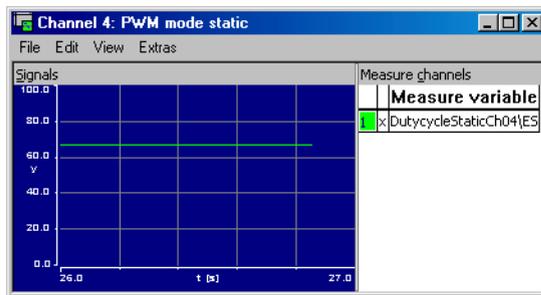
Die Werte der ASCET-Messages werden in den Oszilloskopen und in der numerischen Anzeige angezeigt.

Im folgenden werden die Anzeigen und Verstellmöglichkeiten der einzelnen Modellblöcke beschrieben.

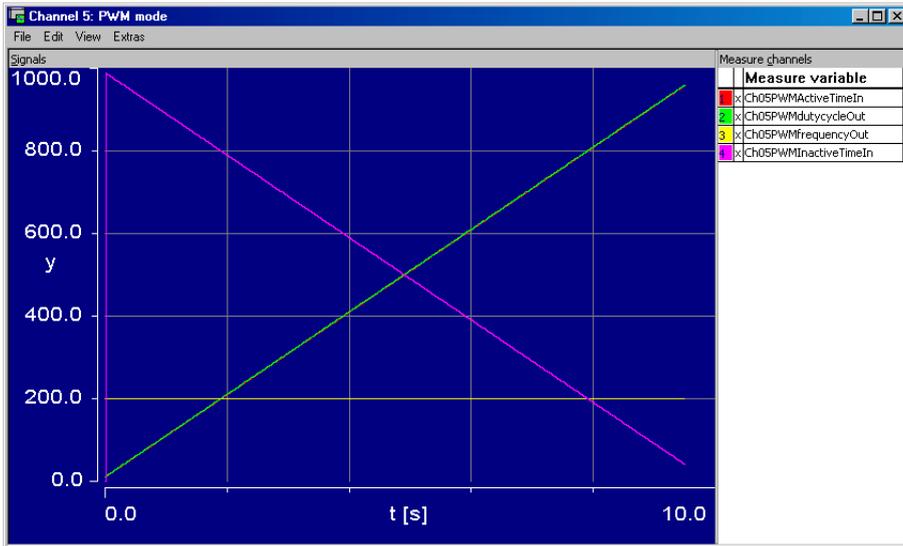
Digital I/O: Das Oszilloskop „Channel 3: Digital I/O“ zeigt die Messages Ch03DigIn und Ch03DigOut.



PWM: Das Oszilloskop „Channel 4: PWM mode static“ zeigt die Variable `DutycycleStaticCh04`, die das Tastverhältnis des PWM-Signals aufnimmt. Da aktive und inaktive Zeit fest vorgegeben sind, ist der Wert der Variablen konstant.

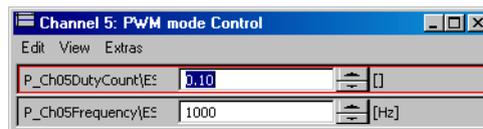


Das Oszilloskop „Channel 5: PWM mode“ zeigt die Eingangs-Messages Ch05PWMActiveTimeIn (rote Kurve) und Ch05PWMinactiveTimeIn (violette Kurve), die die aktive und inaktive Zeit des PWM-Signals beinhalten. Die Ausgangs-Messages Ch05PWMDutyCycleOut (grüne Kurve) und Ch05PWMMFrequencyOut (gelbe Kurve) angezeigt, die Frequenz (in Hz) und Tastverhältnis (in %) des PWM-Signals enthalten. Beim Start des Experiments wird die Y-Achse von Ch05PWMActiveTimeIn angezeigt.

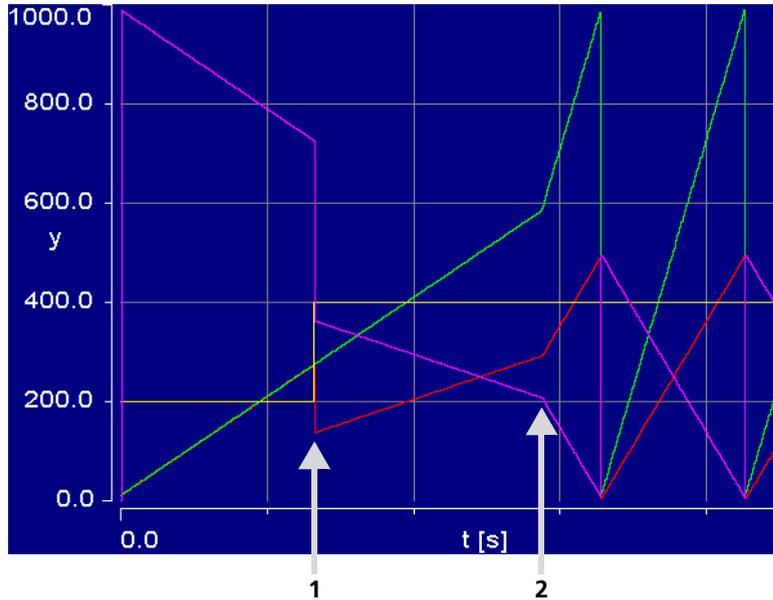


Wenn Sie den Parameter P_Ch05Frequency unten im Verstellfenster „Channel 5: PWM mode Control“ verstellen, ändert sich die Frequenz und damit auch die aktive und inaktive Zeit. Das Tastverhältnis bleibt gleich.

Wenn Sie den Parameter P_Ch05DutyCount oben im Verstellfenster „Channel 5: PWM mode Control“ verstellen, ändert sich der Verlauf des Tastverhältnisses und damit wieder die aktive und inaktive Zeit. Die Frequenz bleibt gleich.

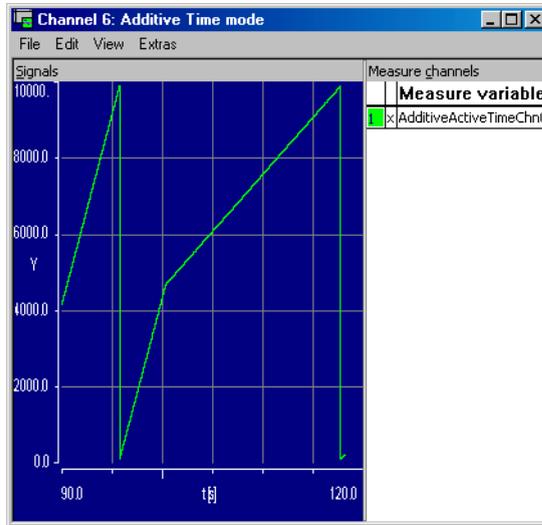


An der Stelle 1 in der folgenden Abbildung wurde die Frequenz von 1000 Hz auf 2000 Hz gesetzt, so dass die Kurven für Ch05PWMActiveTimeIn, Ch05PWMinactiveTimeIn und Ch05PWMfrequencyOut eine Sprungstelle zeigen. P_Ch05DutyCount wurde nicht geändert; die Kurve für Ch05PWMdutyCycleOut behält ihre Steigung.

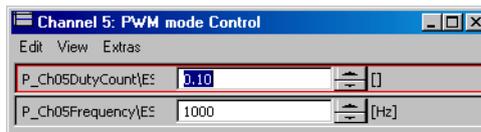


An der Stelle 2 wurde der Parameter P_Ch05DutyCount von 0,1 auf 0,5 gesetzt, so dass die Kurven für Ch05PWMdutyCycleOut, Ch05PWMActiveTimeIn und Ch05PWMinactiveTimeIn abknicken. An dieser Stelle bleibt die Frequenz unverändert.

Additive Time: Das Oszilloskop „Channel 6: Additive Time mode“ zeigt die Variable `AdditiveActiveTimeChn06`, die aufsummierte aktive Zeit des Signals aufnimmt.

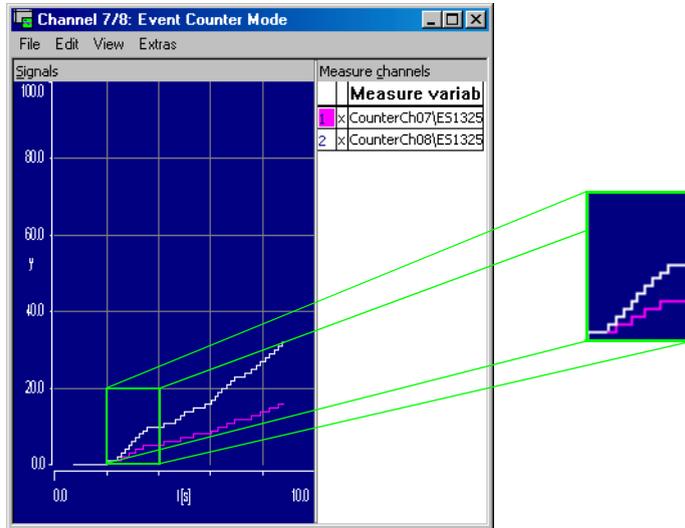


Sie können die Neigung des Sägezahns beeinflussen, indem Sie den Parameter `P_Ch05DutyCount` oben im Verstellfenster „Channel 5: PWM mode Control“ verstellen. Für den ersten Teil der abgebildeten Kurve war ein Wert von `0,1` eingestellt, für den zweiten Teil ein Wert von `0,01`.

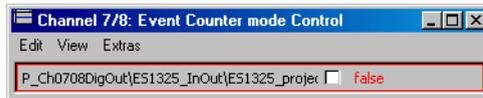


Das Signal `AdditiveActiveTimeChn06` ist unabhängig von der eingestellten Frequenz, da die aufsummierte aktive Zeit über die Tasklaufzeit davon unbeeinflusst bleibt.

Event Counter: Das Oszilloskop „Channel 7/8: Additive Time mode“ zeigt die Variablen CounterCh07 (untere Kurve) und CounterCh08 (obere Kurve) an, die aufsummierte aktive Zeit des Signals aufnimmt.

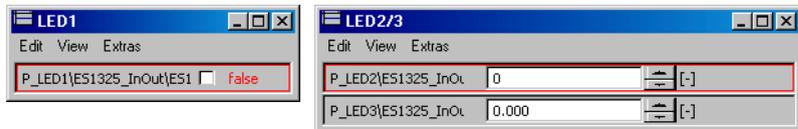


Sie können die beiden Variablen hochzählen, indem Sie den Parameter P_Ch0708DigOut im Verstellfenster „Channel 7/8: Event Counter mode Control“ abwechselnd auf true und false setzen.



CounterCh07 wächst halb so schnell wie CounterCh08, weil Kanal 8 jede Änderung zählt, während Kanal 7 nur Änderungen von false nach true zählt.

LED: In den Fenstern „LED1“ und „LED2/3“ steuern Sie über die Parameter P_LED1, P_LED2 und P_LED3 die LEDs in der Frontplatte der ES1325.



LED1 leuchtet, wenn `P_LED1` den Wert `true` hat, LED2 leuchtet, wenn `P_LED2` ungleich 0 ist, und LED3 leuchtet, wenn der Wert von `P_LED3` nach Abschneiden der Nachkommastellen ungleich null ist.

11.5 Tutorial – ES1325 (mit Trigger)

Das Beispiel zur ES1325 mit Verwendung des Triggers enthält ein Modell mit einem dynamischen PWM-Signal (Block `PWM`, s. Abb. 11-12), dessen Tastverhältnis von einem Sägezahnsignal bestimmt wird.

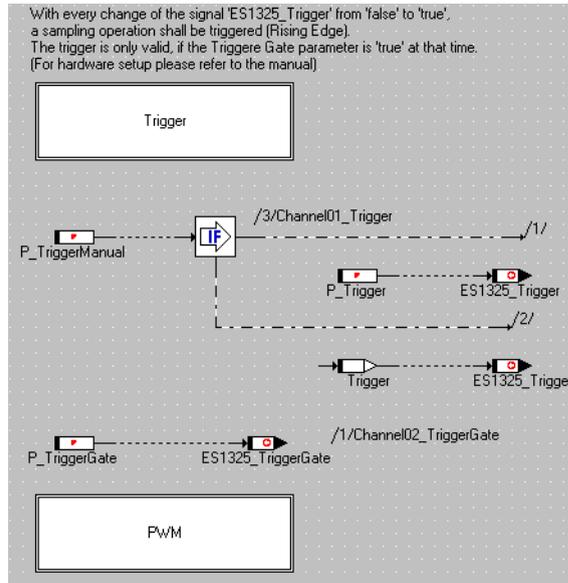


Abb. 11-11 ES1325 (mit Trigger) – Modell

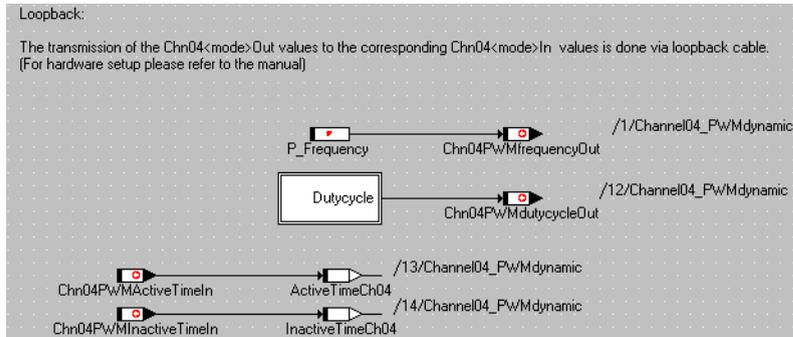


Abb. 11-12 ES1325 (mit Trigger) – PWM

Die Ausgangssignale (Messages Chn04PWMfrequencyOut und Chn04PWMdutyCycleOut) werden kontinuierlich versendet, aber das Lesen der Eingangssignale (Messages Chn04PWMActiveTimeIn und Chn04PWMInactiveTimeIn) wird von einem Hardware-Trigger gesteuert. Dieser wird wahlweise (Parameter P_TriggerManual) automatisch oder manuell ausgelöst.

Bei automatischer Triggerung wird das erste Triggersignal ES1325_Trigger im Block Trigger (Abb. 11-13) berechnet, bei manueller Auslösung wird das Triggersignal über den Parameter P_Trigger bestimmt.

Das zweite Triggersignal ES1325_TriggerGate wird immer über den Parameter P_TriggerGate bestimmt.

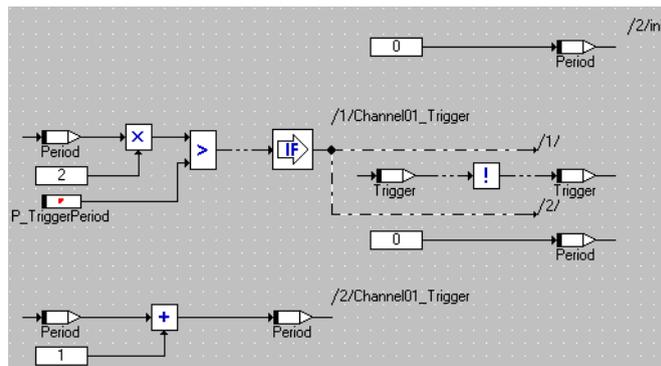


Abb. 11-13 ES1325 – Trigger

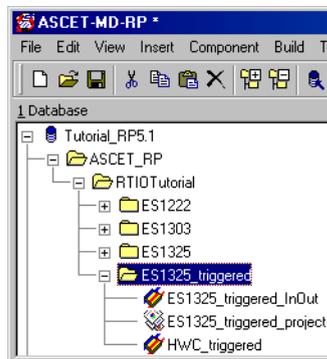
11.5.1 Die ES1325 Karte

Informationen über die Karte und die erforderliche Verkabelung finden Sie in Kapitel 11.4.1 „Die ES1325 Karte“ auf Seite 329.

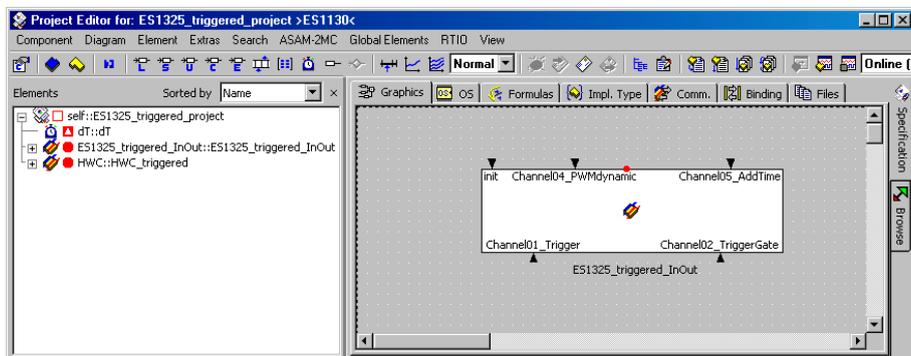
11.5.2 Beispielprojekt

Öffnen des Übungsbeispiels:

- Wählen Sie im Komponentenmanager den Ordner ASCET_RP\RTIOTutorial\ES1325_triggered.



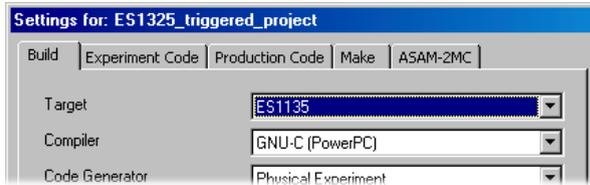
- Wählen Sie das Projekt ES1325_triggered_project.
- Öffnen Sie das Projekt.





- Klicken Sie auf die Schaltfläche **Specify Code Generation Options**.

Das Fenster „Settings for: ES1325_triggered_project“ öffnet sich.



- Wählen Sie im Register „Build“ die Optionen
Target: >ES1130< oder >ES1135<,
Compiler: GNU-C (PowerPC).

Hinweis

Nur Messages, die als „Exported“ deklariert sind, stehen für die RTIO-Kommunikation zur Verfügung.

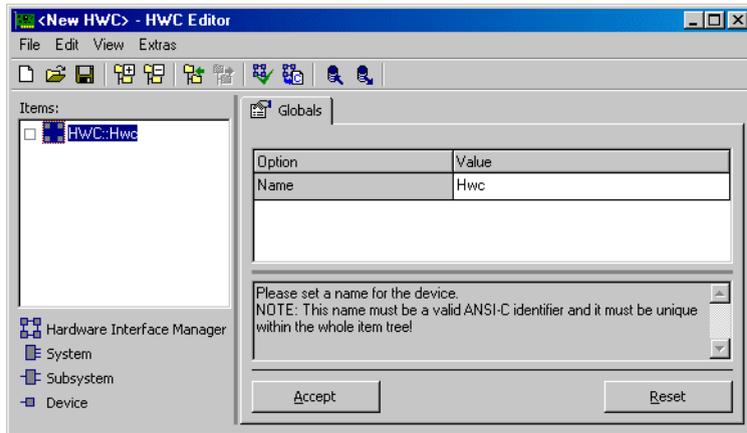
11.5.3 Erstellen der Hardware-Konfiguration

Hinweis

Normalerweise müssen Sie das C-Code-Modul HWC anlegen und in das Projekt einbinden, bevor Sie die Hardware-Konfiguration bearbeiten. Im Tutorial ist Ihnen dieser Schritt jedoch abgenommen.

Öffnen des HWC-Editors

- Wählen Sie im Projekteditor **RTIO** → **Open Editor**.
Der HWC-Editor wird geöffnet.



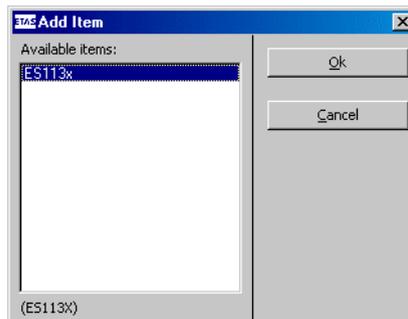
Hardware-Konfiguration (HWC) erstellen:

Die Hardware muss in der Items-Liste als baumartige Struktur beschrieben werden. Das Item `HWC` ist immer vorhanden und bildet die Wurzel des Baumes.

- Wählen Sie im HWC-Editor **Edit** → **Add Item** oder

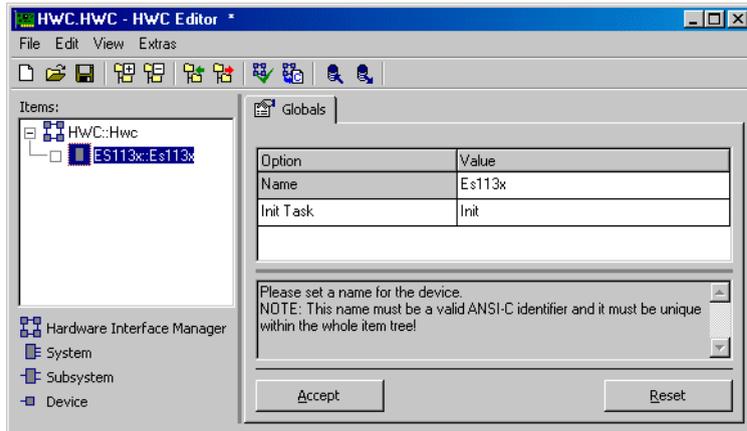


- klicken Sie auf die Schaltfläche **Add Item**. Das Fenster „Add Item“ wird angezeigt.



Über **Add Item** wird immer die Liste der verfügbaren Items der nächsten Hierarchiestufe geöffnet.

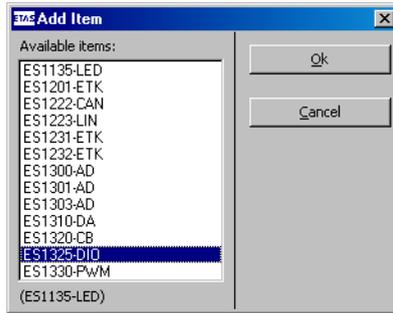
- Wählen Sie den Eintrag ES113x aus.
Dieser Eintrag dient der Beschreibung des ES1000.x-Systems mit integriertem ES1130 oder ES1135 PowerPC-Rechenknoten.
- Klicken Sie **OK**.
In der Liste „Items“ wird das Element ES113x hinzugefügt.



- Wählen Sie in der Liste „Items“ das Element ES113x.
Im Register „Globals“ wird für die Option `Init Task` der Task-Name `Init` vorgegeben. Im OS-Editor des Beispielprojekts gibt es eine gleichnamige `Init-Task`, infolgedessen muss hier keine andere `Init-Task` ausgewählt werden.

ES1325 einbinden und einrichten:

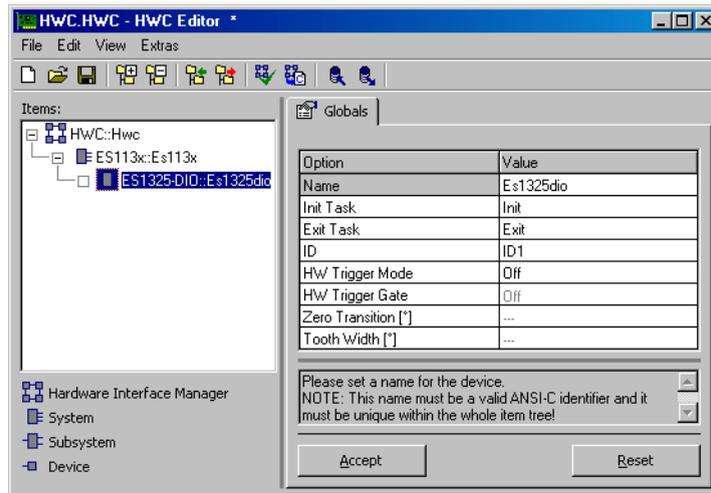
- Öffnen Sie über **Edit** → **Add Item** die Liste der verfügbaren Elemente der nächsten Hierarchiestufe.



- Wählen Sie den Eintrag **ES1325-DIO** aus. Dieser Eintrag dient der Beschreibung der ES1325-Schnittstellen.

- Klicken Sie **OK**.

In der Liste „Items“ wird das Element **ES1325-DIO** hinzugefügt.



- Wählen Sie in der Liste „Items“ das Element **ES1325-DIO**.

Für die Optionen `Init Task` und `Exit Task` können wie im Abschnitt „ES1325 einbinden und einrichten:“ auf Seite 335 die vorgegebenen Task-Namen `Init` und `Exit` verwendet werden. Die Nummer der Karte wird in der Option `ID` automatisch eingestellt. Da das Element **ES1325-DIO** das erste dieses Typs ist, ist der Wert „ID1“ voreingestellt. Sie müssen noch die Einstellungen für den Hardware-Trigger vornehmen.

- Doppelklicken Sie in der Zeile „HW Trigger Mode“ (siehe Seite 240) in die Spalte „Value“. Ein Kombikästchen öffnet sich.
- Wählen Sie den Eintrag `Rising edge`.
- Doppelklicken Sie in der Zeile „HW Trigger Gate“ (siehe Seite 242) in die Spalte „Value“. Ein Kombikästchen öffnet sich.
- Wählen Sie den Eintrag `On`.
Damit ist die Art des verwendeten Hardware-Triggers festgelegt.
- Übernehmen Sie die Einstellungen mit **Accept**.

HW Trigger Mode	Off
HW Trigger Gate	Off
Zero Transition [*]	Rising edge
Tooth Width [*]	Signal state
	Angle based

HW Trigger Gate	On
Zero Transition [*]	On
Tooth Width [*]	Off

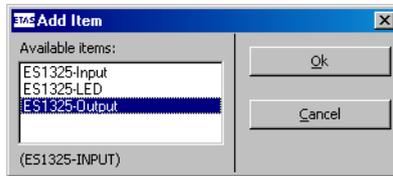
Das Register sollte jetzt so aussehen:

Option	Value
Name	Es1325dio
Init Task	Init
Exit Task	Exit
ID	ID1
HW Trigger Mode	Rising edge
HW Trigger Gate	On
Zero Transition [*]	...
Tooth Width [*]	...

Geräte anlegen:

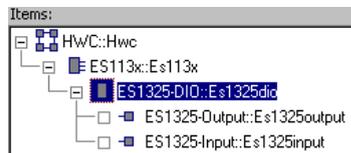
- Wählen Sie in der Liste „Items“ das Element **ES1325-DIO**.

- Öffnen Sie über **Edit** → **Add Item** die Liste der Elemente der nächsten Hierarchiestufe.



- Wählen Sie den Eintrag **ES1325-Output** und klicken Sie auf **OK**.
Dieser Eintrag dient der Beschreibung der Ausgangskanäle der ES1325.
- Fügen Sie das Gerät **ES1325-Input** hinzu.
ES1325-Input dient der Beschreibung der Eingangskanäle der ES1325.

Nun ist der Item-Baum für die Beschreibung des Beispielsystems komplett spezifiziert.



11.5.4 HWC-Einstellungen für die ES1325 vornehmen

Nun müssen die Aus- und Eingänge konfiguriert werden. Die Ausgänge liefern die Signale, die die Eingänge empfangen.

Ausgänge – Gerät ES1325-Output

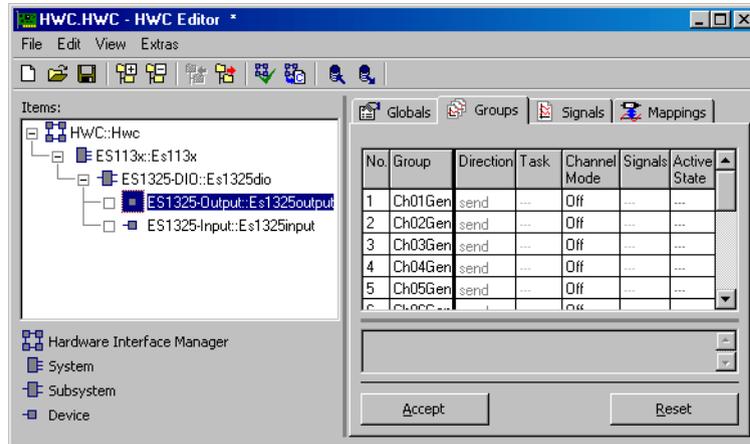
Register „Globals“: Im Register „Globals“ brauchen Sie keine Einstellungen vorzunehmen.

Register „Groups“: Im Register „Groups“ werden die signalgruppenspezifischen Einstellungen vorgenommen.

Das Device **ES1325-Output** hat für jedes Ausgangssignal eine vorgegebene Signalgruppe ($Ch<n>Gen$, $<n> = 01 - 16$). Sie stellen hier für jede verwendete Gruppe den Modus, die Taskzuordnung, die generierten Signale sowie die verwendeten Pegel und Flanken ein.

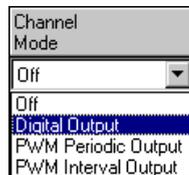
Einstellungen im Register „Groups“ vornehmen:

- Wählen Sie das Register „Groups“.



Nehmen Sie die folgenden Schritte für die Signalgruppen 1, 2 und 4 vor.

- Kanalmodus auswählen



- Wählen Sie in der Spalte „Channel Mode“ für die verschiedenen Gruppen folgende Modi:

Gruppe

Channel Mode

Ch01Gen, Ch02Gen Digital Output

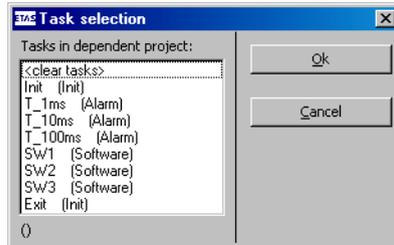
Ch04Gen PWM Periodic Output

In der Spalte „Active State“ wird der Standardwert für den jeweiligen Modus eingetragen.

In der Spalte „Task“ können Sie den verwendeten Gruppen nun eine Task zuordnen, in der die Signalübertragung stattfinden soll.

2. Zuordnen einer Task

- Doppelklicken Sie in die gewünschte Zeile, Spalte „Task“, um das Auswahlfenster „Task selection“ zu öffnen.



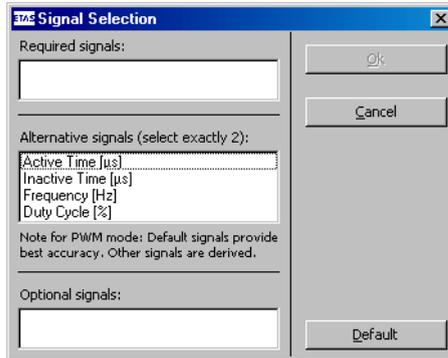
- Wählen Sie in der Task-Auswahlliste die folgenden Tasks, in denen die Datenübertragung erfolgen soll.

Gruppe	Task
Ch01Gen, Ch01Gen	T_1ms
Ch04Gen	T_10ms

- Klicken Sie **OK**.
Die ausgewählten Tasks werden in der Spalte „Task“ angezeigt.

3. Signale auswählen

- Doppelklicken Sie in der gewünschten Zeile; Spalte „Signals“, um das Fenster „Signal Selection“ zu öffnen.



Eine Beschreibung der Signale finden Sie im Abschnitt „Signals“ auf Seite 249.

- Wählen Sie folgende Signale aus.

Gruppe	Signals
Ch01Gen, Ch02Gen	Standardeinstellung übernehmen
Ch04Gen	Frequency [Hz], Duty Cycle [%]

- Bestätigen Sie die Auswahl mit **OK**.

4. Pegel auswählen

Da alle Gruppen die Standardeinstellung High verwenden, brauchen Sie in der Spalte „Active State“ keine Änderungen vornehmen.

- Wenn Sie alle nötigen Einstellungen vorgenommen haben, klicken Sie im HWC-Editor auf **Accept**, um die Einstellungen zu sichern.

Wenn Sie die Signalgruppen eingerichtet haben, sollte das Register so aussehen:

No.	Group	Direction	Task	Channel Mode	Use HW Trigger	Signals	Active State
1	Ch01Gen	send	T_1ms	Digital Output	No	[Select]	High
2	Ch02Gen	send	T_1ms	Digital Output	No	[Select]	High
3	Ch03Gen	send	---	Off	---	---	---
4	Ch04Gen	send	T_10ms	PWM Periodic Output	No	[Select]	High

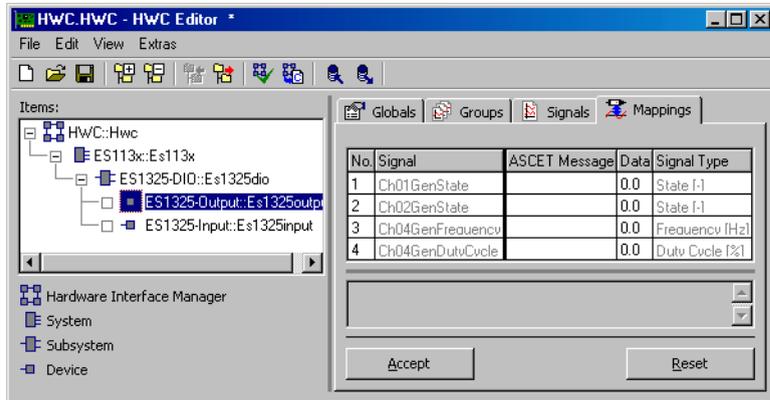
Register „Signals“: In diesem Register sind die Signale enthalten, die Sie im Register „Groups“ generiert haben.

No.	Signal	Formula	Signal Type
1	Ch01GenState	f(phys)=phys	State [1]
2	Ch02GenState	f(phys)=phys	State [1]
3	Ch04GenFrequency	f(phys)=phys	Frequenc [Hz]
4	Ch04GenDutyCycle	f(phys)=phys	Duty Cycle [%]

Sie können in diesem Register die Namen und die Formeln der Signale bearbeiten. Für das Beispiel sind jedoch *keine* Änderungen erforderlich.

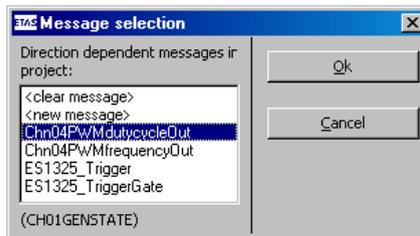
Register „Mappings“: In diesem Register werden die Signale und die ASCET-Messages aus dem Projekt einander zugeordnet. Durch Anklicken der gewünschten Zelle in der Spalte „ASCET Message“ wird ein Auswahldialog geöffnet. Dieser enthält nur Messages, die das Attribut *Exported* besitzen und der Übertragungsrichtung der Signalgruppe (hier: *send*) entsprechen; also:

- Direction = send → Send-Messages



ASCET-Message manuell zuordnen:

- Wählen Sie das Register „Mappings“.
 - Doppelklicken Sie in die gewünschte Zeile, Spalte „ASCET Message“.
- Das Fenster „Message selection“ öffnet sich.



Es enthält alle Send-Messages aus dem ASCET-Projekt.

- Wählen Sie folgende ASCET-Message für die Signale aus.

Signal	ASCET-Message
Ch01GenState	ES1325_Trigger

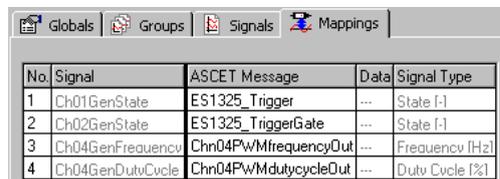
Signal

ASCET-Message

Ch02GenState	ES1325_TriggerGate
Ch04GenFrequency	Chn04PWMfrequencyOut
Ch04GenDutyCycle	Chn04PWMdutyCycleOut

- Klicken Sie auf **OK**.
- Klicken Sie abschließend im HWC-Editor auf **Accept**, um die Einstellungen zu sichern.

Wenn Sie allen Signalen Messages oder Werte zugewiesen haben, sollte das Register so aussehen:



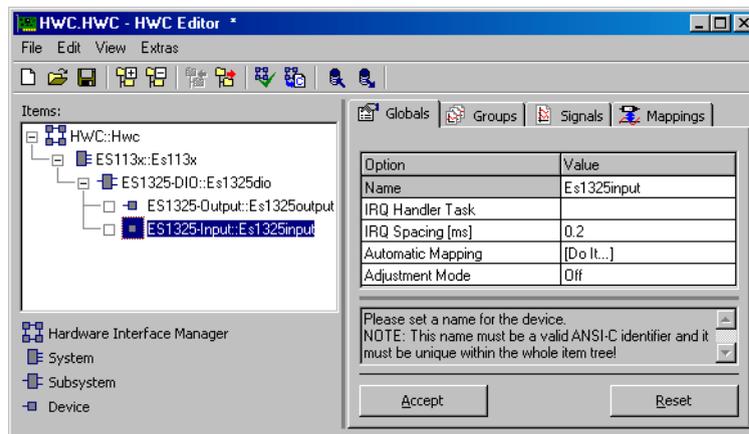
No.	Signal	ASCET Message	Data	Signal Type
1	Ch01GenState	ES1325_Trigger	---	State [-]
2	Ch02GenState	ES1325_TriggerGate	---	State [-]
3	Ch04GenFrequency	Chn04PWMfrequencyOut	---	Frequency [Hz]
4	Ch04GenDutyCycle	Chn04PWMdutyCycleOut	---	Duty Cycle [%]

Eingänge – Gerät ES1325-Input

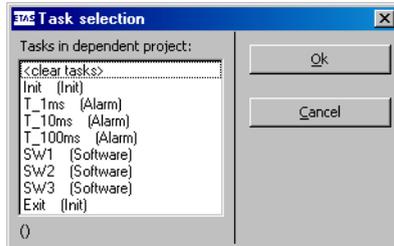
Register „Globals“: Im Register „Globals“ werden die Einstellungen für das Gerät vorgenommen.

Einstellungen im Register „Globals“ vornehmen:

- Wählen Sie das Register „Globals“.



- Doppelklicken Sie in das leere Feld neben der Option `IRQ Handler Task`.
Das Fenster „Task selection“ öffnet sich.



- Wählen Sie die Software-Task `sw1`.
Für die anderen Optionen können Sie die Standardwerte verwenden.
- Übernehmen Sie die Änderung mit **Accept**.

Das Register sollte jetzt so aussehen:

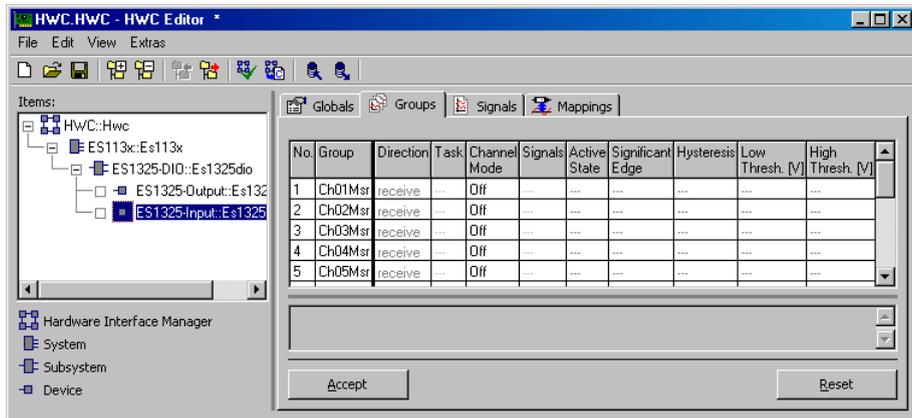
Option	Value
Name	Es1325input
IRQ Handler Task	SW1
IRQ Spacing [ms]	0.2
Automatic Mapping	[Do It...]
Adjustment Mode	Off

Register „Groups“: Im Register „Groups“ werden die signalgruppenspezifischen Einstellungen vorgenommen.

Das Device `ES1325-Input` hat für jedes Eingangssignal eine vorgegebene Signalgruppe (`Ch<n>MSr`, `<n> = 01 – 16`). Sie stellen hier für jede verwendete Gruppe den Modus, die Taskzuordnung, die generierten Signale sowie die verwendeten Pegel und Flanken ein.

Einstellungen im Register „Groups“ vornehmen:

- Wählen Sie das Register „Groups“.



Nehmen Sie die folgenden Schritte für die Signalgruppe 4 vor.

1. Kanalmodus auswählen

- Doppelklicken Sie in Zeile 4, Spalte „Channel Mode“.



Ein Kombikästchen öffnet sich.

- Wählen Sie für die Gruppe Ch04Msr den Modus PWM Input.

In den übrigen Spalten werden die Standardwerte für diesen Modus eingetragen.

2. Verwendung des Hardware-Triggers einstellen

- Doppelklicken Sie in Zeile 4, Spalte „Use HW Trigger“.



Ein Kombikästchen öffnet sich.

- Wählen Sie den Wert Yes.

- Doppelklicken Sie in Zeile 4, Spalte „IRQ“.

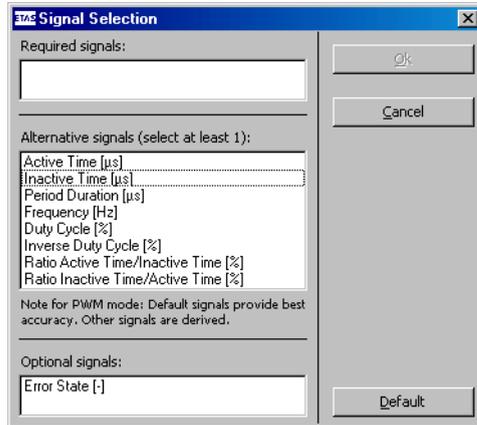
Ein Kombikästchen öffnet sich.

- Wählen Sie den Wert Yes.

Die geänderte Auswahl wird im Feld „IRQ“ angezeigt. Gleichzeitig wird das Feld „Task“ zurückgesetzt und gesperrt.

3. Signale auswählen

- Doppelklicken Sie in Zeile 4, Spalte „Signals“, um das Fenster „Signal Selection“ zu öffnen.



In diesem Fenster legen Sie fest, welche Signale für die Gruppe generiert werden. Eine Beschreibung der Signale finden Sie im Abschnitt „Signals“ auf Seite 249.

- Wählen Sie die Signale **Active Time [µs]** und **Inactive Time [µs]** aus.

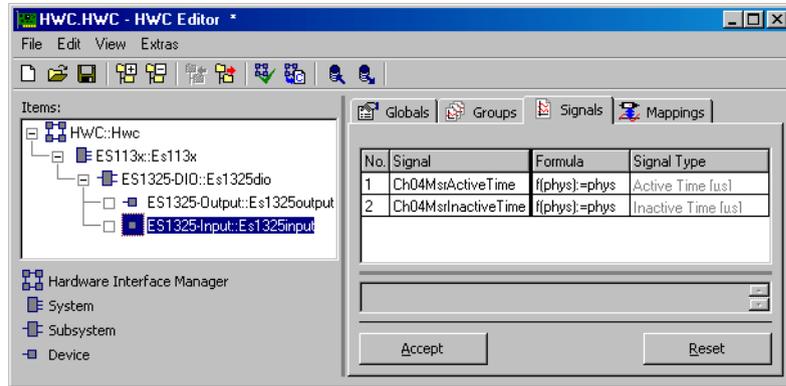
Weitere Einstellungen sind nicht erforderlich, da Sie in den Spalten „Active State“ und „Significant Edge“ die Standardwerte verwenden.

- Wenn Sie alle nötigen Einstellungen vorgenommen haben, klicken Sie im HWC-Editor auf **Accept**, um die Einstellungen zu sichern.

Das Register sollte jetzt so aussehen:

No.	Group	Direction	Task	Channel Mode	Use Hw Trigger	IRQ	Signals	Active State	Significant Edge	Hysteresis	Low Thresh. [V]	High Thresh. [V]
1	Ch01Msr	receive	...	Off
2	Ch02Msr	receive	...	Off
3	Ch03Msr	receive	...	Off
4	Ch04Msr	receive	...	PWM Input	Yes	Yes	[Select]	High	Inactive-Active	TTL	1.728	2.304
5	Ch05Msr	Off

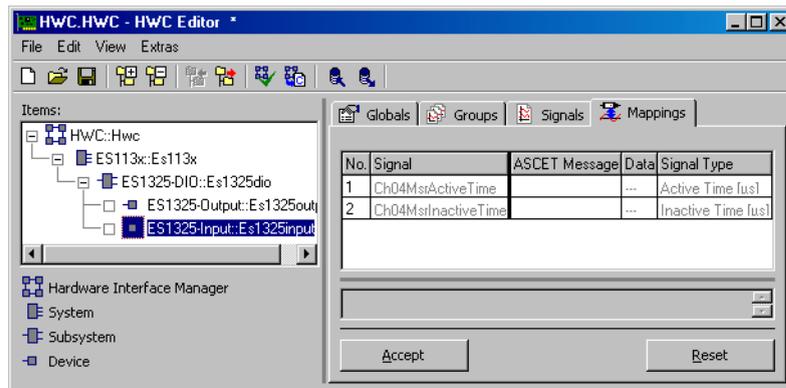
Register „Signals“: In diesem Register sind die Signale enthalten, die Sie im Register „Groups“ generiert haben.



Sie können in diesem Register die Namen und die Formeln der Signale bearbeiten. Für das Beispiel sind jedoch *keine* Änderungen erforderlich.

Register „Mappings“: In diesem Register werden die Signale und die ASCET-Messages aus dem Projekt einander zugeordnet. Durch Anklicken der gewünschten Zelle in der Spalte „ASCET Message“ wird ein Auswahldialog geöffnet. Dieser enthält nur Messages, die das Attribut *Exported* besitzen und der Übertragungsrichtung der Signalgruppe (hier: *receive*) entsprechen; also:

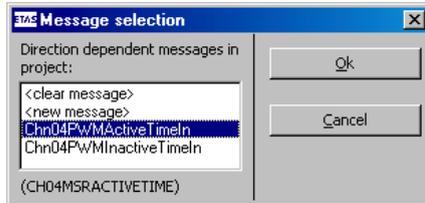
- Direction = receive → Receive-Messages



ASCET-Message manuell zuordnen:

- Wählen Sie das Register „Mappings“.

- Doppelklicken Sie in die gewünschte Zeile, Spalte „ASCET Message“.
- Das Fenster „Message selection“ öffnet sich.



Es enthält alle Receive-Messages aus dem ASCET Projekt.

- Wählen Sie folgende ASCET-Messages für die Signale aus.

Signal

Ch04MsrActiveTime

Ch04MsrInactiveTime

ASCET-Message

Chn04PWMActiveTimeIn

Chn04PWMIinactiveTimeIn

- Klicken Sie auf **OK**.
- Klicken Sie abschließend im HWC-Editor auf **Accept**, um die Einstellungen zu sichern.

Wenn Sie allen Signalen Messages zugewiesen haben, sollte das Register so aussehen:

No	Signal	ASCET Message	Data	Signal Type
1	Ch04MsrActiveTime	Chn04PWMActiveTimeIn	---	Active Time [us]
2	Ch04MsrInactiveTime	Chn04PWMIinactiveTimeIn	---	Inactive Time [us]

- Klicken Sie auf **OK**.
- Klicken Sie abschließend im HWC-Editor auf **Accept**, um die Einstellungen zu sichern.

11.5.5 Hardware-Konfiguration speichern

Die erstellte Hardware-Konfiguration kann im File-Container des Projekts oder als DOS-Datei (Erweiterung * .HXX) gespeichert werden. Wie das geht, ist in Kapitel 11.2.5 auf Seite 304 beschrieben.

Hinweis

Die Lösung des Beispiels ist unter dem Namen ES1325 .hwx abgelegt. Achten Sie darauf, diese nicht zu überschreiben.

11.5.6 Code für HWC-Modul erzeugen

Anschließend muss die Generierungssequenz für das HWC-Modul gestartet werden. Wie das geht, ist in Kapitel 11.2.6 auf Seite 306 beschrieben.

11.5.7 Experimentieren mit dem Beispielprojekt

Alle RTIO-spezifischen Aktionen sind nun beendet, somit kann der HWC-Editor geschlossen werden. Als weiterer Arbeitsgang schließt sich nun noch die normale Codegenerierung für das Experimentaltarget an, die aus dem Projekteditor gestartet wird.

Hinweis

Es empfiehlt sich, das HWC-Modul nicht in die grafische Darstellung des Projekteditors mit aufzunehmen, da dieses Modul durch jeden RTIO-Generierungsprozess verändert wird. Man erhält eine ungünstige grafische Darstellung des HWC-Moduls.

Codegenerierung für das Experimentaltarget:

- Wählen Sie im Projekteditor **Component** → **Build**, um den Code für das gesamte Projekt zu erzeugen.
- Wählen Sie **Component** → **View Generated Code**, um den generierten Code zu betrachten.

Online experimentieren:

- Falls noch nicht geschehen, verbinden Sie die Ein- und Ausgänge der ES1325 gemäß Abschnitt „Anschlüsse“ auf Seite 330 und schalten Sie die Stromversorgung der ES1000.x ein.

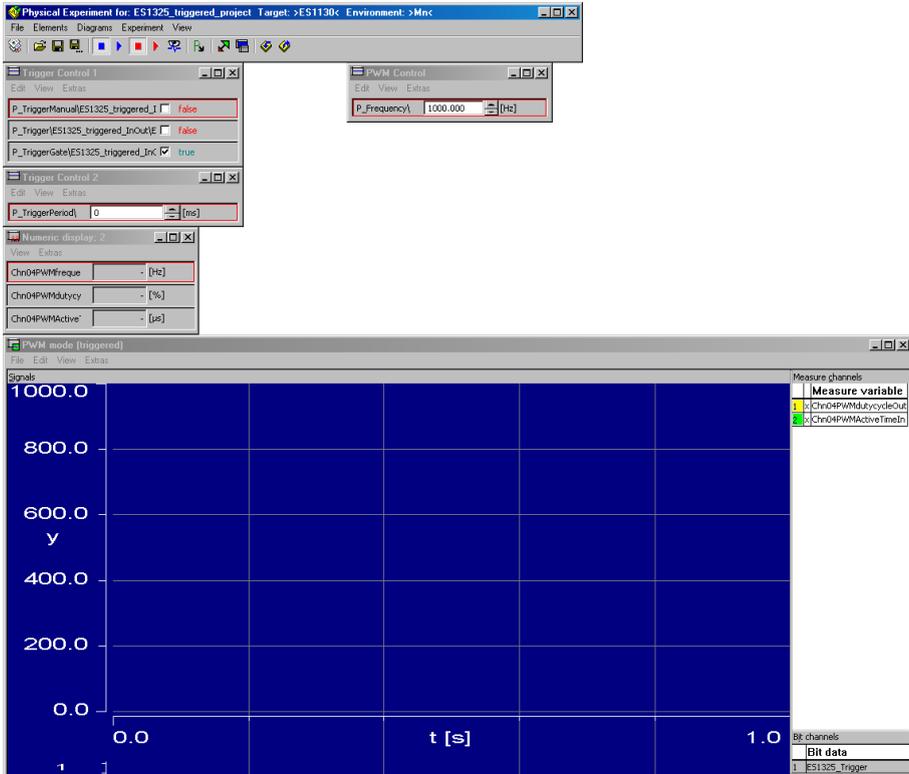


- Wählen Sie im Projekteditor aus dem Kombikästchen „Experiment Target“ den Eintrag `Online (RP)`.

Der Eintrag `Offline (RP)` ist für Offline-Experimente auf dem Target gedacht.

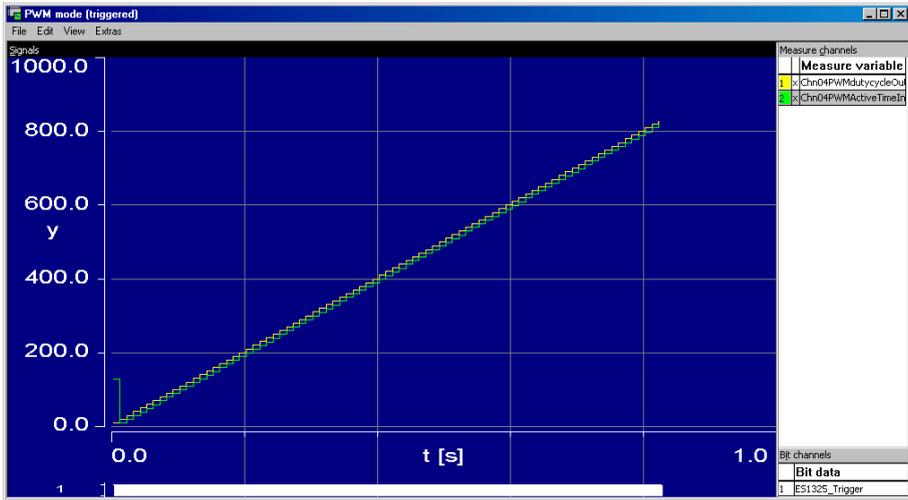
- Wählen Sie **Component** → **Open Experiment**.

Das Fenster „Physical Experiment“ sowie die vordefinierte Experimentierumgebung bestehend aus einem Oszilloskop, einer numerischen Anzeige und drei Verstellfenstern öffnen sich.



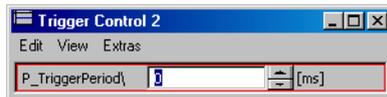
Die Standardeinstellung geht von einer automatischen Berechnung des des ersten Triggersignals und einem konstanten zweiten Triggersignal aus (Parameter `P_TriggerManual` ist `false`, Parameter `P_TriggerGate` ist `true`).

Das Oszilloskop „PWM mode (triggered)“ zeigt die Ausgangs-Messung `Chn04PWMDutyCycleOut` (gelbe Kurve) und die Eingangs-Messung `Chn04PWMActiveTimeIn` (grüne Kurve) sowie im unteren Teil die Messung `ES1325_Trigger`, die das erste Triggersignal beinhaltet.

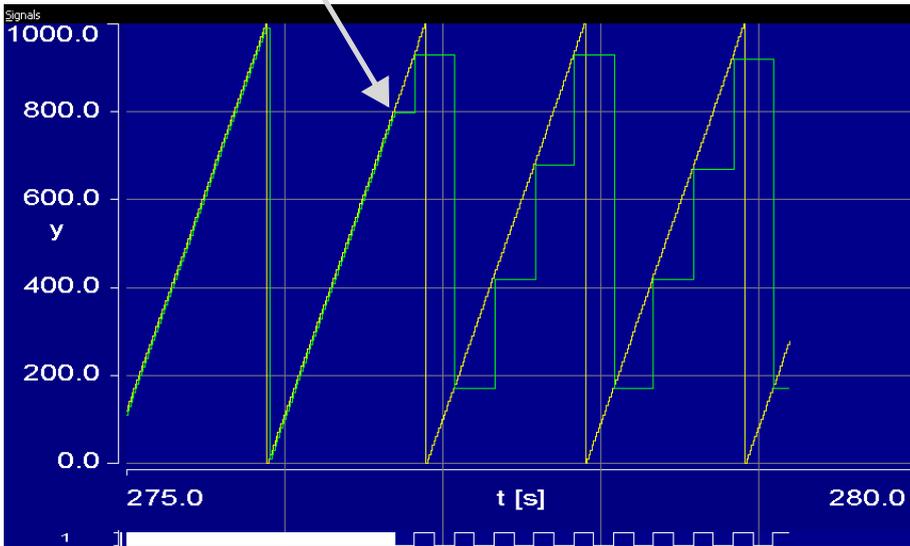


Wenn Sie die Messung mit den vorgegebenen Werten starten, erhalten Sie zwei um einen Schritt versetzte Kurven.

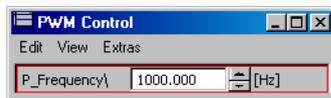
Wenn Sie im Verstellfenster „Trigger Control 2“ den Wert des Parameters `P_TriggerPeriod` erhöhen, verlängert sich die Zeit zwischen zwei steigenden Flanken. Entsprechend seltener wird `Chn04PWMActiveTimeIn` eingelesen.



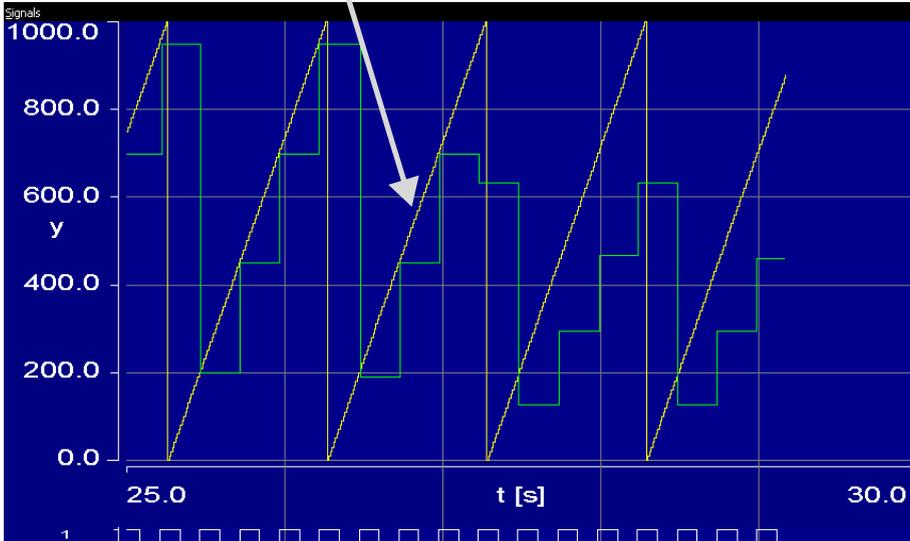
An der markierten Stelle der folgenden Abbildung wurde der Parameter `P_TriggerPeriod` von 0 ms auf 250 ms gesetzt. Das Triggersignal unten im Anzeigebereich des Oszilloskops liefert nur noch alle 250 ms eine steigende Flanke, entsprechend selten wird die Message `Chn04PWMdutyCycleOutIn` eingelesen, und die zugehörige Kurve hat große Stufen.



Wenn Sie den Parameter `P_Frequency` im Verstellfenster „PWM Control“ verstellen, ändert sich die Frequenz und damit auch die aktive und inaktive Zeit. Das Tastverhältnis bleibt gleich.

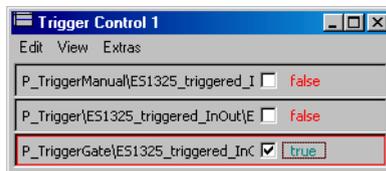


An der markierten Stelle der folgenden Abbildung wurde die Frequenz von 1000 Hz auf 1500 Hz gesetzt. Die Kurve für Chn04PWMDutyCycleOut bleibt unverändert, aber die Kurve für Chn04PWMActiveTimeIn hat nun ein niedrigeres Maximum.

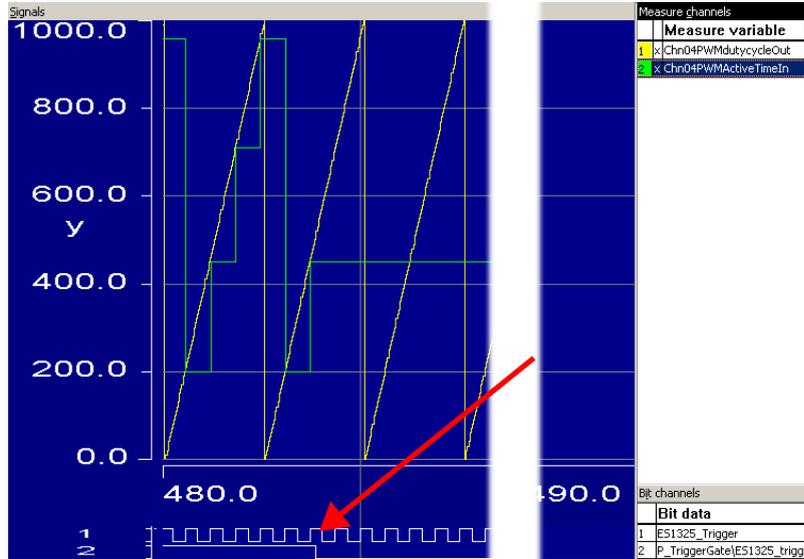


Für diese Abbildung wurde der Bereich der X-Achse im Fenster „Display setup“, Feld „Extent“ (zu erreichen aus dem Oszilloskop mit [Extras → Setup](#)), auf 5 Sekunden gesetzt.

Im Verstellfenster „Trigger Control 1“ können Sie die Triggersteuerung beeinflussen.



Wenn Sie z.B. den Parameter `P_TriggerGate` auf `false` setzen, wird `Chn04PWMActiveTimeIn` gar nicht mehr eingelesen, da das zweite Triggersignal `false` ist. Die Kurve bleibt auf dem letzten Wert. Die Kurve für `Chn04PWMdutyCycleOut` bleibt wieder unverändert.

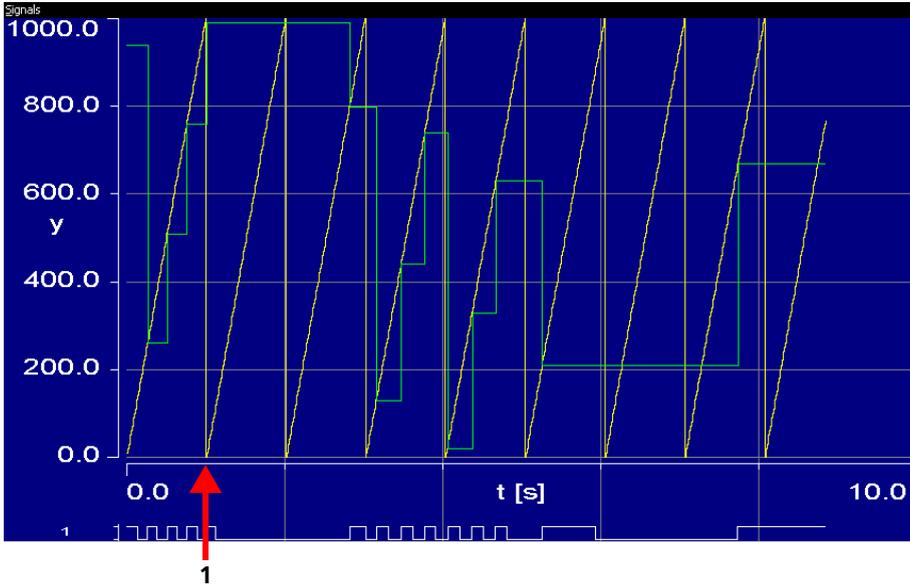


Für diese Abbildung wurde der Bereich der X-Achse auf 10 Sekunden gesetzt und die Variable `P_TriggerGate` ins Oszilloskop eingefügt.

Schalten Sie `P_TriggerGate` wieder auf `true`, um die Messung von `Chn04PWMActiveTimeIn` fortzusetzen.

Wenn Sie den Parameter `P_TriggerManual` im Fenster „Trigger Control 1“ auf `true` setzen, aktivieren Sie die manuelle Steuerung des ersten Triggersignals. Jedes Mal, wenn Sie nun den Parameter `P_Trigger` von `false` auf `true` umschalten, wird ein Triggersignal erzeugt und die Message `Chn04PWMActiveTimeIn` eingelesen.

In der folgenden Abbildung wurde an der Stelle 1 die manuelle Triggerung aktiviert. Am Verlauf des Signals `ES1325_Trigger` läßt sich danach erkennen, wie `P_Trigger` verstellt wurde. Mit jeder steigenden Flanke des Triggersignals wurde `Chn04PWMActiveTimeIn` eingelesen.



Für diese Abbildung wurde der Bereich der X-Achse im Fenster „Display setup“, Feld „Extent“ (zu erreichen aus dem Oszilloskop mit **Extras** → **Setup**), auf 10 Sekunden gesetzt.

12 ETAS Network Manager

In diesem Abschnitt werden die Konfiguration des ETAS-Netzwerks mit Hilfe des ETAS Network Managers und die dazugehörige Netzwerktopologie beschrieben.

12.1 Übersicht

ASCET-RP unterstützt verschiedene Konfigurationen für den Hardware-Zugang über Ethernet:

- Verwendung mehrerer Netzwerkadapter:
 - ein Netzwerkadapter für das Firmen-Netzwerk und
 - ein Netzwerkadapter für die ETAS-Hardware oder
- Verwendung eines Netzwerkadapters:
 - automatisches Umschalten zwischen Firmen-Netzwerk und der ETAS-Hardware.

Hinweis

Sie benötigen mit ASCET-RP V5.4 keinen separaten Netzwerkadapter mehr, um die ETAS-Hardware an Ihren PC anzuschließen. Sie können ein und denselben Netzwerkadapter sowohl für das Firmen-Netzwerk als auch für das ETAS-Netzwerk verwenden.

Bei der Auswahl des Netzwerkadapters für die ETAS-Hardware werden Sie vom ETAS Network Manager unterstützt.

Der ETAS Network Manager gibt Ihnen einen Überblick über die verfügbaren Netzwerkadapter Ihres PCs sowie der Art der IP-Adress-Zuweisung. Ist mehr als ein Netzwerkadapter im System vorhanden, können Sie hier den Netzwerkadapter auswählen, mit dem Sie die ETAS-Hardware an Ihren PC anschließen. Außerdem können Sie den Adressbereich für die IP-Zuweisung der ETAS-Hardware festlegen.

Für die Auswahl des Netzwerkadapters und die Konfiguration der Netzwerkumgebung für die ETAS-Hardware benötigen Sie keine Administratorrechte. Sie können zwischen dem ETAS-Netzwerk und dem Firmen-Netzwerk wechseln, ohne den PC neu zu starten.

Hinweis

Mit dem Network Manager können Sie keine Konfiguration oder Konfigurationsänderung des Netzwerkadapters durchführen. Hierfür sind Administratorrechte erforderlich (siehe Dokumentation Ihres verwendeten Betriebssystems).

12.2 ETAS-Hardware-Adressierung

Über das ETAS-Netzwerk können Sie mehrere Geräte (auch gleichen Typs) an Ihren PC anschließen. Die angeschlossenen Geräte werden im lokalen ETAS-Netzwerk durch ihre einmalige IP-Adresse identifiziert.

Ein in ASCET-RP integrierter IP-Manager vergibt die IP-Adressen aus einem Adressen-Pool an die angeschlossenen Geräte.

Den Adressbereich des Adressen-Pools können Sie im ETAS Network Manager festlegen.

12.3 Netzwerkkadapter-Adressierung

12.3.1 Art der Netzwerkkadapter-Adressierung

Die Art der Netzwerkkadapter-Adressierung im Firmen-Netzwerk ist vom verwendeten Betriebssystem und der Konfiguration des Netzwerkkadapters abhängig:

Betriebssystem	Art der Netzwerkkadapter-Adressierung			
	manuell	DHCP	DHCP+APIPA	DHCP+ alternative IP-Adresse
Windows NT	ja	ja	nein	nein
Windows 98 SE	ja	ja	ja	nein
Windows 2000	ja	ja	ja	nein
Windows XP	ja	ja	ja	ja

Das ETAS-Netzwerk unterstützt folgende Netzwerkkadapter-Adressierungen:

Betriebssystem	Art der Netzwerkkadapter-Adressierung			
	manuell	DHCP	DHCP+APIPA	DHCP+ alternative IP-Adresse
Windows NT	ja	ja	nein	nein
Windows 98 SE	ja	nein	ja	nein
Windows 2000	ja	nein	ja	nein
Windows XP	ja	nein	ja	ja

Wollen Sie die Netzwerkkadapter sowohl für das Firmen-Netzwerk als auch für das ETAS-Netzwerk verwenden, können Sie die Netzwerkkadapter, die ausschließlich die Adressierung DHCP unterstützen, nicht für den Doppelbetrieb verwenden (Ausnahme: Windows NT). DHCP kann nur in Kombination mit APIPA oder alternativer IP-Adresse verwendet werden.

12.3.2 Manuelle Adressierung der Netzwerkadapter

Das Adressieren eines Netzwerkadapters ist abhängig vom Betriebssystem.

Die Anleitung für die Adressierung des Netzwerkadapters Ihres PCs können Sie der Dokumentation Ihres Betriebssystems entnehmen.

Um den Netzwerkadapter manuell zu adressieren, benötigen Sie Administratorrechte. Wenden Sie sich gegebenenfalls an Ihren Systemadministrator.

Ist der Netzwerkadapter manuell adressiert, d.h., besitzt er eine statische IP-Adresse, kann der Fall eintreten, dass Sie versehentlich nach ETAS-Hardware suchen oder ETAS-Hardware initialisieren wollen, obwohl der PC mit dem Firmen-Netzwerk verbunden ist. Sie können im Network Manager festlegen, dass für diesen Fall eine Warnung angezeigt wird, bevor eine IP-Adresse an eine ETAS-Hardware vergeben wird.

12.3.3 Adressierung des Netzwerkadapters über DHCP

Die Adressierung über DHCP setzt voraus, dass der DHCP-Server verfügbar ist. Sollte der DHCP-Server nicht verfügbar sein oder gibt es keinen DHCP-Server (wie im ETAS-Netzwerk), ist der Netzwerkadapter nicht konfiguriert.

Für diesen Fall hat jedes Betriebssystem (Ausnahme: Windows NT) einen Automatismus, damit der Netzwerkadapter automatisch eine IP-Adresse erhält:

Windows 98 SE

Windows 98 SE verwendet automatisch die DHCP-Adresse im Firmen-Netzwerk und die APIPA-Adresse im ETAS-Netzwerk. Sofern der PC am DHCP-Netz gebootet wurde, wird die Umkonfiguration auf APIPA beim Start der ETAS-Software (z.B. ASCET-RP) angestoßen. Diese dauert mindestens 60 Sekunden. Wird der PC nach Benutzung der ETAS-Software wieder an das DHCP-Netz gesteckt, erfolgt eine Umkonfiguration des Netzwerkadapters auf eine DHCP-Adresse. Diese Umkonfiguration kann eventuell erst nach 5 Minuten abgeschlossen sein. Um diese Zeit zu verringern, hat der Benutzer die Möglichkeit, in einem MS-DOS-Eingabefenster mit dem Befehl **ipconfig /renew** die Umkonfiguration manuell aufzurufen.

Eine Netzwerkadapter-Adressierung über DHCP ohne APIPA-Adresse wird nicht unterstützt.

Windows 2000

Windows 2000 prüft automatisch, ob eine Verbindung zum DHCP-Server besteht. Ist dies nicht der Fall, so wird die IP-Adresse automatisch über APIPA vergeben. Im ETAS-Netzwerk wird immer die APIPA-Adresse verwendet. Beim Umstecken zwischen DHCP-Netz und ETAS-Hardware sollte darauf geachtet werden, dass ein Verbindungsausfall durch das Betriebssystem detektiert wird, da nur dann eine Umkonfiguration angestoßen wird. Dies dauert bis zu 10

Sekunden. Für die eigentliche Umkonfiguration von einer DHCP-Adresse auf eine APIPA benötigt das Betriebssystem 60 Sekunden. Wird der Netzwerkadapter wieder an das DHCP-Netz angeschlossen, findet die Konfiguration auf eine DHCP-Adresse unmittelbar nach Verbindungsdetektion statt.

Eine Netzwerkadapter-Adressierung über DHCP ohne APIPA-Adresse wird nicht unterstützt.

Windows XP

Windows XP prüft automatisch, ob eine Verbindung zum DHCP-Server besteht. Ist dies nicht der Fall, so wird die IP-Adresse automatisch entweder über APIPA vergeben oder es wird die benutzerspezifische alternative IP-Adresse verwendet. Im ETAS-Netzwerk wird immer entweder die APIPA-Adresse oder die alternative IP-Adresse verwendet. Beim Umstecken zwischen DHCP-Netz und ETAS-Hardware sollte darauf geachtet werden, dass ein Verbindungsausfall durch das Betriebssystem detektiert wird, da nur dann eine Umkonfiguration angestoßen wird. Dies dauert bis zu 10 Sekunden. Für die eigentliche Umkonfiguration von einer DHCP-Adresse auf eine APIPA oder die alternative benötigt das Betriebssystem 60 Sekunden. Wird der Netzwerkadapter wieder an das DHCP-Netz angeschlossen, findet die Konfiguration auf eine DHCP-Adresse unmittelbar nach Verbindungsdetektion statt.

Eine Netzwerkadapter-Adressierung über DHCP ohne alternative Adressierung wird nicht unterstützt.

Windows NT

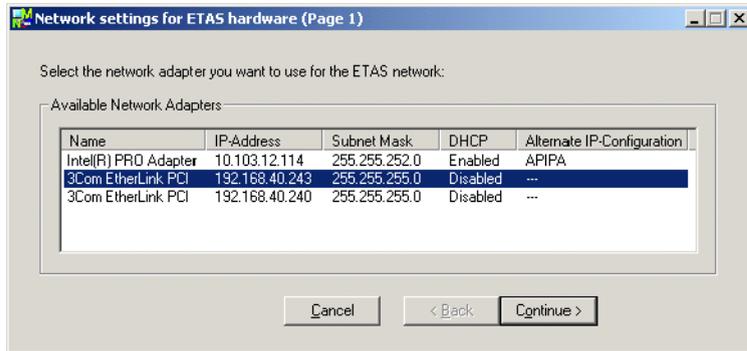
In Windows NT wird die DHCP-Adresse für eine bestimmte Zeit, die sogenannte Lease-Dauer, vergeben. Nach Ablauf dieser Zeit ist der Netzwerkadapter nicht mehr konfiguriert. Im Network Manager ist es möglich, diese Lease-Dauer zu verlängern, so dass Sie die IP-Adresse des Netzwerkadapters verwenden können, auch wenn Sie längere Zeit nicht im Firmen-Netzwerk arbeiten. Die Lease-Zeit wird erst mit dem ersten Hardware-Zugriff durch die ETAS-Software (z.B. ASCET-RP) angestoßen. Die Erhöhung der Leasezeit dauert ca. 20 Sekunden. Beim Beenden der ETAS-Software wird der Lease auf den ursprünglichen Wert zurückgesetzt.

Hinweis

Die Veränderung der Leasezeit ist nur dann sinnvoll, wenn längere Zeit (größer als die existierende Leasezeit) nicht am DHCP-Netzwerk gearbeitet wird. Um IP-Adreßkonflikte ausschließen zu können, sollte der PC erst 20 Sekunden nach dem Beenden der ETAS-Software an das DHCP-Netzwerk angeschlossen werden.

12.4 Bedienoberfläche

12.4.1 Dialogfenster „Network settings for ETAS hardware (Page 1)“

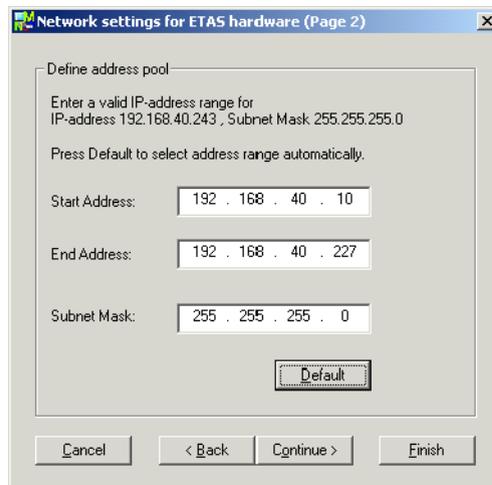


Folgende Informationen über die verfügbaren Netzwerkadapter werden angezeigt:

- **Name**
Name des Netzwerkadapters. Der Eintrag ist in diesem Fenster nicht editierbar.
- **IP-Adress**
IP-Adresse des Netzwerkadapters. Der Eintrag ist in diesem Fenster nicht editierbar.
- **Subnet Mask**
Einstellung für die Subnetzmaske. Der Eintrag ist in diesem Fenster nicht editierbar.
- **DHCP**
Zeigt an, ob der Netzwerkadapter für DHCP konfiguriert ist:
 - Enabled
Der Netzwerkadapter ist für DHCP konfiguriert.
 - Disabled
Der Netzwerkadapter ist mit einer festen IP-Adresse konfiguriert.
- **Alternate IP Configuration**
Zeigt die alternative IP-Adressierung des Netzwerkadapters, wenn dieser für DHCP konfiguriert ist. Die Anzeige ist abhängig vom verwendeten Betriebssystem.

- APIPA
Automatic Private IP Addressing: Verfahren zur Automatisierung der IP-Konfiguration von Netzwerkverbindungen
- ---
Eine alternative IP-Adresse ist nicht vorhanden oder deren Verwendung wird vom Betriebssystem nicht unterstützt (Windows NT).
- User Defined
Eine benutzerspezifische alternative IP-Adresse kann vom Anwender definiert werden (Windows XP).

12.4.2 Dialogfenster „Network settings for ETAS hardware (Page 2)“



Grundsätzlich lassen sich die Werte durch direktes Editieren bzw. durch das Setzen auf die Standardeinstellung über ein Dialogfenster verändern.

Folgende Netzwerkparameter können eingestellt werden:

- **Start Address**
Erste IP-Adresse des IP-Adressbereichs der ETAS-Hardware.
- **End Address**
Letzte IP-Adresse des IP-Adressbereichs der ETAS-Hardware.
- **Subnet Mask**
Zugehörige Subnetzmaske.

Reservierung von IP-Adressen

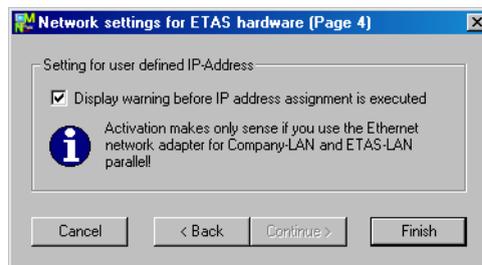
Im bisher für ETAS-Hardware benutzten IP-Adressbereich (192.168.40.1 – 192.168.40.254 mit der Subnetzmaske 255.255.255.0) sind die folgenden IP-Adressen für bestimmte ETAS-Hardware reserviert:

IP-Adresse	ETAS-Hardware
192.168.40.10	ES1120
192.168.40.11	ES1130
192.168.40.12	ES780
192.168.40.13	reserviert
192.168.40.14	LABCAR-RTPC
192.168.40.15	ES1135

Diese Adressen werden ausschließlich an diese Geräte vergeben und können somit nicht für andere ETAS-Hardware verwendet werden. Das ist bei der Definition des Adressbereichs unbedingt zu berücksichtigen.

12.4.3 Dialogfenster „Network settings for ETAS Hardware (Page 4)“

Dieses Dialogfenster wird Ihnen nur angezeigt, wenn der ausgewählte Netzwerkadapter manuell adressiert ist.



Folgende Parameter können eingestellt werden:

- **Display warning before IP address assignment is executed**

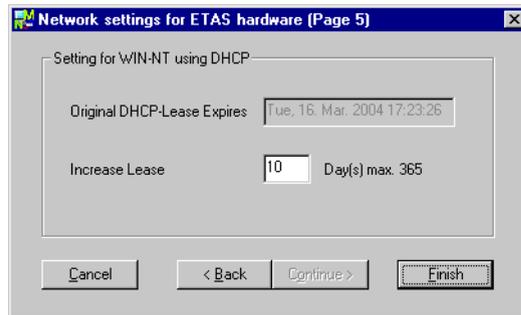
Mit der Option wird festgelegt, dass eine Warnung angezeigt wird, bevor eine IP-Adresse an ETAS-Hardware zugewiesen wird.

Hinweis

Die Aktivierung dieser Warnung ist nur sinnvoll, wenn über diesen Netzwerkkadpter wahlweise der PC an einem Firmennetzwerk oder ETAS-Messmodule im ETAS-Netzwerk betrieben werden sollen.

12.4.4 Dialogfenster „Network settings for ETAS hardware (Page 5)“

Dieses Fenster wird nur angezeigt, wenn Sie Windows NT als Betriebssystem verwenden und der ausgewählte Netzwerkkadpter für DHCP konfiguriert ist.



Folgende Informationen über die verfügbaren Netzwerkkadpter werden angezeigt:

- **Original DHCP-Lease Expires**
Zeigt an, wann die DHCP-Lease-Dauer abläuft. Dieses Feld ist nicht editierbar.
- **Increase Lease**
Hier können Sie die Anzahl der Tage festlegen, um die die Lease-Dauer verlängert werden soll. Sie können einen ganzzahligen positiven Wert bis 365 zuweisen.

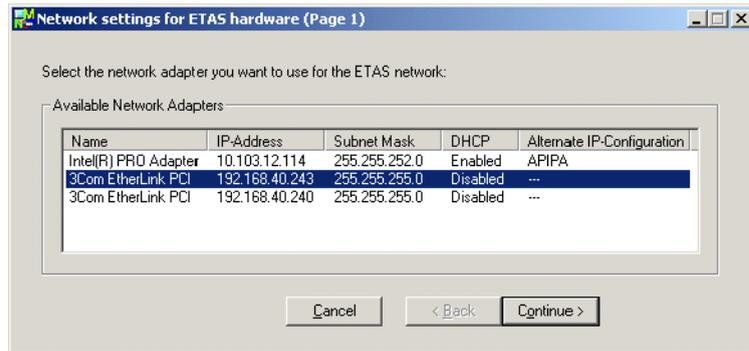
12.5 Netzwerkkadpter für ETAS-Hardware konfigurieren

Die ersten Schritte sind für die Konfiguration von Netzwerkkadptern mit fester IP-Adresse und von Netzwerkkadptern in einer DHCP-Umgebung gleich.

Network Manager starten:

- Wählen Sie im Windows-Startmenü **Start** → **Programme** → **ETAS** → **ASCET 5.1** → **ETAS Network settings**.

Das Dialogfenster „Network settings for ETAS hardware (Page 1)“ wird geöffnet.



Netzwerkadapter auswählen:

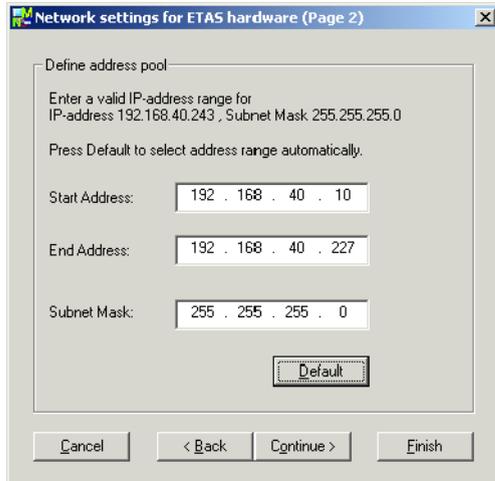
- Wählen Sie im Feld „Available Network Adapters“ den Netzwerkadapter aus, die Sie für das Firmen- und das ETAS-Netzwerk verwenden wollen.

Sie können nur Netzwerkadapter auswählen, deren Adressierungsart vom ETAS-Netzwerk unterstützt wird.

Mit **Cancel** können Sie den Vorgang abbrechen.

- Klicken Sie auf **Continue**, um den ausgewählten Netzwerkadapter zu konfigurieren.

Das Dialogfenster „Network settings for ETAS hardware (Page 2)“ wird geöffnet.



Hinweis

Die Schaltfläche **Finish** ist nur bei der Konfiguration von Netzwerkadaptern in einer DHCP-Umgebung vorhanden.

Netzwerkadapter mit fester IP-Adresse konfigurieren

Adressenpool festlegen:

- Klicken Sie im Fenster „Network settings for ETAS hardware (Page 2)“ auf den Eintrag („Start Address“, „End Address“ oder „Subnet Mask“), den Sie ändern möchten.

- Editieren Sie den Wert direkt (Texteingabe)
oder

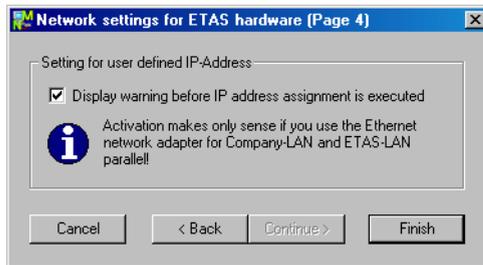
- klicken Sie auf **Default**.

Der Adressbereich und die Einstellung für die Subnetz-Maske werden vom Network Manager automatisch eingetragen. Sie können diese Einstellungen übernehmen oder überschreiben.

- Klicken Sie auf **Continue**, um weitere Einstellungen vorzunehmen.

Mit **Cancel** schließen Sie das Dialogfenster, ohne vorgenommene Änderungen zu speichern. Mit **Back** gelangen Sie in das vorhergehende Dialogfenster.

Das Dialogfenster „Network settings for ETAS hardware (Page 4)“ wird geöffnet.



Benutzerdefinierte IP Adresse einstellen:

- Aktivieren Sie die Option **Display warning before IP address assignment is executed**, wenn Sie festlegen möchten, dass eine Warnung angezeigt wird, bevor eine IP-Adresse an die ETAS-Hardware zugewiesen wird

Hinweis

Die Aktivierung dieser Warnung ist nur sinnvoll, wenn über diesen Netzwerkadapter wahlweise der PC an einem Firmennetzwerk oder ETAS-Messmodule im ETAS-Netzwerk betrieben werden sollen.

- Klicken Sie auf **Finish**.
Die Konfiguration wird beendet und das Dialogfenster geschlossen. Die Einstellungen werden gespeichert.
Mit **Cancel** schließen Sie das Dialogfenster, ohne vorgenommene Änderungen zu speichern. Mit **Back** gelangen Sie in das vorhergehende Dialogfenster.

- Starten Sie ASCET-RP und alle weitere ETAS-Software neu, damit die Änderungen übernommen werden.

Der Neustart ist lediglich erforderlich, wenn der Konfigurator nicht automatisch seitens ASCET-RP bei einer Hardwaresuche oder Initialisierung aufgerufen wurde.

Netzwerkadapter in DHCP-Umgebung konfigurieren

Adressenpool festlegen:

- Klicken Sie im Fenster „Network settings for ETAS hardware (Page 2)“ auf den Eintrag („Start Address“, „End Address“ oder „Subnet Mask“), den Sie ändern möchten.
- Editieren Sie den Wert direkt über die Tastatur (Texteingabe)

oder

- klicken Sie auf **Default**.

Der Adressbereich und die Einstellung für die Subnetz-Maske werden vom Network Manager automatisch eingetragen. Sie können diese Einstellungen übernehmen oder überschreiben.

1. Wenn Sie den Netzwerkadapter über DHCP mit APIPA oder alternativer IP-Adresse adressieren, beenden Sie die Konfiguration wie folgt:

- Klicken Sie auf **Finish**.

Die Konfiguration wird beendet und das Dialogfenster geschlossen. Die Einstellungen werden gespeichert.

- Starten Sie ASCET-RP neu, damit die Änderungen übernommen werden.

Der Neustart ist lediglich erforderlich, wenn der Konfigurator nicht automatisch seitens ASCET-RP bei einer Hardwaresuche oder Initialisierung aufgerufen wurde.

2. Wenn Sie den Netzwerkadapter über DHCP adressieren und Windows NT als Betriebssystem verwenden, setzen Sie die Konfiguration wie folgt fort:

- Klicken Sie auf **Continue**.
Das Dialogfenster „Network settings for ETAS hardware (Page 5)“ wird geöffnet.



Windows NT mit DHCP einstellen:

- Tragen Sie in das Feld „Increase Lease“ die Anzahl der Tage ein, um die die Lease-Dauer verlängert werden soll.
Mit **Cancel** schließen Sie das Dialogfenster, ohne die Änderung zu speichern. Mit **Back** gelangen Sie in das vorhergehende Dialogfenster.
- Klicken Sie auf **Finish**.
Die Konfiguration des Netzwerkadapters wird beendet und das Dialogfenster geschlossen. Die Änderungen werden gespeichert.
- Starten Sie ASCET-RP neu, damit die Änderungen übernommen werden.
Der Neustart ist lediglich erforderlich, wenn der Konfigurator nicht automatisch seitens ASCET-RP bei einer Hardwaresuche oder Initialisierung aufgerufen wurde.

12.6 Probleme bei der Hardware-Anbindung über Ethernet

In diesem Abschnitt finden Sie eine Übersicht der bei Drucklegung bekannten Probleme, die im Zusammenhang mit der Hardwareanbindung über die Ethernet-Schnittstelle auftreten können.

12.6.1 APIPA unter Windows 98 SE, 2000 oder XP deaktiviert

Der alternative Mechanismus für die IP-Adressierung (APIPA) ist in Windows 98 SE, 2000 und XP standardmäßig aktiv. Er wird jedoch in manchen Firmennetzen aus Gründen der Netzwerksicherheit deaktiviert. In diesem Fall können Sie eine Netzwerkkarte, die für DHCP-Adressierung konfiguriert ist, nicht verwenden, um damit auf ETAS-Hardware zuzugreifen. Der ETAS Network Manager gibt eine entsprechende Fehlermeldung aus.

Sie können dieses Problem beheben, indem Sie entweder eine zweite Netzwerkkarte verwenden oder indem Sie den APIPA-Mechanismus in der Windows Registry wieder aktivieren. Zum Aktivieren des APIPA-Mechanismus benötigen Sie Administratorrechte auf dem entsprechenden PC. Bevor Sie den Mechanismus wieder aktivieren, sollten Sie sich in jedem Fall mit dem zuständigen Netzwerkadministrator in Verbindung setzen.

APIPA-Mechanismus aktivieren:

- Wählen Sie im Windows-Startmenü **Start** → **Ausführen**.
- Geben Sie im Fenster „Ausführen“ `regedit` ein und klicken Sie **OK**.
Der Registrierungs-Editor wird geöffnet.
- Wählen Sie im Verzeichnisbaum des Registrierungs-Editors den Ordner
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services`.
- Wählen Sie **Bearbeiten** → **Suchen**, um alle Einträge mit dem Schlüssel `IPAutoconfigurationEnabled` zu suchen.
- Setzen Sie für jeden Eintrag den Wert für diesen Schlüssel auf 1, um den APIPA-Mechanismus zu aktivieren.
Die Windows Registry kann mehrere Einträge mit diesem Schlüssel enthalten, da der APIPA-Mechanismus sowohl für den TCP/IP-Dienst insgesamt als auch separat für jede einzelne Netzwerkkarte deaktiviert werden kann.
- Schließen Sie den Registrierungs-Editor.

Wenn Sie keine Einträge mit dem hier genannten Schlüssel in Ihrer Windows-Registry finden, wurde der APIPA-Mechanismus nicht deaktiviert.

12.6.2 Personal Firewalls

In Windows XP ist ein *Personal Firewall*-Programm im Lieferumfang enthalten und standardmäßig aktiviert. Auf vielen anderen Systemen finden sich mittlerweile häufig entsprechende Programme von unabhängigen Anbietern wie Symantec, McAfee oder BlackIce.

Firewall-Programme können in ASCET-RP die Hardwarekommunikation über die Ethernetschnittstelle behindern. Dabei werden, obwohl die Schnittstelle richtig konfiguriert ist, beim automatischen Suchen nach Hardware angeschlossene Geräte nicht gefunden. Es besteht eine hohe Wahrscheinlichkeit, dass ein installiertes *firewall* Programm die Kommunikation behindert.

Deaktivieren Sie in diesem Fall entweder das Firewall-Programm vollständig, solange Sie mit ASCET-RP arbeiten, oder geben Sie im Firewall-Programm die IP-Adresse 192.168.40.240 (Ports 18000–18005) frei. Nähere Informationen zum Deaktivieren und zum Freigeben von Adressen entnehmen Sie der Benutzerdokumentation zu Ihrem Firewall-Programm.

13 Anhang: API-Funktionen

Dieser Anhang enthält die API-Funktionen, die von ASCET-RP für das Experimentaltarget ES113x unterstützt werden. Diese Funktionen definieren die Schnittstelle zwischen ASCET-RP und folgenden Anwendungen:

- ERCOS^{EK} (Kapitel 13.1)
- NVRAM (Kapitel 13.2)
- Watchdog (Kapitel 13.3)
- LEDs (Kapitel 13.4)
- andere (Kapitel 13.5)

13.1 API-Funktionen (ERCOS^{EK})

Dieses Kapitel enthält eine detaillierte Beschreibung der API-Funktionen (**A**pplication **P**rogramming **I**nterface) zu ERCOS^{EK}. Diese Serviceroutinen definieren die Schnittstelle zwischen der Anwendung und ERCOS^{EK}. Jeder Abschnitt beschäftigt sich mit einer Gruppe von Serviceroutinen, die untereinander funktionell verwandt sind. Die Beschreibungsstruktur jeder Serviceroutine ist wie folgt:

_exampleRoutine

Funktion	Eine kurze Funktionsbeschreibung der Funktion.
Syntax	Die Syntax wird hier in Form eines C-Funktions-Prototyps angegeben. Die verwendeten C-Typen werden im folgenden Kapitel beschrieben.
Beschreibung	Dieser Abschnitt enthält eine detaillierte Beschreibung der Serviceroutine, eine Beschreibung der Parameter sowie weitere Einzelheiten und Anmerkungen, die der Benutzer berücksichtigen sollte, wenn er die Serviceroutine benutzt.
Rückgabewert	Hier werden der Typ und Wertebereich des Rückgabewertes (wenn vorhanden) und seine Bedeutung angegeben.
Beispiel	Das Beispiel zeigt eine typische Anwendung der Funktion.
Siehe auch	Liste verwandter Funktionen.
Hinweis	Einige Funktionsbeschreibungen enthalten einen Hinweis, der nützliche Zusatzinformationen liefert.

Folgende Aufstellung liefert einen kurzen Überblick über alle ERCOS^{EK}-Befehle, die von ASCET-RP für das Experimentaltarget ES113x unterstützt werden. Eingehendere Informationen (Syntax, Beispiele usw.) finden sich in den nachfolgenden Kapiteln.

Befehl	Funktion	Seite
Betriebsmodi		
DeclareAppMode	Dient als externe Deklaration eines Betriebsmodus.	407
SetNextAppMode	Schaltet zum spezifizierten Betriebsmodus um, nachdem alle aktiven Tasks verarbeitet worden sind.	407
Tasks		
DeclareTask	Dient als externe Deklaration einer Task.	408
ActivateTask	Aktiviert eine SW-Task.	408
Systemzeit		
GetSystemTime	Gibt die aktuelle Systemzeit zurück.	409
GetSystemTimeLow	Gibt den niederwertigen Teil der aktuellen Systemzeit zurück.	410
GetSystemTimeHigh	Gibt den höherwertigen Teil der aktuellen Systemzeit zurück.	410
Interrupt-Behandlung		
EnableAllInterrupts	Gibt global alle Interrupts frei.	411
DisableAllInterrupts	Sperrt global alle Interrupts.	411
dT-Anfrage		
GetDeltaT	Gibt den Wert von dT zurück.	412

13.1.1 Betriebsmodi

Das Konzept der Betriebsmodi (Englisch: *application modes*) ermöglicht die effiziente Verwaltung verschiedener Verarbeitungszustände in der Anwendungssoftware. Ein Betriebsmodus wird von einer Reihe von Tasks definiert, die in diesem Modus aktiv sind, sowie von einer oder mehreren optionalen Zeitsteuertabellen. Betriebsmodi für ein Motor-Steuergerät können zum Bei-

spiel sein: Normalbetrieb (Steuerung des technischen Prozesses), Selbstdiagnose, Flash-EPROM-Programmierung. Es kann immer nur ein Betriebsmodus aktiv sein. Ein Betriebsmodus besteht aus zwei Phasen: die erste Phase ist die Initialisierungsphase. Hier werden die Initialisierungsroutinen der Anwendung verarbeitet. Während dieser Phase sind Interrupts gesperrt. Nach der Initialisierung werden die Interrupts aktiviert und die Ausführungsphase beginnt. Hier werden die aktivierten Tasks der Anwendung gemäß ihren festgelegten Prioritäten verarbeitet.

DeclareAppMode

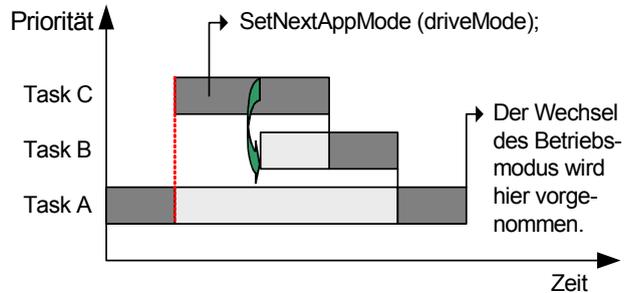
Funktion	Dient als externe Deklaration eines Betriebsmodus.
Syntax	<pre>#define DeclareAppMode(AppID) extern AppModeType AppID</pre>
Beschreibung	Wenn innerhalb eines Moduls ein Betriebsmoduswechsel durchgeführt wird, der Deskriptor des Betriebsmodus jedoch in einem anderen Modul definiert ist, muss die Verwendung des Betriebsmodus-Deskriptors durch DeclareAppMode() deklariert werden. Funktion und Gebrauch dieses Dienstes sind ähnlich derer der externen Deklaration von Variablen.
Beispiel	<pre>extern uint excCtr; extern uint randx; DeclareAppMode(idleMode);</pre>
Siehe auch	DeclareTask

SetNextAppMode

Funktion	Wechselt nach der Verarbeitung aller aktiven Tasks zum angegebenen Betriebsmodus.
Syntax	<pre>StatusType SetNextAppMode(AppModeType appMode)</pre>
Beschreibung	SetNextAppMode() fordert einen Wechsel zum Betriebsmodus an, auf den der Zeiger appMode verweist. Das Betriebssystem führt den Wechsel aus, sobald sich keine weitere Task im Zustand <i>Running</i> befindet, d.h. wenn das Betriebssystem untätig ist. Nachfolgende Taskaktivierungen über ChainTask() oder RestartTask() (s. ERCOS ^{EK} -Handbuch) werden ignoriert.

Rückgabewert E_OK Anfrage erfolgreich verarbeitet.

Beispiel `SetNextAppMode(driveMode);`



Siehe auch -

13.1.2 Tasks

In ERCOS^{EK} gibt es zwei Task-Typen: Software-Tasks (SW-Tasks), die durch **ActivateTask()** aktiviert werden, wobei die Verarbeitung durch den ERCOS^{EK} Scheduler koordiniert wird, und Hardware-Tasks (HW-Tasks), die durch einen Interrupt aktiviert werden. In diesem Fall wird das Scheduling von der Interrupt-Steuerlogik des Prozessors durchgeführt, d.h. von der Hardware.

DeclareTask

Funktion Dient als externe Deklaration einer Task.

Syntax `#define DeclareTask(TaskID)`
`extern TaskType TaskID`

Beschreibung Wird eine Task von einem Modul benutzt, jedoch in einem anderen Modul definiert, muss ihre Verwendung durch **DeclareTask()** deklariert werden. Die Funktion und Benutzung dieses Dienstes sind ähnlich wie bei der externen Deklaration von Variablen.

Beispiel `extern uint excCtr;`
`extern uint randx;`
`DeclareTask(synchroSeq);`

Siehe auch `DeclareAppMode`

ActivateTask

Funktion Aktiviert eine SW-Task.

Syntax `StatusType ActivateTask(TaskType task)`

Beschreibung	ActivateTask() fordert das OS auf, die durch task angegebene SW-Task zu verarbeiten. Ist diese Task-Aktivierung erfolgreich (s. Rückgabewert), wird die Verarbeitung der Task gemäß ihrer Priorität vom ERCOS ^{EK} -Scheduler geplant. Sind verschiedene Aktivierungen einer Task zulässig (gemäß der BCC2-Definition) und die aktuelle Anzahl von Aktivierungen einer Task > 1 ist, wird diese Task temporär im FIFO-Puffer gespeichert. Kann ActivateTask() nicht erfolgreich ausgeführt werden, schaltet das OS zur benutzerspezifischen Fehlerfunktion um.	
Rückgabewert	E_OK	Aktivierung erfolgreich.
	E_OS_LIMIT	Keine Aktivierung, da die Höchstzahl der Task-Aktivierungen oder der Tasks im FIFO-Puffer der jeweiligen Prioritätsebene bereits erreicht.
Beispiel	ActivateTask(synchroSeq);	
Siehe auch	-	

13.1.3 Systemzeit

Eine diskrete Systemzeit ist die Zeitbasis von ERCOS^{EK}. Für Zielplattformen, die keine auf der Hardware basierende Systemzeit anbieten, wird die Systemzeit mit dem Start des Betriebssystems auf 0 gesetzt. Die Systemzeit, die normalerweise mit der Breite von zwei Maschinenwörtern gezählt wird, wird als Bezugszeit für Alarmdienste und die ERCOS^{EK}-Zeitsteuertabelle verwendet. Die Zeit, die vergeht, bis ein Überlauf der Systemzeit auftritt, hängt von der CPU und der Frequenz des verwendeten Hardware-Timers ab. Die Systemzeit wird von keinem Wechsel des Betriebsmodus unterbrochen oder zurückgesetzt.

Die Systemzeit wird in Ticks des zugrundeliegenden Hardware-Timers gezählt. Das API-Makro `SYSTEM_TICK_DURATION` gibt die Dauer eines solchen Tick in Nanosekunden zurück.

GetSystemTime

Funktion	Gibt die aktuelle Systemzeit zurück.
Syntax	<code>TimeType GetSystemTime(void)</code>
Beschreibung	GetSystemTime() gibt die Systemzeit in Ticks zurück. Die Breite hängt vom System ab (32 Bit bei 16-Bit-Systemen und 64 Bit bei 32-Bit-Systemen).

Rückgabewert Aktuelle Systemzeit.
Beispiel `TimeType now;
now = GetSystemTime();`
Siehe auch `GetSystemTimeLow, GetSystemTimeHigh`

GetSystemTimeLow

Funktion Gibt den niederwertigen Teil der aktuellen Systemzeit zurück.
Syntax `TickType GetSystemTimeLow(void)`
Beschreibung **GetSystemTimeLow()** gibt den niederwertigen Teil der aktuellen Systemzeit in Ticks zurück. Dies sind die unteren 16 Bit bei einer ERCOS^{EK} Implementierung mit einer 32 Bit breiten Systemzeit; bei einer Implementierung mit einer 64 Bit breiten Systemzeit sind es die unteren 32 Bit.
Rückgabewert Niederwertiger Teil der aktuellen Systemzeit.
Beispiel `TickType lowPartOfNow;
lowPartOfNow = GetSystemTimeLow();`
Siehe auch `GetSystemTime, GetSystemTimeHigh`

GetSystemTimeHigh

Funktion Gibt den höherwertigen Teil der aktuellen Systemzeit zurück.
Syntax `TickType GetSystemTimeHigh(void)`
Beschreibung **GetSystemTimeHigh()** gibt den höherwertigen Teil der aktuellen Systemzeit in Ticks zurück. Dies sind die oberen 16 Bit bei einer ERCOS^{EK} Implementierung mit einer 32 Bit breiten Systemzeit; bei einer Implementierung mit einer 64 Bit breiten Systemzeit sind es die oberen 32 Bit.
Rückgabewert Höherwertiger Teil der aktuellen Systemzeit.
Beispiel `TickType highPartOfNow;
highPartOfNow = GetSystemTimeHigh();`
Siehe auch `GetSystemTime, GetSystemTimeLow`

13.1.4 Interruptbehandlung

ERCOS^{EK} stellt eine Routine zur Verfügung, um kontextrelevante Daten bei einer Interrupt-Serviceroutine zu sichern und wiederherzustellen. Weiterhin kann mit der ERCOS^{EK} API-Funktion auf den mit Sicherheit gültigen Unterbrechungsdeskriptor zugegriffen werden.

EnableAllInterrupts

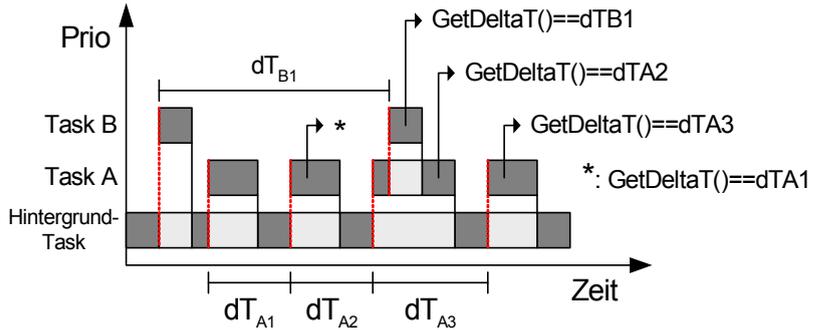
Funktion	Gibt global die Interrupts frei.
Syntax	<code>void EnableAllInterrupts(void)</code>
Beschreibung	EnableAllInterrupts() gibt global die Interrupts für den Controller-Kern frei, ohne die selektive Maskierung zu beeinflussen. Wurde zuvor DisableAllInterrupts() mehrfach aufgerufen, erfolgt die Freigabe erst, wenn EnableAllInterrupts() in der entsprechenden Anzahl aufgerufen wurde. Dadurch wird eine saubere Schachtelung der Aufrufe – ohne Seiteneffekte – unterstützt.
Rückgabewert	Ohne
Siehe auch	<code>DisableAllInterrupts</code>

DisableAllInterrupts

Funktion	Sperrt global alle Interrupts.
Syntax	<code>void DisableAllInterrupts(void)</code>
Beschreibung	DisableAllInterrupts() sperrt global die Interrupts und sichert den Zustand verschachtelter Aufrufe.
Rückgabewert	Ohne
Siehe auch	<code>EnableAllInterrupts</code>

13.1.5 dT-Abfrage

ERCOS^{EK} stellt eine Serviceroutine für die Abfrage der zwischen dem Start und dem letzten Start der aktuell laufenden Task vergangenen Zeit zur Verfügung (siehe Abbildung unten). Die angegebene Zeit bezieht sich immer auf die Task, von der aus die Funktion aufgerufen wurde.



Der von `GetDeltaT()` zurückgegebene Wert dT ist sehr nützlich für mathematische Berechnungen, z. B. eines Integrals:

$$F(x) = \int_0^x f(T) dT$$

GetDeltaT

Funktion	Gibt den Wert von dT zurück.
Syntax	<code>TickType GetDeltaT(void)</code>
Beschreibung	<p>GetDeltaT() gibt die Zeit an, die zwischen zwei aufeinanderfolgenden Taskausführungen vergangen ist.</p> <p>Anmerkung: Wenn diese Zeit die Hälfte der Breite des Hardware-Timers überschreitet, ist der Rückgabewert nicht zuverlässig.</p> <p>Diese Funktion wird nur im ERCOS^{EK} Debugmodus unterstützt. Siehe Kapitel „Debuginformationen innerhalb des Taskmonitors“ im ERCOS^{EK} Handbuch, um detaillierte Informationen über das Debugging einer auf ERCOS^{EK} basierenden Anwendung zu erhalten.</p>

Rückgabewert Wert von dT in Ticks.
 Beispiel `TickType deltaT;`
`deltaT = GetDeltaT();`
 Siehe auch –

13.2 API-Funktionen (NVRAM)

Das in Kapitel 6.2.1 beschriebene Standardverhalten des NVRAM-Managers in der ES1135-Firmware kann über die folgenden Schnittstellen aus einem ASCET-Modell (C-Code item) heraus verändert werden.

nvrAmInitModelVars

Funktion	Initialisiert die NV-Variablen	
Syntax	<code>uint32 nvrAmInitModelVars(void)</code>	
Beschreibung	Diese Funktion initialisiert die NV-Variablen mit dem NVRAM-Inhalt, wenn dieser gültig und passend ist. Die Initialisierung darf nur einmal angestoßen werden (via C-Code, Experimentierumgebung oder automatisch), und nur, bevor eine Aktualisierung des NVRAM-Inhalts stattfand.	
Rückgabewert	<code>EC_NVRAM_SUCCESS</code>	Erfolg
	<code>EC_NVRAM_NO_NV_VARIABLES</code>	keine NV-Variablen im Modell
	<code>EC_NVRAM_INADMISSIBLE_USE</code>	Funktion wurde bereits aufgerufen
	<code>EC_NVRAM_INTERNAL_ERROR</code>	interner Fehler
	<code>EC_NVRAM_NO_MATCH</code>	NVRAM-Inhalt stimmt nicht mit aktuellem Modell überein.
Beispiel	–	
Siehe auch	<code>nvrAmCheckForInitializedVars</code>	

nvrAmSetUpdateInterval

Funktion	Setzt das Intervall für die automatische NVRAM-Aktualisierung.	
Syntax	<code>uint32 nvrAmSetUpdateInterval(uint32 interval_sec)</code>	
Beschreibung	Setzt das Intervall für die automatische NVRAM-Aktualisierung. Dies ist die gewünschte Zeit zwischen zwei Aktualisierungen. Wenn die Systemlast hoch ist, kann das tatsächliche Zeitintervall größer sein (hängt wesentlich vom verlangten Konsistenzgrad ab). Wenn das tatsächliche Intervall das gewünschte um den Faktor 10 übersteigt, wird in der Experimentierumgebung eine Warnung ausgegeben.	
Rückgabewert	<code>EC_NVRAM_SUCCESS</code>	Erfolg
	<code>EC_NVRAM_INVALID_ARG</code>	<code>interval_sec > 30</code>
Parameter	<code>interval_sec</code>	Aktualisierungsintervall in Sekunden. Muss ein Wert zwischen 0 und 30 sein (0: keine periodische Aktualisierung).
Beispiel	–	
Siehe auch	<code>nvrAmGetUpdateInterval</code>	

nvrAmGetUpdateInterval

Funktion	Liefert das Intervall für die automatische NVRAM-Aktualisierung.	
Syntax	<code>uint32 nvrAmGetUpdateInterval(void)</code>	
Beschreibung	Liefert das Intervall für die automatische NVRAM-Aktualisierung. Dies ist die gewünschte Zeit zwischen zwei Aktualisierungen. Wenn die Systemlast hoch ist, kann das tatsächliche Zeitintervall größer sein (hängt wesentlich vom verlangten Konsistenzgrad ab). Wenn das tatsächliche Intervall das gewünschte um den Faktor 10 übersteigt, wird in der Experimentierumgebung eine Warnung ausgegeben.	
Rückgabewert	<code>interval_sec</code>	Aktualisierungsintervall in Sekunden.
Beispiel	–	
Siehe auch	<code>nvrAmSetUpdateInterval</code>	

nvrAmSetConsistencyLevel

Funktion	Setzt den Konsistenzgrad der NV-Variablen.	
Syntax	<code>uint32 nvrAmSetConsistencyLevel (T_consistencyLevel level)</code>	
Beschreibung	Setzt den Konsistenzgrad für die NV-Variablen. <i>Keine Konsistenz:</i> NVRAM-Aktualisierung ohne Rücksicht auf Konsistenz innerhalb einer NV-Variablen oder zwischen verschiedenen NV-Variablen. <i>Niedrige Konsistenz:</i> Datenkonsistenz innerhalb einer NV-Variablen (Skalar, Array, Matrix; <i>nicht</i> Kennlinie/-feld) wird garantiert. <i>Hohe Konsistenz:</i> Alle NV-Variablen werden, ohne Unterbrechung durch das Modell, in der Idle-Task aktualisiert.	
Rückgabewert	<code>EC_NVRAM_SUCCESS</code>	Erfolg
	<code>EC_NVRAM_INVALID_ARG</code>	ungültiges Argument <code>level</code>
Parameter	<code>level</code>	<code>NVRAM_NO_CONSISTENCY</code> <code>NVRAM_LOW_CONSISTENCY</code> <code>NVRAM_HIGH_CONSISTENCY</code>
Beispiel	–	
Siehe auch	<code>nvrAmGetConsistencyLevel</code>	

nvrAmGetConsistencyLevel

Funktion	Liest den Konsistenzgrad der NV-Variablen.	
Syntax	<code>T_consistencyLevel nvrAmGetConsistencyLevel(void)</code>	
Beschreibung	Liest den Konsistenzgrad für die NV-Variablen. <i>Keine Konsistenz:</i> NVRAM-Aktualisierung ohne Rücksicht auf Konsistenz innerhalb einer NV-Variablen oder zwischen verschiedenen NV-Variablen. <i>Niedrige Konsistenz:</i> Datenkonsistenz innerhalb einer NV-Variablen (Skalar, Array, Matrix; <i>nicht</i> Kennlinie/-feld) wird garantiert. <i>Hohe Konsistenz:</i> Alle NV-Variablen werden ohne Unterbrechung durch das Modell in der Idle-Task aktualisiert.	
Rückgabewert	<code>NVRAM_NO_CONSISTENCY</code>	keine Konsistenz
	<code>NVRAM_LOW_CONSISTENCY</code>	niedrige Konsistenz

nvrAmGetConsistencyLevel

	<code>NVRAM_HIGH_CONSISTENC</code>	hohe Konsistenz
	<code>Y</code>	
Beispiel	–	
Siehe auch	<code>nvrAmSetConsistencyLevel</code>	

nvrAmEnableAutoUpdate

Funktion	Aktiviert die automatische Aktualisierung der NV-Variablen.	
Syntax	<code>uint32 nvrAmEnableAutoUpdate(void)</code>	
Beschreibung	Aktiviert die automatische Aktualisierung der NV-Variablen. Dies umfasst sowohl periodische Aktualisierung als auch durch die Exit-Task angestoßene Aktualisierung.	
Rückgabewert	<code>EC_NVRAM_SUCCESS</code>	Erfolg
Beispiel	–	
Siehe auch	<code>nvrAmDisableAutoUpdate</code> , <code>nvrAmCheckForAutoUpdate</code>	

nvrAmDisableAutoUpdate

Funktion	Deaktiviert die automatische Aktualisierung der NV-Variablen.	
Syntax	<code>uint32 nvrAmDisableAutoUpdate(void)</code>	
Beschreibung	Deaktiviert die automatische Aktualisierung der NV-Variablen. Dies umfasst sowohl periodische Aktualisierung als auch durch die Exit-Task angestoßene Aktualisierung.	
Rückgabewert	<code>EC_NVRAM_SUCCESS</code>	Erfolg
Beispiel	–	
Siehe auch	<code>nvrAmEnableAutoUpdate</code> , <code>nvrAmCheckForAutoUpdate</code>	

nvrAmCheckForAutoUpdate

Funktion	Prüft, ob die automatische Aktualisierung der NV-Variablen aktiviert ist.	
Syntax	<code>uint8 nvrAmCheckForAutoUpdate(void)</code>	
Rückgabewert	<code>true</code>	Automatische Aktualisierung ist aktiviert

nvrAmCheckForAutoUpdate

	<code>false</code>	Automatische Aktualisierung ist deaktiviert
Beispiel	–	
Siehe auch	<code>nvrAmEnableAutoUpdate</code> , <code>nvrAmDisableAutoUpdate</code>	

nvrAmManualUpdateExit

Funktion	Bewirkt eine abschließende Aktualisierung des NVRAM-Inhalts.
Syntax	<code>void nvrAmManualUpdateExit (void)</code>
Beschreibung	Diese Funktion sollte nach dem letzten Anwenderprozess in die Exit-Task platziert werden, damit eine abschließende Aktualisierung des NVRAM-Inhalts bei Verlassen des Anwender-Betriebsmodus sichergestellt ist. Wenn Fehler auftreten, werden Fehlermeldungen in der Experimentierumgebung angezeigt.
Beispiel	–
Siehe auch	<code>nvrAmManualUpdateBackground</code> , <code>nvrAmManualUpdateBlocked</code>

nvrAmManualUpdateBackground

Funktion	Startet eine manuelle Aktualisierung des NVRAM-Inhalts.
Syntax	<code>void nvrAmManualUpdateBackground (void)</code>
Beschreibung	Diese Funktion startet eine manuelle Aktualisierung des NVRAM-Inhalts. Manuelle Aktualisierung hat Vorrang vor automatischer Aktualisierung. Daher wird eine aktivierte (aber noch nicht laufende) automatische Aktualisierung abgebrochen. Aber wenn eine zyklische Aktivierung tatsächlich in Arbeit ist (die Idle-Task wird durch eine präemptive Task, die diese Funktion aufruft, unterbrochen), ist der Start der manuellen Aktualisierung unmöglich. Diese Funktion springt sofort zurück ins Programm, denn die Aktualisierung läuft im Hintergrund (Idle-Task) ab. Die Komplettierung dieser Funktion kann mit <code>nvrAmCheckRunningUpdate()</code> geprüft werden.

Anmerkung: Es wird **nicht** empfohlen, diese Funktion zu verwenden, wenn automatische Aktualisierung aktiviert ist.

Rückgabewert	<code>EC_NVRAM_SUCCESS</code>	Erfolg
	<code>EC_NVRAM_NO_NV_VARIABLE</code>	keine NV-Variable im Modell S

nvrManualUpdateBackground

	<code>EC_NVRAM_FATAL_ERROR</code>	Schwerer Fehler ist zuvor aufgetreten
	<code>EC_NVRAM_OVERFLOW</code>	Überlauf des NVRAM. Größe / Anzahl der NV-Variablen verringern.
	<code>EC_NVRAM_UPDATE_RUNNING</code>	Frühere Aktualisierung (manuell / automatisch) läuft. Start der manuellen Aktualisierung fehlgeschlagen.
Beispiel	–	
Siehe auch	<code>nvrManualUpdateBlocked</code> , <code>nvrManualUpdateExit</code>	

nvrManualUpdateBlocked

Funktion	Startet eine manuelle Aktualisierung des NVRAM-Inhalts (auf der aktuellen Prioritätsebene blockierend).	
Syntax	<code>uint32 nvrManualUpdateBlocked</code> <code>(uint32 timeoutUs)</code>	
Beschreibung	<p>Diese Funktion startet eine manuelle Aktualisierung des NVRAM-Inhalts. Manuelle Aktualisierung hat Vorrang vor automatischer Aktualisierung. Daher wird eine eine aktivierte (aber noch nicht laufende) automatische Aktualisierung abgebrochen. Aber wenn eine zyklische Aktivierung tatsächlich in Arbeit ist (die Idle-Task wird durch eine präemptive Task, die diese Funktion aufruft, unterbrochen) ist der Start der manuellen Aktualisierung unmöglich. Diese Funktion blockiert die aktuelle Prioritätsebene, bis der Inhalt aller NV-Variablen in den lokalen Puffer geschrieben wurden, oder bis ein Timeout eintritt. Nachdem die Funktion ins Programm zurückgesprungen ist, wird der Aktualisierungsprozess (aus dem lokalen Puffer in den NVRAM schreiben) in der Idle-Task fortgesetzt (auch wenn ein Timeout eintrat).</p> <p>Die Komplettierung dieser Funktion kann mit <code>nvrManualUpdateRunningUpdate()</code> geprüft werden.</p> <p>Anmerkung: Es wird nicht empfohlen, diese Funktion zu verwenden, wenn automatische Aktualisierung aktiviert ist.</p>	
Rückgabewert	<code>EC_NVRAM_SUCCESS</code>	Erfolg
	<code>EC_NVRAM_NO_NV_VARIABLES</code>	keine NV-Variable im Modell

nvrAmManualUpdateBlocked

	<code>EC_NVRAM_FATAL_ERROR</code>	Schwerer Fehler ist zuvor aufgetreten
	<code>EC_NVRAM_OVERFLOW</code>	Überlauf des NVRAM. Größe / Anzahl der NV-Variablen verringern.
	<code>EC_NVRAM_UPDATE_RUNNING</code>	Frühere Aktualisierung (manuell / automatisch) läuft. Start der manuellen Aktualisierung fehlgeschlagen.
Parameter	<code>timeoutUs</code>	Timeout-Periode in μ s.
Beispiel	–	
Siehe auch	<code>nvrAmManualUpdateBlocked</code> , <code>nvrAmManualUpdateExit</code> , <code>nvrAmCheckRunningUpdate</code>	

nvrAmCheckRunningUpdate

Funktion	Prüft, ob eine manuelle NVRAM-Aktualisierung gestartet wurde.	
Syntax	<code>uint8 nvrAmCheckRunningUpdate(void)</code>	
Beschreibung	Diese Funktion prüft, ob eine manuelle NVRAM-Aktualisierung, die mit <code>nvrAmStartManualUpdateBackground</code> oder <code>nvrAmStartManualUpdateBlocked</code> gestartet wurde, noch im Hintergrund läuft.	
Rückgabewert	<code>false</code>	Aktualisierung wurde beendet oder nicht erfolgreich gestartet.
	<code>true</code>	Aktualisierung läuft noch.
Beispiel	–	
Siehe auch	<code>nvrAmManualUpdateBackground</code> , <code>nvrAmManualUpdateBlocked</code>	

nvrAmCheckForInitializedVars

Funktion	Prüft, ob die NV-Variablen initialisiert wurden.	
Syntax	<code>uint8 nvrAmCheckForInitializedVars(void)</code>	
Beschreibung	Diese Funktion prüft, ob die NV-Variablen im Modell mit dem NVRAM-Inhalt initialisiert wurden. Dies kann durch die automatische Aktualisierung per Experimentierumgebung oder per Initialisierung über die C-Code-API geschehen.	

nvrAmCheckForInitializedVars

Rückgabewert	<code>true</code>	NV-Variablen wurden mit dem NVRAM-Inhalt initialisiert.
	<code>false</code>	NV-Variablen wurden nicht mit dem NVRAM-Inhalt, sondern mit ihren Standardwerten initialisiert.
Beispiel	–	
Siehe auch	<code>nvrAmInitModelVars</code>	

nvrAmGetUpdateAgeMs

Funktion	Liefert die Zeit seit dem letzten Ende einer Aktualisierung.	
Syntax	<code>uint32 nvrAmGetUpdateAgeMs(void)</code>	
Beschreibung	Diese Funktion liefert die Zeit, die seit dem letzten Ende einer Aktualisierung (manuell oder automatisch) verstrichen ist.	
Rückgabewert	<code>updateAge</code>	Zeit in Millisekunden
Beispiel	–	
Siehe auch	–	

nvrAmClear

Funktion	Löscht den NVRAM-Inhalt.	
Syntax	<code>uint32 nvrAmClear(void)</code>	
Beschreibung	Diese Funktion löscht den Inhalt des NVRAM. Der Speicher wird mit Nullen initialisiert.	
Beispiel	–	
Siehe auch	–	

13.3 API-Funktionen (Watchdog)

Der Simulationscontroller ES1135 besitzt einen Hardware-Watchdog. Die Funktionalität ist in Kapitel 6.2.2 zusammengefasst. Die folgenden Schnittstellen werden von der Firmware bereitgestellt.

13.3.1 Watchdog-Konfiguration

wdSetSafetyMode

Funktion	Schaltet in den den Sicherheitsmodus.	
Syntax	<code>uint32 wdSetSafetyMode (uint32 event, uint32 period)</code>	
Beschreibung	<p>Diese Funktion schaltet aus dem präoperativen oder dem RSEF-Modus in den sicherheitsorientierten Modus um. Dies kann nicht rückgängig gemacht werden, außer durch Abschalten der Stromversorgung.</p> <p>Der Parameter event wählt die Aktion, die ausgeführt werden soll, wenn der Watchdog abläuft.</p> <p><code>WD_EVENT_DISABLE</code> schaltet den Watchdog ab.</p> <p><code>WD_EVENT_PPC750_RESET</code> setzt den IBM 750GX Simulationsprozessor zurück.</p> <p><code>WD_EVENT_PPC750_INT</code> löst einen Interrupt für den Simulationsprozessor aus.</p> <p>Der Parameter period (Zeit, nach der der Watchdog abläuft) kann im Bereich von 0.25 ms bis 4096 ms eingestellt werden.</p>	
Rückgabewert	<code>EC_CFW_SUCCESS</code>	Erfolg
	<code>EC_CFW_WD_SAFETY_MODE</code>	Watchdog ist bereits im Sicherheitsmodus.
	<code>EC_CFW_INVALID_ARG</code>	ungültiges Ereignis oder Periodenwert
Parameter	<code>event</code>	<code>WD_EVENT_DISABLE</code> <code>WD_EVENT_PPC750_RESET</code> <code>WD_EVENT_PPC750_INT</code>

wdSetSafetyMode

period	WD_PERIOD_4096MS
	WD_PERIOD_1024MS
	WD_PERIOD_256MS
	WD_PERIOD_64MS
	WD_PERIOD_16MS
	WD_PERIOD_4MS
	WD_PERIOD_1MS
	WD_PERIOD_0_25MS

Beispiel

```
uint32 period;
uint32 event;
uint32 retVal;
event = WD_EVENT_DISABLE;
period = WD_PERIOD_4096MS;
retVal = wdSetSafetyMode(event, period);
```

Siehe auch `wdSetPeriod`, `wdSetEvent`

wdSetReducedSafetyMode

Funktion Schaltet in den RSEF-Modus.

Syntax `uint32 wdSetReducedSafetyMode(void)`

Beschreibung Diese Funktion schaltet aus dem präoperativen Modus in den RSEF-Modus (reduced safety, enhanced functionality) um.

Anmerkung: Diese Funktion wird bereits vom Boot-Loader aufgerufen. Daher hat diese Funktion keinen Einfluss auf die Verwendung von ASCET-RP, weil das Modell mit dem Watchdog im RSEF-Modus startet.

Der Loader deaktiviert auch die Watchdog-Ereignisse. Anschließend können Watchdog-Periode und -Ereignis mit `wdSetPeriod` und `wdSetEvent` verändert werden.

Rückgabewert `EC_CFW_SUCCESS` Erfolg

`EC_CFW_WD_SAFETY_MODE` Watchdog ist im Sicherheitsmodus. Dies kann nicht rückgängig gemacht werden.

`EC_CFW_WD_RSEF_MODE` Watchdog ist bereits im RSEF-Modus.

Beispiel -

Siehe auch `wdSetPeriod`, `wdSetEvent`

wdSetPeriod

Funktion	Setzt die Watchdog-Periode.	
Syntax	<code>uint32 wdSetPeriod(uint32 period)</code>	
Beschreibung	Diese Funktion bestimmt die Watchdog-Periode (Zeit, nach der der Watchdog abläuft), die im Bereich von 0.25 ms bis 4096 ms eingestellt werden kann.	
Rückgabewert	<code>EC_CFW_SUCCESS</code>	Erfolg
	<code>EC_CFW_WD_SAFETY_MODE</code>	Watchdog im Sicherheitsmodus. Keine Änderung der Periode möglich.
	<code>EC_CFW_INVALID_ARG</code>	ungültiger Periodenwert
	<code>EC_CFW_WD_PRE_OP_MODE</code>	Watchdog im präoperativen Modus. Zuerst in den RSEF-Modus wechseln.
Parameter	<code>period</code>	<code>WD_PERIOD_4096MS</code> <code>WD_PERIOD_1024MS</code> <code>WD_PERIOD_256MS</code> <code>WD_PERIOD_64MS</code> <code>WD_PERIOD_16MS</code> <code>WD_PERIOD_4MS</code> <code>WD_PERIOD_1MS</code> <code>WD_PERIOD_0_25MS</code>
Beispiel	<pre>uint32 period; uint32 retVal; period = WD_PERIOD_4096MS; retVal = wdSetPeriod(period);</pre>	
Siehe auch	<code>wdSetSafetyMode</code> <code>wdSetEvent</code>	

wdSetEvent

Funktion	Sets the event to be handled, if the watchdog expires.	
Syntax	<code>uint32 wdSetEvent(uint32 event)</code>	
Beschreibung	Diese Funktion wählt die Aktion, die bei Ablauf des Watchdog ausgeführt werden soll. WD_EVENT_DISABLE schaltet den Watchdog ab. WD_EVENT_PPC750_RESET setzt den IBM 750GX Simulationsprozessor zurück. WD_EVENT_PPC750_INT löst einen Interrupt für den Simulationsprozessor aus.	
Rückgabewert	EC_CFW_SUCCESS	Erfolg
	EC_CFW_WD_SAFETY_MODE	Watchdog im Sicherheitsmodus. Keine Änderung der Periode möglich.
	EC_CFW_INVALID_ARG	ungültiger Wert für das Ereignis
	EC_CFW_WD_PRE_OP_MODE	Watchdog im präoperativen Modus. Zuerst in den RSEF-Modus wechseln.
Parameter	event	WD_EVENT_DISABLE WD_EVENT_PPC750_RESET WD_EVENT_PPC750_INT
Beispiel	<pre>uint32 event; uint32 retVal; event = WD_EVENT_DISABLE; retVal = wdSetEvent(event);</pre>	
Siehe auch	wdSetSafetyMode, wdSetPeriod	

13.3.2 Watchdog-Service

wdService

Funktion	Bedient den Watchdog.
Syntax	<code>Void wdSetEvent(void)</code>
Beschreibung	Diese Funktion bedient den Watchdog. Das heißt, sie initialisiert den Watchdog-Timer mit dem über <code>wdSetPeriod()</code> vorgegebenen Wert.
Beispiel	<code>wdService();</code>
Siehe auch	<code>wdEnableAutoService</code> , <code>wdDisableAutoService</code>

wdEnableAutoService

Funktion	Aktiviert automatisches Bedienen.
Syntax	<code>void wdEnableAutoService(void)</code>
Beschreibung	Diese Funktion aktiviert das automatische Bedienen des Watchdog. Sie bedient den Watchdog im Abstand von 30 ms, wenn Interrupts aktiviert sind. Zusätzliches Bedienen kann über die RTIO-Gerätetreiber erfolgen. Dieser Dienst ist standardmäßig aktiviert.
Beispiel	<code>wdEnableAutoService();</code>
Siehe auch	<code>wdService</code> , <code>wdDisableAutoService</code>

wdDisableAutoService

Funktion	Deaktiviert automatisches Bedienen.
Syntax	<code>void wdDisableAutoService(void)</code>
Beschreibung	Diese Funktion deaktiviert das automatische Bedienen des Watchdog. Anmerkung: Es ist Sache des Modells, den Watchdog angemessen zu bedienen. Bedenken Sie, dass die Deaktivierung der automatischen Bedienung auch RTIO-interne Dienstaufrufe deaktiviert. Da RTIO-Treiberaufrufe (speziell Treiber Init und Exit) potentiell länger blockieren, sollte die automatische Bedienung in den Init- und Exit-Tasks aktiviert sein.
Beispiel	<code>wdDisableAutoService();</code>
Siehe auch	<code>wdService</code> , <code>wdEnableAutoService</code>

13.3.3 Interruptsteuerung

wdIntEnable

Funktion	Aktiviert die Watchdog-Interruptbehandlung.
Syntax	<code>void wdIntEnable(void)</code>
Beschreibung	Diese Funktion aktiviert die Watchdog-Interruptbehandlung. Verwenden Sie zuvor <code>wdSetEvent()</code> , um das Watchdog-Ereignis geeignet abzubilden. Der Aufruf von <code>wdIntEnable()</code> hat nur Einfluss auf die Übertragung der Interrupts. <code>wdIntPend()</code> kann verwendet werden, selbst wenn Watchdog-Interrupts deaktiviert sind.
Beispiel	<code>wdIntEnable();</code>
Siehe auch	<code>wdSetEvent</code> , <code>wdIntPend</code> , <code>wdIntDisable</code> , <code>wdIntAck</code>

wdIntDisable

Funktion	Deaktiviert die Watchdog-Interruptbehandlung.
Syntax	<code>void wdIntDisable(void)</code>
Beschreibung	Diese Funktion deaktiviert die Watchdog-Interruptbehandlung.
Beispiel	<code>wdIntDisable();</code>
Siehe auch	<code>wdIntEnable</code>

wdIntPend

Funktion	Prüft, ob ein Interrupt ansteht.
Syntax	<code>uint8 wdIntPend(void)</code>
Beschreibung	Diese Funktion prüft, ob ein Watchdog-Interrupt ansteht. Verwenden Sie zuvor <code>wdSetEvent()</code> , um das Watchdog-Ereignis geeignet abzubilden.
Rückgabewert	<code>false</code> kein Watchdog-Interrupt steht an

wdIntPend

Beispiel `true` Watchdog-Interrupt steht an

```
if(wdIntPend() == true)
{
    intPollCount++;
    /* Reset Interrupt */
    wdIntAck();
}
```

Siehe auch `wdSetEvent`, `wdIntDisable`, `wdIntAck`

wdIntAck

Funktion Bestätigt Watchdog-Interrupt.

Syntax `void wdIntAck(void)`

Beschreibung Diese Funktion bestätigt den Watchdog-Interrupt. Der Watchdog-Zähler (automatischer Neustart, nachdem ein Ereignis ausgelöst wurde) wird durch diesen Aufruf nicht beeinflusst. Wenn der Watchdog-Zähler initialisiert werden soll, rufen Sie zuvor `wdService()` auf.

Beispiel

```
if(wdIntPend() == true)
{
    intPollCount++;
    /* Reset Interrupt */
    wdIntAck();
}
```

Siehe auch `wdSetEvent`, `wdIntDisable`, `wdIntPend`

13.3.4 Watchdog Status

wdCheckReducedSafetyMode

Funktion Prüft, ob Watchdog im RSEF-Modus ist.

Syntax `uint8 wdCheckReducedSafetyMode(void)`

Beschreibung Diese Funktion prüft, ob der Watchdog im RSEF-Modus (reduced safety, enhanced function) ist. Wenn ja, können die Watchdog-Einstellungen zur Laufzeit geändert werden.

Rückgabewert `false` Watchdog läuft im Sicherheitsmodus.

wdCheckReducedSafetyMode

	<code>true</code>	Watchdog läuft im RSEF-Modus.
Beispiel	<pre>asdWriteUserDebug("Active = %u ReducedSafety = %u \n", wdCheckActive(), wdCheckReducedSafetyMode());</pre>	
		Zu <code>asdWriteUserDebug</code> : siehe Kapitel 13.5
Siehe auch	<code>wdSetSafetyMode</code> , <code>wdCheckActive</code>	

wdCheckActive

Funktion		Prüft, ob Watchdog aktiviert ist.
Syntax	<code>uint8 wdCheckActive(void)</code>	
Beschreibung		Diese Funktion prüft, ob der Watchdog gegenwärtig aktiviert ist. Dies hängt von den Ereigniseinstellungen ab, ebenso davon, ob ein Debugger an die ES1135 angeschlossen ist.
Rückgabewert	<code>false</code>	Watchdog ist gegenwärtig deaktiviert.
	<code>true</code>	Watchdog ist gegenwärtig aktiviert.
Beispiel	<pre>asdWriteUserDebug("Active = %u ReducedSafety = %u \n", wdCheckActive(), wdCheckReducedSafetyMode());</pre>	
		Zu <code>asdWriteUserDebug</code> : siehe Kapitel 13.5
Siehe auch	<code>wdSetSafetyMode</code> , <code>wdSetEvent</code> , <code>wdCheckReducedSafetyMode</code>	

13.4 API-Funktionen (ES1135-LEDs)

Der Simulationscontroller ES1135 hat drei konfigurierbare LEDs. Diese sind in Kapitel 6.2.3 kurz beschrieben. Folgende Schnittstellen zu den LEDs stehen zur Verfügung.

userLed[n]On

Funktion	Schaltet LED [n] ein.
Syntax	<code>void userLed1On(void)</code> <code>void userLed2On(void)</code> <code>void userLed3On(void)</code>
Beschreibung	Diese Funktionen schalten die jeweiligen LEDs ein.
Beispiel	<code>userLed1On();</code>
Siehe auch	<code>userLed[n]Off</code> , <code>userLed[n]Toggle</code>

userLed[n]Off

Funktion	Schaltet LED [n] aus.
Syntax	<code>void userLed1Off(void)</code> <code>void userLed2Off(void)</code> <code>void userLed3Off(void)</code>
Beschreibung	Diese Funktionen schalten die jeweiligen LEDs aus.
Beispiel	<code>userLed1Off();</code>
Siehe auch	<code>userLed[n]On</code> <code>userLed[n]Toggle</code>

userLed[n]Toggle

Funktion	Schaltet LED [n] um.
Syntax	<code>void userLed1Toggle(void)</code> <code>void userLed2Toggle(void)</code> <code>void userLed3Toggle(void)</code>
Beschreibung	Diese Funktionen schalten die jeweiligen LEDs um.
Beispiel	<code>userLed1Toggle();</code>
Siehe auch	<code>userLed[n]Off</code> <code>userLed[n]On</code>

13.5 Sonstige API-Funktionen

Einige weitere API-Funktionen stehen zur Verfügung.

asdWriteUserError

Funktion	Schreibt einen Kommentar in das ASCET-Monitorfenster.
Syntax	Entspricht der ANSI-C-Funktion <code>printf</code> .
Beschreibung	Diese Funktion zeigt vom Anwender definierte Meldungen im ASCET-Monitorfenster an.
Beispiel	<pre>uint8 number = 1; asdWriteUserError("Example %u \n", number);</pre>
Siehe auch	<code>asdWriteUserDebug</code>

asdWriteUserDebug

Funktion	Schreibt einen Kommentar in das Target Debugger-Fenster von ASCET.
Syntax	Entspricht der ANSI-C-Funktion <code>printf</code> .
Beschreibung	Diese Funktion zeigt vom Anwender definierte Meldungen im Target Debugger-Fenster von ASCET an.
Beispiel	<pre>uint8 number = 1; asdWriteUserDebug("Example %u \n", number);</pre>
Siehe auch	<code>asdWriteUserError</code>

Index

A

ActivateTask 408
API-Funktionen (ERCOSEK)
 siehe auch *Service-routinen (ERCOSEK)*
API-Funktionen (LEDs)
 siehe auch *Service-routinen (LEDs)*
API-Funktionen (NVRAM)
 siehe auch *Service-routinen (NVRAM)*
API-Funktionen (sonst.)
 siehe auch *Service-routinen (sonst.)*
API-Funktionen (Watchdog)
 siehe auch *Service-routinen (Watchdog)*
Application Mode
 siehe *Betriebsmodus*
ASCET 15, 86
ASCET Rapid Prototyping 15
ASCET-Optionen
 Register "Hardware" 16
`ascetsd.ini` 29
`asdWriteUserDebug` 430
`asdWriteUserError` 430
Automatic Mapping 172

B

Betriebsmodus 406
Bypass Labels 185
Bypass-Offset 137

C

CAN Bypass Protokoll CBP 180
CAN-Bypass
 Hardwarekonfiguration 182
CAN-Bypass Device
 Globals 183
 Groups 186
 Signals 188
CAN-CTRL Subsystem
 Globals 166
CAN-IO Device
 Globals 169
 Groups 175
 Signals 177
CBP 180

Compiler 28
 Diab Data 27
 eigenen ~ verwenden 23
 GNU Cross ~ 24

D

DeclareAppMode 407
DeclareTask 408
Deklarationen 407, 408
Diab Data Compiler 27
DIO Device
 Globals 236
 Groups 237
 Signals 238
DisableAllInterrupts 411
DISTAB-Verfahren 131
dT 33
dT (delta t) 412

E

EnableInterrupt 411
ERCOSEK
 API-Funktionen 405
ERCOS^{EK} 11
ES1000 11
 TCP/IP-Protokoll 72
ES1120 72
ES1130 72
 dT 33
ES1135 72
 Besonderheiten 73
 dT 33
 LED-API-Funktionen 429
 LEDs 85
 NVRAM 73
 Watchdog 83
ES1135-LED Device
 Globals 144
 Groups 145
 Signals 145
ES1200 146
 Einbindung 147
ES1201-ETK 146
ES1201-ETK Subsystem
 Globals 146
ES1222-CAN 164, 180

ES1222-CAN Subsystem
 Globals 165
ES1223-LIN 189
ES1223-LIN/CAN Subsystem
 Globals 190
ES1231.1-ETK 196
ES1231-ETK Subsystem
 Globals 197
ES1232-ETK 199
ES1232-ETK Subsystem
 Globals 199
ES1300-AD 219
ES1300-AD Device
 Globals 219
 Groups 221
 Signals 222
ES1301-AD 222
ES1301-AD Device
 Globals 223
 Groups 224
 Signals 225
ES1303-AD 225
ES1303-AD Device
 Globals 226
 Groups 229
 Signals 231
ES1310-DA 231
ES1310-DA Device
 Globals 232
 Groups 233
 Signals 234
ES1320-CB (DIO) 234
ES1320-CB Subsystem
 Globals 235
ES1325-DIO 238
ES1325-DIO Subsystem
 Globals 239
ES1325-Input Device
 Globals 243
 Groups 247
 Signals 253
ES1325-LED Device
 Globals 259
 Groups 260
 Signals 261

- ES1325-Output Device
 - Globals 254
 - Groups 255
 - Signals 258
 - ES1330-PWM 261
 - ES1330-PWM Subsystem
 - Globals 262
 - ETAS Netzwerk 389
 - Adressierung 390
 - DHCP 391
 - Fehlersuche 401
 - Hardware-Anbindung 18
 - manuelle Adressierung 391
 - Network Manager 393
 - Netzwerkadapter konfigurieren 396
 - reservierte Adressen 395
 - Verwendung aktivieren 18
 - Ethernet-Schnittstelle 21
 - ETK-Bypass 127
 - ASCET Projekt 129
 - Datenaustausch 131
 - DISTAB-Verfahren 131
 - Funktionsweise 130
 - Hardwarekonfiguration 128
 - ETK-BYPASS Device
 - Globals 149
 - Groups 158
 - Signals 161
 - ETK-BYPASS Device (ES1231)
 - Signals 198
 - ETK-BYPASS-ADV Device
 - Globals 211
 - Groups 213
 - Signals 217
 - ETK-CTRL Subsystem
 - Globals 147
 - ETK-CTRL-ADV Subsystem
 - Globals 206
 - ETK-CTRL-BAS Subsystem
 - alte Projekte verwenden 201
 - altes Projekt mit 100 MBit/s 203
 - altes Projekt mit 8 MBit/s 202
 - Globals 200
 - Experiment
 - Online-- durchführen 37–43
 - Experimental-Target 11
 - Experimentieren mit ASCET 35–44
 - C-Code Debugger 43
 - Experiment beenden 42
 - Experiment einrichten 40
 - Experiment starten 40
 - Messung beenden 42
 - Messung starten 40
 - Standalone-Modus 43
 - Experimentieren mit INCA 44–54
 - Arbeitsumgebung auswählen 47
 - Back-Animation 51
 - Gerät auswählen 49
 - INCA Datenbankpfad 45
 - Projekt auswählen 49
 - Transfer aufrufen 44
 - Transfer starten 51
 - Experimentieren mit INTECRIO 54–66
 - Arbeitsumgebung auswählen 59
 - ASCET-Projekt erstellen 54
 - Back-Animation 64
 - Experiment starten 63
 - Fenster „INTEC RIO Project Transfer“ 55
 - INTECRIO-Build-Prozess auswählen 60
 - INTECRIO-Version auswählen 58
 - Option "Ignore internally connected messages" 57
 - Pfad f. Dateien einstellen 58
 - Systemprojekt auswählen 59
 - Transfer aufrufen 55
 - Transfer durchführen 61
 - EXPORT Unterverzeichnis 11
- ## G
- GetDeltaT 412
 - GetSystemTime 409
 - GetSystemTimeHigh 410
 - GetSystemTimeLow 410
 - GNU Cross Compiler 24
- ## H
- Hardware Configuration Editor
 - siehe *HWC-Editor*
 - Hardware Configuration Modul 68

- Hardware-Anbindung
 - mit ETAS Network Manager 18
 - ohne ETAS Network Manager 21
- Hardware-Auswahlfenster 19
 - manuell öffnen 19
- HWC-Editor 69, 89–122
 - Bedienelemente 90
 - Edit-Modus 69
 - Menü „Edit“ 99
 - Menü „Extras“ 105
 - Menü „File“ 95
 - Menü „View“ 104
 - Register „Globals“ 114
 - Register „Groups“ 117
 - Register „Mappings“ 120
 - Register „Signals“ 118
 - Schaltflächen 90
- HWC-Item 143
- HWC-Modul 68
 - Codegenerierung 69

I

- INCA
 - siehe *Experimentieren mit INCA*
- Installationsprogramm 11
- INTECRIO
 - siehe *Experimentieren mit INTECRIO*
- Interrupts
 - Freigeben 411
 - Sperrern 411
- Items
 - implementierte 143

L

- LIN-CTRL Subsystem
 - Globals 190
- LIN-IO Device
 - Globals 192
 - Groups 194
 - Signals 195

N

- Service-routinen (LEDs)
 - userLed 429
- userLed 429

- Netzwerkadapter konfigurieren 396
 - in DHCP-Umgebung 400
 - mit fester IP-Adresse 398
- Netzwerk-konfiguration
 - s. *ETAS Netzwerk*
- Non-volatile RAM
 - s. *NVRAM*
- NVRAM 73
 - Aktualisierung v. NV-Variablen 76
 - API-Funktionen 413
 - Datenkonsistenz 77
 - defekter Inhalt 78
 - Grundlagen 73
 - Hardware-Unterstützung 74
 - hohe Konsistenz 77
 - Inhalt löschen 77
 - Initialisierung v. NV-Variablen 76
 - keine Konsistenz 77
 - modell-kontrollierte Konsistenz 78
 - niedrige Konsistenz 77
 - NV-Identifizier 74
 - Tips 82
- nvrAmCheckForAutoUpdate 416
- nvrAmCheckForInitializedVars 419
- nvrAmCheckRunningUpdate 419
- nvrAmClear 420
- NVRAM-Cockpit 79
 - arbeiten mit ~ 79
- nvrAmDisableAutoUpdate 416
- nvrAmEnableAutoUpdate 416
- nvrAmGetConsistencyLevel 415
- nvrAmGetUpdateAgeUs 420
- nvrAmGetUpdateInterval 414
- nvrAmInitModelVars 413
- nvrAmManualUpdateBackground 417
- nvrAmManualUpdateBlocked 418
- nvrAmManualUpdateExit 417
- nvrAmSetConsistencyLevel 415
- nvrAmSetUpdateInterval 414

O

- Off 429
- On 429

- Online-Experiment
 - durchführen 37–43
 - Experimentierumgebung öffnen 39
 - Hardware auswählen 38
 - Standalone 43
 - starten 37

P

- PowerPC-Unterverzeichnis 15
- PPC-Modul ES1130 72
- PPC-Modul ES1135 72
- PWM-COUNTER Device
 - Globals 263
 - Groups 266
 - Signals 266

R

- RTIO-Codegenerierung 123–126
 - HWC-Modul 123
 - Prozessreihenfolge 126
- RTIO-Package 67
 - Architektur 67

S

- serielle ETks 210
 - indirekte Übertragung 216
- Serviceroutinen (ERCOSEK)
 - ActivateTask 408
 - DeclareAppMode 407
 - DeclareTask 408
 - DisableAllInterrupts 411
 - EnableInterrupt 411
 - GetDeltaT 412
 - GetSystemTime 409
 - GetSystemTimeHigh 410
 - GetSystemTimeLow 410
 - SetNextAppMode 407
- Serviceroutinen (NVRAM)
 - nvrAmCheckForAutoUpdate 416
 - nvrAmCheckForInitializedVars 419
 - nvrAmCheckRunningUpdate 419
 - nvrAmClear 420
 - nvrAmDisableAutoUpdate 416
 - nvrAmEnableAutoUpdate 416
 - nvrAmGetConsistencyLevel 415
 - nvrAmGetUpdateAgeUs 420
 - nvrAmGetUpdateInterval 414

- nvrAmInitModelVars 413
- nvrAmManualUpdateBackground 417
- nvrAmManualUpdateBlocked 418
- nvrAmManualUpdateExit 417
- nvrAmSetConsistencyLevel 415
- nvrAmSetUpdateInterval 414
- Serviceroutinen (sonst.)
 - asdWriteUserDebug 430
 - asdWriteUserError 430
- Serviceroutinen (Watchdog)
 - wdCheckActive 428
 - wdCheckReducedSafetyMode 427
 - wdDisableAutoService 425
 - wdEnableAutoService 425
 - wdIntAck 427
 - wdIntDisable 426
 - wdIntEnable 426
 - wdIntPend 426
 - wdService 425
 - wdSetEvent 424
 - wdSetPeriod 423
 - wdSetReducedSafetyMode 422
 - wdSetSafetyMode 421
- SetNextAppMode 407
- Steuereinheit ES1120 72
- Systempfad 86

T

- Target
 - PowerPC 15
 - Schnittstellen einrichten (mit ETAS Network Manager) 18
 - Schnittstellen einrichten (ohne Network Manager) 21
- Target ES1130 23
- Target ES1135 23
- target.ini 21
- Targetverzeichnis 15
- Task
 - Aktivierung 408
- TCP/IP-Protokoll 72
- Toggle 429

U

- Umschalten
 - Betriebsmodus 407

Umwandeln alter Projekte 31

W

- Watchdog 83
 - API-Funktionen 420
 - Interruptsteuerung 84
 - Modus 83
 - Periode 83
 - Service 83
 - Service-Register 83
- wdCheckActive 428
- wdCheckReducedSafetyMode 427
- wdDisableAutoService 425
- wdEnableAutoService 425
- wdIntAck 427
- wdIntDisable 426
- wdIntEnable 426
- wdIntPend 426
- wdService 425
- wdSetEvent 424
- wdSetPeriod 423
- wdSetReducedSafetyMode 422
- wdSetSafetyMode 421

Z

- Zeit
 - Systemzeit 409