

API (C-Code-Modul) für ES5398.1 Fault Insertion Board Dokumentation



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 Einzellizenz geliefert. Benutzung und Vervielfältigung ist nur in Übereinstimmung mit den vertraglichen Abmachungen gestattet.

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

© **Copyright 2019** ETAS GmbH, Stuttgart

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

V1.0.0 R08 DE - 05.2019

Inhalt

1	Einführung	5
1.1	Über dieses Handbuch	6
1.1.1	Benutzerprofil	6
1.1.2	Umgang mit dem Handbuch	6
2	Integration des ES5398.1 C-Code-Moduls in ein LABCAR-OPERATOR-Projekt	9
2.1	Neues Projekt mit ES5398.1 oder ES4440.1/2 bauen	9
3	API-Beschreibung	14
3.1	Programmabläufe	14
3.1.1	Einfachfehler	14
3.1.2	Mehrfachfehler in Stand-Alone-Anwendungen	15
3.1.3	Einfachfehler in Master/Slave Anwendungen	16
3.1.4	Mehrfachfehler in Master/Slave-Anwendungen	20
3.2	Detaillierte Beschreibung der Befehle	21
3.2.1	Allgemeine Befehlsstruktur	22
3.2.2	Definitionen für alle Funktionen	23
3.2.3	Fehlercodes	24
3.2.4	Open_Load	27
3.2.5	Open_Load_realtime	28
3.2.6	ShortCut_xUBATTy_20A	29
3.2.7	ShortCut_xUBATTy_20A_realtime	30
3.2.8	Pin2PinFirstChWithoutLoad	31
3.2.9	Pin2PinSecondChannelWithoutLoad	32
3.2.10	Pin2PinFirstChRealtimeWithLoad	33
3.2.11	Pin2PinSecondChRealtimeWithLoad	34
3.2.12	Reset_all_errors	35
3.2.13	Activate_relay	36
3.2.14	Activate_realtime_switch	38

4 ETAS Kontaktinformation	39
Index	41

1 Einführung

Die ES5398.1 ist eine Einsteckkarte für das ES5300 Systemgehäuse, welche zur Nachbildung von elektrischen Fehlerfällen dient. Diese Nachbildung von Fehlerfällen wird zum Test der Diagnose-Funktionen eines Steuergerätes verwendet.

Über LABCAR-OPERATOR steht ein C-Code-Modul für die ES5398.1 zur Verfügung. In diesem Handbuch wird die API des C-Code-Moduls für die ES5398.1 beschrieben.

Das C-Code-Modul kann sowohl für die ES5398.1 als auch für die ES4440.1/2 verwendet werden.

Das ES5398.1 Fault Insertion Board ist über einen RTOS-Treiber mit dem RTPC zu verbinden.

Hinweis

Die aktuelle ES5398 RTOS Treiber API Dokumentation finden Sie direkt auf dem RTPC unter `rtpc/api` (siehe Screenshot).

Name	Last modified	Size	Description
Parent Directory		-	
HEADER.html	2017-05-11 07:18	438	
canlib.html	2017-01-16 08:57	25K	
canlib_protocol_template/	2017-01-16 08:59	-	
es4440_cmd.html	2017-06-09 08:29	3.8K	
es5398_api.html	2017-08-28 10:57	8.0K	
ixxat_can.html	2017-01-16 08:58	30K	
ixxat_lin.html	2017-01-16 08:58	33K	
j1939.html	2017-01-16 08:57	12K	
libusb.html	2017-06-09 08:29	18K	
linux-error-codes.txt	2017-01-16 08:57	7.0K	
model-data-exchange.html	2017-06-09 08:29	24K	
netfilter.html	2017-06-09 08:29	15K	
nonrtudp.html	2017-06-09 08:29	5.0K	
rs232.html	2017-06-09 08:29	6.5K	
rtos_comm.html	2017-01-16 08:57	16K	
rtos_extension.html	2017-01-16 08:57	5.7K	
rtos_extension_template/	2017-06-09 08:29	-	
rtplugins-intro-doc.html	2017-01-16 08:57	7.1K	
rtudp.html	2017-06-09 08:29	15K	
tcpclient.html	2017-06-09 08:29	8.2K	
trigger-task-introduction.html	2017-01-16 08:58	10K	
tys_rs232.html	2017-06-09 08:29	3.7K	
udp_serv.html	2017-06-09 08:29	4.3K	

Apache/2.4.25 (Debian) Server at rtpc Port 80

1.1 Über dieses Handbuch

Dieses Handbuch besteht aus den folgenden Kapiteln:

- „Einführung“ auf Seite 5
Dieses Kapitel
- „Integration des ES5398.1 C-Code-Moduls in ein LABCAR-OPERATOR-Projekt“ auf Seite 9
In diesem Kapitel wird die Integration eines C-Code-Moduls in ein LAB-CAR-OPERATOR-Projekt erklärt. Das C-Code-Modul dient zur Ansteuerung der ES5398.1.
- „API-Beschreibung“ auf Seite 14
In diesem Kapitel werden mögliche Programmabläufe für die ES5398.1 dargestellt.

1.1.1 Benutzerprofil

Dieses Handbuch richtet sich an Fachpersonal in den Bereichen Entwicklung und Test von Kfz-Steuergeräten. Fachwissen im Bereich Mess- und Steuergerätetechnik wird vorausgesetzt.

1.1.2 Umgang mit dem Handbuch

Darstellung von Information

Alle vom Anwender auszuführenden Tätigkeiten werden in einem sogenannten „Use-Case“-Format dargestellt. D.h., dass das zu erreichende Ziel zuerst in der Titelzeile kurz definiert wird, und die jeweiligen Schritte, die notwendig sind, um dieses Ziel zu erreichen, dann in einer Liste aufgeführt werden. Die Darstellung sieht wie folgt aus:

Zieldefinition

eventuelle Vorabinformation...

1. Schritt 1
eventuelle Erläuterung zu Schritt 1...
2. Schritt 2
eventuelle Erläuterung zu Schritt 2...
3. Schritt 3
eventuelle Erläuterung zu Schritt 3...

eventuelle abschließende Bemerkungen...

konkretes Beispiel:

Erstellen einer neuen Datei

Vor dem Erstellen einer neuen Datei darf keine andere geöffnet sein.

1. Wählen Sie **Datei** → **Neu**.
Die Dialogbox „Datei Erstellen“ erscheint.

2. Geben Sie den Namen für die Datei im Feld „Dateiname“ ein.

Der Dateiname darf nicht mehr als 8 Zeichen lang sein.

3. Klicken Sie **OK**.

Die neue Datei wird erstellt und unter dem von ihnen angegebenen Namen abgelegt. Sie können nun mit der Datei arbeiten.

Typografische Konventionen

Folgende typografischen Konventionen werden verwendet:

Wählen Sie Datei → Öffnen .	Menübefehle werden fett/blau dargestellt.
Klicken Sie OK .	Schaltflächen werden fett/blau dargestellt.
Drücken Sie <EINGABE>.	Tastaturbefehle werden in spitzen Klammern in Kapitälchen dargestellt.
Das Dialogfenster „Datei öffnen“ erscheint.	Namen von Programmfenstern, Dialogfenstern, Feldern u.ä. werden in Anführungszeichen gesetzt.
Wählen Sie die Datei <code>setup.exe</code> aus.	Text in Auswahllisten, Programmcode, sowie Pfad- und Dateinamen werden in der Schriftart <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

Wichtige Hinweise für den Anwender werden so dargestellt:

Hinweis

Wichtiger Hinweis für den Anwender

2 Integration des ES5398.1 C-Code-Moduls in ein LABCAR-OPERATOR-Projekt

In diesem Kapitel wird die Integration eines C-Code-Moduls in ein LABCAR-OPERATOR-Projekt erklärt. Das C-Code-Modul dient zur Ansteuerung der ES5398.1. Alle zur Konfiguration und Ansteuerung der ES5398.1 benötigten Source-Files werden auf der ES5398.1 Software Integration DVD (F-00K-111-293) zur Verfügung gestellt.

Hinweis

Die ES5398.1 wird ab LABCAR-OPERATOR V5.4.4 unterstützt.

Das C-Code-Modul für die ES5398.1 kann auch für die ES4440.1/2 verwendet werden.

Die benötigten Source-Files finden Sie im Unterordner `ES5398 Software Integration\ES4440Access`.

2.1 Neues Projekt mit ES5398.1 oder ES4440.1/2 bauen

Um das C-Code-Modul in Ihr LABCAR-OPERATOR-Projekt zu integrieren und für die ES5398.1 oder ES4440.1/2 zu konfigurieren, folgen Sie den unten beschriebenen Schritten.

Integrieren des C-Code-Moduls für die ES5398.1 / ES4440.1/2

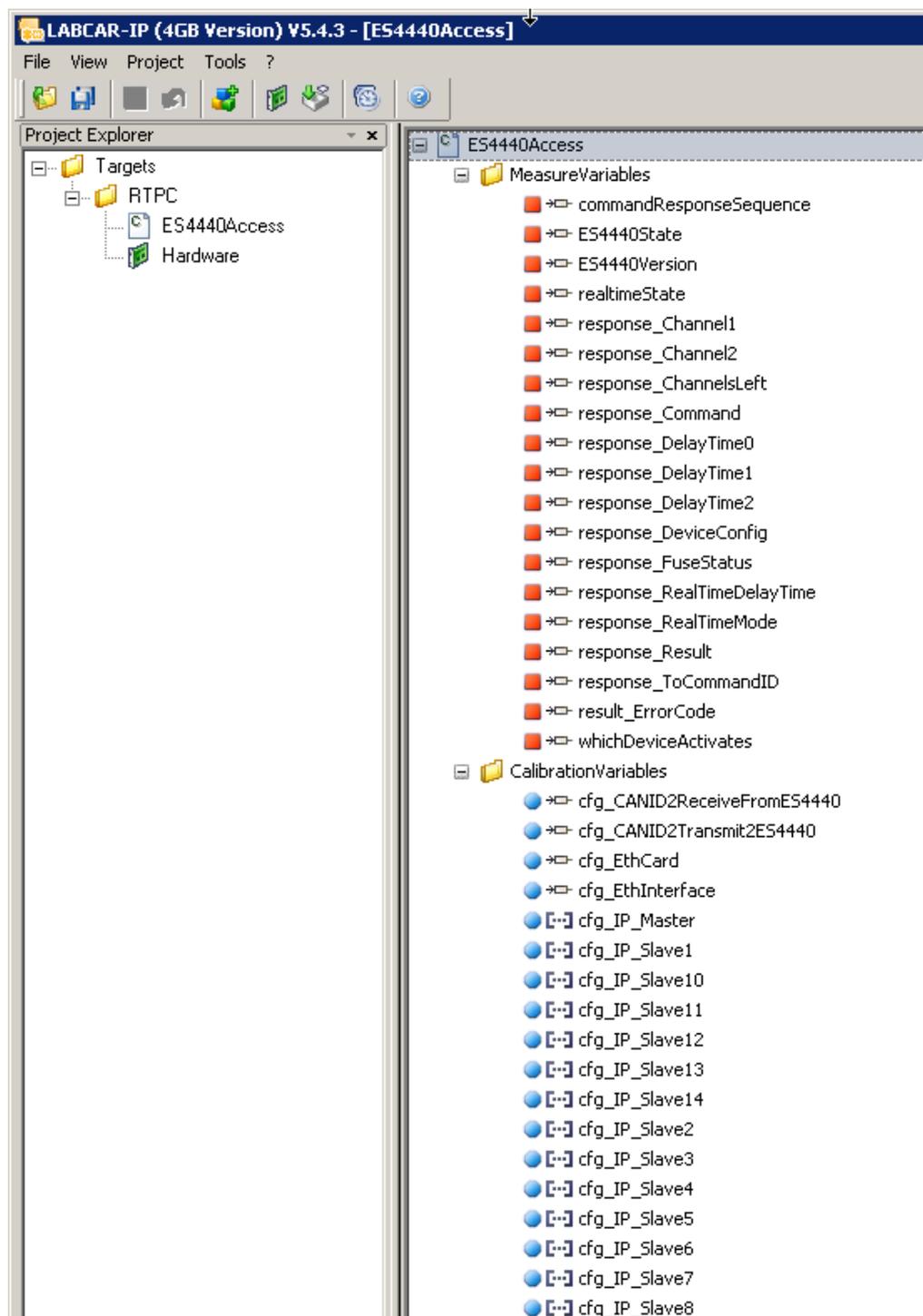
1. Öffnen Sie LABCAR-OPERATOR
2. Wählen Sie den Menüpunkt **File** → **New Project**.
3. Wählen Sie ein LABCAR-OPERATOR 5.x-Projekt mit Default-Target-Name "RTPC" und "No model".
4. Speichern Sie das neue Projekt.
5. Integrieren Sie das C-Code-Modul in das Projekt. Klicken Sie dazu auf die Schaltfläche **Add Module** → **Add C-Code Module** → **Use existing-Module (will be copied)**.
6. Klicken Sie **Browse**.
7. Wählen Sie das File `ES4440Access.lmd` aus dem Unterverzeichnis `\ES5398 Software Integration\ES4440Access`.

Hinweis

Der gesamte Module-Ordner mit allen Unterverzeichnissen wird in den Projektordner kopiert.

Konfigurieren des C-Code-Moduls für die ES5398.1 / ES4440.1/.2

1. Doppelklicken Sie den Eintrag **ES4440Access** im "Project Explorer" und klicken Sie **OK**.
Die MeasureVariables und CalibrationVariables des C-Code-Moduls erscheinen im Fenster "ES4440Access":



2. Konfigurieren Sie die folgenden CalibrationVariables :

cfg_EthInterface
 cfg_IP_Master
 cgf_IP_SlaveX

Klicken Sie dazu auf die CalibrationVariable und dann im Fenster "Model" auf **Value**. Der Wert in der Spalte dahinter kann nun angepasst werden.

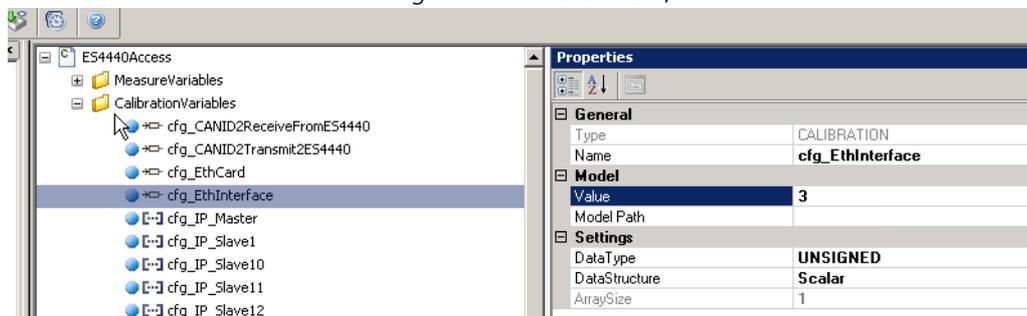
CalibrationVariable	Wert	Bemerkung
cfg_EthInterface	3	Aktiviert Treiber für ES5398.1
	0	Aktiviert Treiber für ES4440.1/2
cfg_IP_Master	Board ID	Globale Board ID: Global Instance auf RTPC-Web-Interface, 4. Wert der IP-Adresse
cgf_IP_SlaveX	Board ID	Globale Board ID: Global Instance auf RTPC-Web-Interface, 4. Wert der IP-Adresse

Diese Variablen bleiben während der Ausführung des Experiments konstant.

3. Um den Treiber für die ES5398.1 oder die ES4440.1/2 zu aktivieren, setzen sie für cfg_EthInterface den Wert auf 3 bzw. auf 0.

Beispiel für die Konfiguration von cfg_EthInterface:

Value = 3 bedeutet Konfiguration für ES5398.1, siehe Tabelle oben.

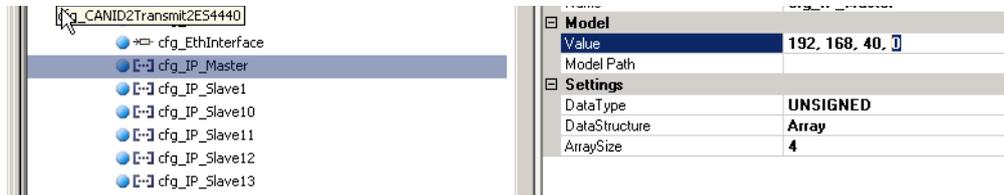


4. Um die globale Board-ID ("Global Instance" auf dem RTPC-Web-Interface) den Master- und Slave-Einheiten zuzuordnen, setzen Sie den vierten Wert der IP-Adresse entsprechend der folgenden Tabelle:

CalibrationVariable	ES5398.1 Global Instance (4. Wert IP-Adresse)	ES4440.1/2 Global Instance (4. Wert IP-Adresse)
cfg_IP_Master	0	40
cgf_IP_Slave1	1	41
cgf_IP_Slave2	2	42
cgf_IP_Slave3	3	43
...

Beispiel für die Konfiguration von cfg_IP_Master:

Der vierte Wert der IP-Adresse ist entscheidend. „0“ entspricht einer ES5398.1-Master-Einsteckkarte mit globaler Board ID „0“ (siehe Tabelle oben).



5. Klicken Sie **Project** → **OS Configuration**.
Vergewissern Sie sich, dass der Prozess "cmod_run_ES4440_ES4440Access" im Standard-Zyklus enthalten ist.
6. Bauen Sie den Real-Time-Code auf dem RTPC durch **Project** → **Build**.

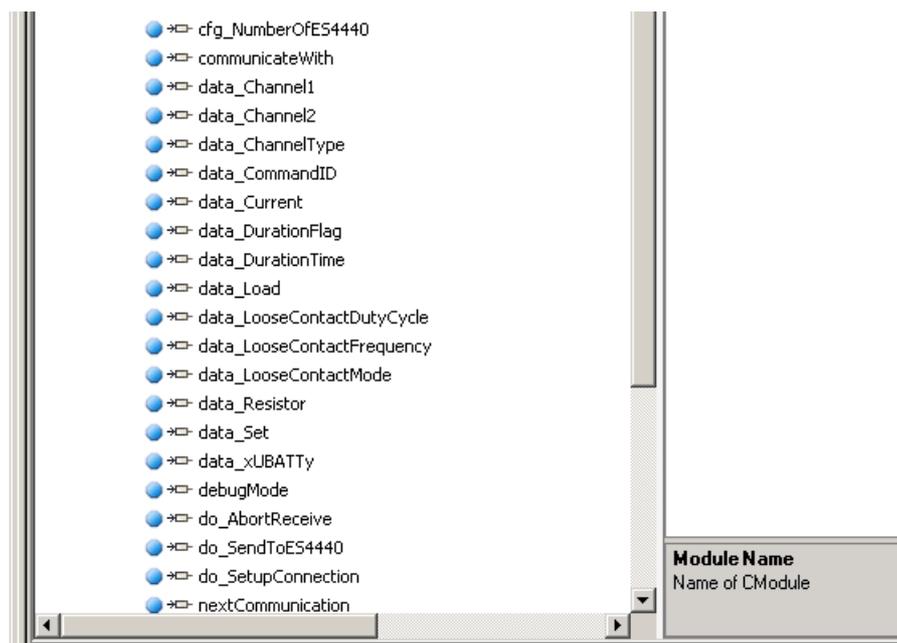
Hinweis

Clean intermediate files wird empfohlen.

Damit ist die Integration des C-Code-Moduls in das LABCAR-OPERATOR-Projekt abgeschlossen.

Hinweis

Das Fenster "ES4440Access" enthält noch weitere CalibrationVariables, die für die ES5398.1 oder die ES4440.1/2 bei der Programmierung für den automatischen Betrieb gesetzt werden müssen. Wie die Werte gesetzt werden können, ist im Verlauf des Kapitels „Detaillierte Beschreibung der Befehle“ auf Seite 21 erklärt.



3 API-Beschreibung

In diesem Kapitel werden mögliche Programmabläufe skizziert und es erfolgt eine Beschreibung der Befehle für das API.

Im Einzelnen enthält dieses Kapitel folgende Informationen:

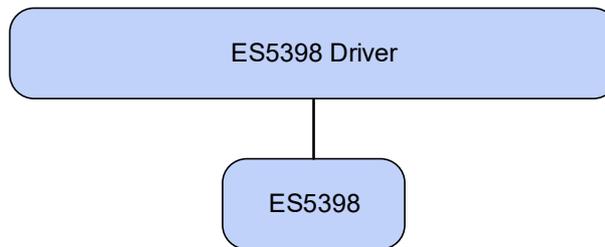
- „Programmabläufe“ auf Seite 14
- „Detaillierte Beschreibung der Befehle“ auf Seite 21

3.1 Programmabläufe

In diesem Kapitel werden mögliche Programmabläufe für die ES5398.1 dargestellt.

3.1.1 Einfachfehler

Die in diesem Abschnitt beschriebenen Programmabläufe gelten für Einfachfehler, die auf einer ES5398.1 ausgeführt werden.



Programmablauf für Fehler, die mit Relais erzeugt werden

1. Fehlerkonfiguration

Hochstromfehler:

- Open_Load
oder
- ShortCut_xUBATTy_20A
oder
- Pin2PinFirstChWithoutLoad / Pin2PinSecondChWithoutLoad

2. Fehleraktivierung

- Activate_relay

3. Zurücksetzen des Fehlers

- Reset_all_errors

Programmablauf für Fehler, die mit MOSFETs realisiert werden

1. Fehlerkonfiguration

- Open_Load_realtime
oder
- ShortCut_xUBATTy_20A_realtime
oder

- Pin2PinFirstChRelatimeWithLoad
oder
- Pin2PinSecondChRealtimewithLoad
oder
- RInline_realtime
oder
- Pullup_Pulldown_xUBATTy_20A_realtime

2. Fehleraktivierung

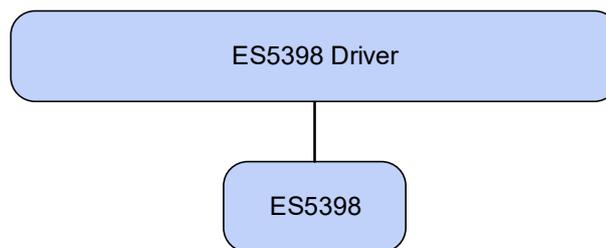
- Activate_realtime_switch

3. Zurücksetzen des Fehlers

- Reset_all_errors

3.1.2 Mehrfachfehler in Stand-Alone-Anwendungen

Die in diesem Abschnitt beschriebenen Programmabläufe gelten für Mehrfachfehler, die auf einer ES5398.1 ausgeführt werden.



Programmablauf für Fehler, die mit Relais erzeugt werden

1. Fehlerkonfiguration

(max. 10 Befehle werden der Reihe nach an die ES5398.1 verschickt)

- Open_Load
und/oder
- ShortCut_xUBATTy_20A

2. Fehleraktivierung

- Activate_relay

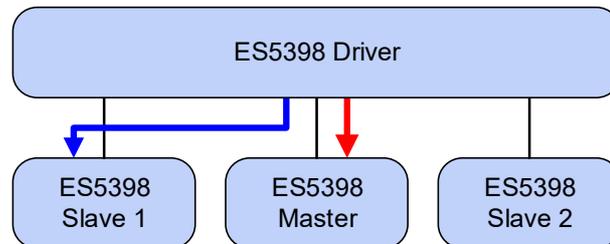
3. Zurücksetzen der Fehler

- Reset_all_errors

3.1.3 Einfachfehler in Master/Slave Anwendungen

Die in diesem Abschnitt beschriebenen Programmabläufe gelten für Einfachfehler, die auf einem ES5398.1 Master/Slave-System ausgeführt werden.

Programmablauf für Fehler, die mit Relais erzeugt werden



1. Fehlerkonfiguration (auf Slave 1)

Max. 10 Hochstromfehler:

- Open_Load
und/oder
- ShortCut_xUBATTy_20A

2. Fehleraktivierung (auf Master)

- Activate_relay

3. Zurücksetzen der Fehler (auf Slave 1) *

- Reset_all_errors

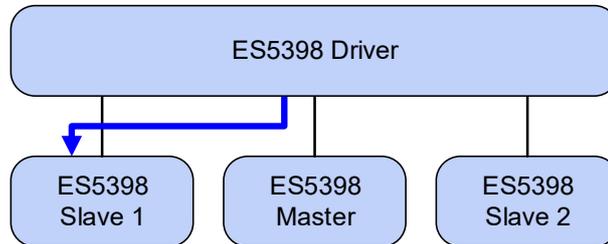
4. Zurücksetzen der Fehler (auf Master) **

- Reset_all_errors

* Der Befehl „Zurücksetzen des Fehlers“ auf einem Slave wird zunächst nur gespeichert.

** Der Befehl „Zurücksetzen des Fehlers“ auf dem Master bewirkt ein synchrones Zurücksetzen der Fehler auf dem Master und allen Slaves, die diesen Befehl zuvor gespeichert haben.

Programmablauf für Fehler, die mit MOSFETs erzeugt werden



1. Fehlerkonfiguration

- `Open_Load_realtime`
oder
- `ShortCut_xUBATTy_20A_realtime`

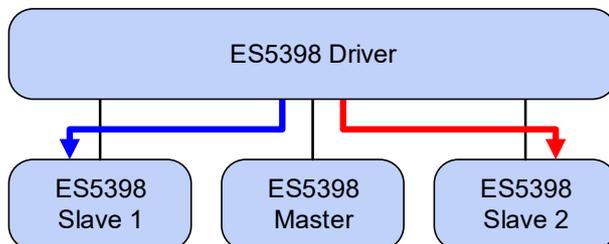
2. Fehleraktivierung

- `Activate_realtime_switch`

3. Zurücksetzen des Fehlers

- `Reset_all_errors`

Sonderfall: Pin-to-Pin-Fehler mit Last

**1. Fehlerkonfiguration für ersten Pin (auf Slave1)**

- `Pin2PinFirstChRelatimeWithLoad`

2. Fehlerkonfiguration für zweiten Pin (auf Slave 2)

- `Pin2PinSecondChRealtimeWithLoad`

3. Fehleraktivierung (auf Slave1)

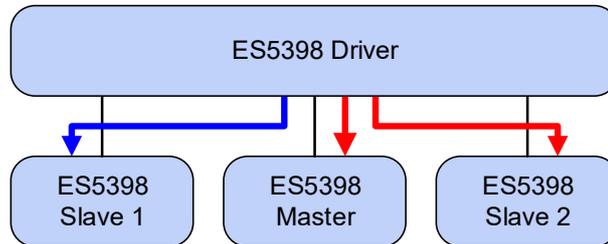
- `Activate_realtime_switch`

4. Zurücksetzen des Fehlers (auf Slave1)

- `Reset_all_errors`

5. Zurücksetzen des Fehlers (auf Slave 2)

- `Reset_all_errors`

Sonderfall: Pin-to-Pin ohne Last

- 1. Fehlerkonfiguration für ersten Pin (auf Slave 1)**
 - `Pin2PinFirstChWithoutLoad`
- 2. Fehlerkonfiguration für zweiten Pin (auf Slave 2)**
 - `Pin2PinSecondChWithoutLoad`
- 3. Fehleraktivierung (auf Master)**
 - `Activate_relay`
- 4. Zurücksetzen des Fehlers (auf Slave 1)**
 - `Reset_all_errors`
- 5. Zurücksetzen des Fehlers (auf Slave 2)**
 - `Reset_all_errors`
- 6. Zurücksetzen des Fehlers (auf Master)**
 - `Reset_all_errors`

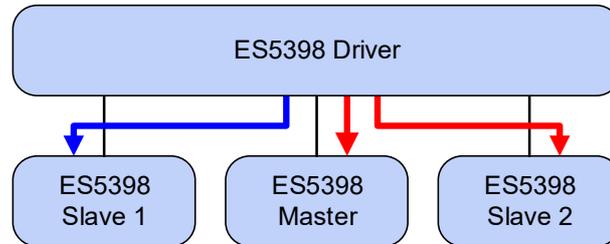
Hinweis

Dieser Fehler wird mit Relais geschaltet - es befindet sich keine Sicherung im Fehlerpfad zwischen Pin 1 und Pin 2!

3.1.4 Mehrfachfehler in Master/Slave-Anwendungen

Die in diesem Abschnitt beschriebenen Programmabläufe gelten für Mehrfachfehler, die auf einem Master/Slave-System ausgeführt werden.

Programmablauf für Fehler, die mit Relais erzeugt werden



1. Fehlerkonfiguration (auf Slave 1)

Max. 10 Hochstromfehler:

- Open_Load
und/oder
- ShortCut_xUBATTy_20A

2. Fehlerkonfiguration (auf Slave 2)

Max. 10 Hochstromfehler:

- Open_Load
und/oder
- ShortCut_xUBATTy_20A

3. Fehleraktivierung (auf Master)

- Activate_relay

4. Zurücksetzen des Fehlers (auf Slave1)

- Reset_all_errors

5. Zurücksetzen des Fehlers (auf Slave 2)

- Reset_all_errors

6. Zurücksetzen des Fehlers (auf Master)

- Reset_all_errors

3.2 Detaillierte Beschreibung der Befehle

In diesem Abschnitt finden Sie die vollständige Syntaxbeschreibung aller Befehle.

Im Einzelnen sind dies:

- „Open_Load“ auf Seite 27
- „Open_Load_realtime“ auf Seite 28
- „ShortCut_xUBATTy_20A“ auf Seite 29
- „ShortCut_xUBATTy_20A_realtime“ auf Seite 30
- „Pin2PinFirstChWithoutLoad“ auf Seite 31
- „Pin2PinSecondChannelWithoutLoad“ auf Seite 32
- „Pin2PinFirstChRealtimeWithLoad“ auf Seite 33
- „Pin2PinSecondChRealtimeWithLoad“ auf Seite 34
- „Reset_all_errors“ auf Seite 35
- „Activate_relay“ auf Seite 36
- „Activate_realtime_switch“ auf Seite 38

3.2.1 Allgemeine Befehlsstruktur

Im Folgenden finden Sie eine Beschreibung der allgemeinen Struktur der Send- und Receive-Botschaften zwischen C-Code-Modul und ES5398.1-Treiber.

Send-Botschaft

COM_command_name(arguments)		
1. Byte	Command ID	<i>Wert</i>
2. Byte	Parameter 0	<i>Wert</i>
3. Byte	Parameter 1	<i>Wert</i>
4. Byte	Parameter 2	<i>Wert</i>
5. Byte	Parameter 3	<i>Wert</i>
6. Byte	Parameter 4	<i>Wert</i>
7. Byte	Parameter 5	<i>Wert</i>
8. Byte	Parameter 6	<i>Wert</i>

Tab. 3-1 Struktur einer Send-Botschaft

Receive-Botschaft

Antwort		
1. Byte	Command ID	<i>Wert</i>
2. Byte	Parameter 0	<i>Wert</i>
3. Byte	Parameter 1	<i>Wert</i>
4. Byte	Parameter 2	<i>Wert</i>
5. Byte	Parameter 3	<i>Wert</i>
6. Byte	Parameter 4	<i>Wert</i>
7. Byte	Parameter 5	<i>Wert</i>
8. Byte	Parameter 6	<i>Wert</i>

Tab. 3-2 Struktur der Receive-Botschaft

Hinweis

Die ES5398.1 verwendet ein Send- / Receive-Protokoll. Send Botschaften werden vom System durch eine zugehörige Receive-Botschaft bestätigt. Neue Send-Botschaften dürfen erst nach Erhalt dieser Receive-Botschaft gesendet werden.

3.2.2 Definitionen für alle Funktionen

Die Informationen in diesem Abschnitt sind für alle Befehle gültig.

Hinweis

Für die CalibrationVariables (siehe "ES4440Access"-Fenster auf Seite 12) sind an den entsprechenden Stellen die zu setzenden Werte angegeben.

Kanalnummern

Die Kanäle werden von 0 - 39 abgezählt.

Im Fenster "ES4440Access" C-Code-Modul entspricht dies den CalibrationVariables data_Channel11 oder data_Channel12.

Definition für Parameter 1 der Send-Botschaft:

Der Parameter 1 (3.Byte) hat bei allen Funktionen denselben Aufbau:

		CalibrationVariable
Bit 0	load load = 1: Fehlersimulation mit Last load = 0: Fehlersimulation ohne Last	data_Load: auf 0 oder 1 setzen
Bit 1	xUBatty xUBatty = 0: +UBatt_A	data_xUBATTy: auf 0, 1 oder 2 setzen
Bit 2	xUBatty = 1: -UBatt	
Bit 3	xUBatty = 2: +UBatt_B	
Bit 4	Nicht verwendet	
Bit 5	set set = 1: Fehler gesetzt set = 0: Fehler zurückgesetzt	data_Set: auf 0 oder 1 setzen
Bit 6	duration_flag duration_flag = 0: Fehler liegt unendlich (d.h. bis zum Reset) an. duration_flag = 1: Fehlerdauer wird von „duration_time“ festgelegt.	data_Duration-Flag: auf 0 oder 1 setzen
Bit 7	nicht verwendet	

3.2.3 Fehlercodes

Fehlercodes werden in der Receive-Message übermittelt und erscheinen in der MeasureVariable `response_Command`.

Fehlercodes, bisher für ES4440.1 gültig, teilweise für ES5398.1

Die Fehlercodes in der folgenden Tabelle sind für ES4440.1/2 gültig. Teilweise gelten sie auch für die ES5398.1.

Fehlercodes, die neu sind und ausschließlich für die ES5398.1 gelten, sind in der darauf folgenden Tabelle aufgelistet (siehe „Neue Fehlercodes für ES5398.1“ auf Seite 25).

- Weiß hinterlegt: für ES5398.1 und ES4440.1/2
- Grau hinterlegt: nur für ES4440.1/2

Ergebnis	Bedeutung
0x0	Befehl OK
0x21	Falscher Parameter für Slave-Adresse (> 16)
0x22	Unbekannter Befehl
0x23	Falscher Datentyp beim Schreiben des Flash
0x24	Falscher Parameter beim LED-Test
0x25	Zu große Zahl in IP-Adresse (muss < 256 sein)
0x26	Falscher Parameter für CAN Baudrate
0x27	Falscher Parameter für CAN-Terminierung
0x28	Falscher Parameter für CAN-Identifiertyp
0x29	Parameter für Kanal der Kaskade zu groß (muss < 15 sein)
0x2a	Falscher Parameter für Widerstandskaskade
0x2b	nicht verwendet
0x2c	Falsche Adresse bei Flash-Lesezugriff (gültige Werte < 513)
0x2d	Falsche Datenlänge bei Flash-Lesezugriff (gültige Werte < 17)
0x2e	Falsche Adresse bei Flash-Schreibzugriff (gültige Werte < 513)
0x2f	Falsche Datenlänge bei Flash-Schreibzugriff (gültige Werte < 17)
0x30	PLD-Fehler
0x31	Fehler EEPROM-Prüfsumme
0x32	CAN-Controller lässt sich nicht ansprechen
0x41	Simulationsbefehl gibt Plausibilitätsfehler zurück
0x42	Referenzrelais wurde nicht erkannt
0x43	„duration_time“ hat nicht den Wert „0xffff“, obwohl Fehler unbefristet anliegen soll.
0x44	Simulationsbefehl wurde nicht erkannt
0x45	Interner Hardwarefehler PLD: Befehl konnte nicht richtig geschaltet werden

Ergebnis	Bedeutung
0x46	Wert von „duration time“ ausserhalb des gültigen Bereichs (1 bis 6000 oder 0xFFFF)
0x47	Letzte Fehlersimulation ist noch aktiv, mit <code>Reset_all_errors</code> löschen
0x48	Max. Anzahl Relais erreicht
0x49	Fehler mit MultiErrorFlag
0x4a	Angegebene Kanalnummer außerhalb des gültigen Bereichs
0x4b	Frequenz oder Tastverhältnis außerhalb des gültigen Bereichs
0x4c	Systemtemperatur > 60 °C
0x4d	Temperatur der Widerstandskaskade > 60 °C
0x4e	Temperatur MOSFET > 60 °C
0x4f	Fühler für Systemtemperatur defekt
0x50	Fühler für Temperatur der Widerstandskaskade defekt
0x51	Fühler für Temperatur der MOSFETs defekt
0x52	Railspannung nicht korrekt (möglicher Kurzschluss)
0x53	Ungültiger Widerstandswert

Neue Fehlercodes für ES5398.1

Einige Fehlercodes sind neu und gelten ausschließlich für die ES5398.1. Sie sind in der folgenden Tabelle aufgelistet:

Ergebnis	Bedeutung	Detail
0x60	ADC Überspannung	“+VBATA exceeds the limit for Relay Errors”
0x61	ADC Überspannung	“+VBATA exceeds the limit for Relay Errors”
0x62	ADC Überspannung	“RAIL1 voltage exceeds the limit for Relay Errors”
0x63	ADC Überspannung	“RAIL2 voltage exceeds the limit for Relay Errors”
0x64	ADC Überspannung	“Ext. resistor voltage exceeds the limit for Relay Errors”
0x65	ADC Überstrom	“Current in failure path exceeds limit”
0x66	MOSFET Über-Temp.	“Realtime switch exceeds limit for temperature”
0x70	“TER Error in SPI to Load-Rail relays”	

Ergebnis	Bedeutung	Detail
0x71	"TER Error in SPI to OpenLoad relays"	
0x72	"TER Error in SPI to ECU-Rail relays"	
0x73	"TER Error in SPI to Error relays"	

3.2.4 Open_Load

Unterbricht eine Leitung zwischen Steuergerät und Last. Dieser Fehler wird mit Relais geschaltet - bis zu zehn Fehler können gleichzeitig geschaltet werden. Der Wert von „channels left“ in der Befehlsantwort gibt an, wieviele Kanäle noch für weitere Fehler zur Verfügung stehen.

Send-Botschaft

Open_Load(channel_nr, duration_flag, set)			CalibrationVariable
1. Byte	Command ID	0x1	data_CommandID
2. Byte	Parameter 0	channel number	data_Channel11
3. Byte	Parameter 1	set, duration_flag (siehe Seite 23)	data_Set, data_Duration- Flag (Werte siehe Seite 23)
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	nicht verwendet	

Receive-Botschaft

Antwort			MeasureVariable
1. Byte	Command ID	0x1	
2. Byte	Parameter 0	channel number	
3. Byte	Parameter 1	channels left	
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	command result	response_Command

Der Parameter 6 command result entspricht der MeasureVariable **response-Command** im "ES4440Access"-Fenster (siehe Seite 10). Das gilt auch im Folgenden.

3.2.5 Open_Load_realtime

Unterbricht eine Leitung zwischen Steuergerät und Last. Dieser Fehler wird mit MOSFETs geschaltet und ist nur als Einfachfehler realisierbar.

Send-Botschaft

Open_Load_realtime (channel_nr, duration_flag)			CalibrationVariable
1. Byte	Command ID	0x2	
2. Byte	Parameter 0	ES5398.1 channel number	
3. Byte	Parameter 1	duration_flag	data_Duration-Flag (Werte siehe Seite 23)
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	nicht verwendet	

Receive-Botschaft

Antwort			MeasureVariable
1. Byte	Command ID	0x2	
2. Byte	Parameter 0	ES5398.1 channel number	
3. Byte	Parameter 1	nicht verwendet	
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	command result	response_Command

3.2.6 ShortCut_xUBATTy_20A

Erzeugt bei einem Hochstromkanal einen Kurzschluss einer Leitung gegen eine Batteriespannung. Dieser Fehler wird mit Relais geschaltet und ist mehrfach realisierbar.

Send-Botschaft

ShortCut_xUBATTy_20A (channel_nr, load, xUBATTy, duration_flag, set)			CalibrationVariable
1. Byte	Command ID	0x3	
2. Byte	Parameter 0	ES5398.1 channel number	
3. Byte	Parameter 1	load, xUBATTy, set, duration_flag)	data_Load, data_xUBATTy, data_Set, data_DurationFlag (Werte siehe Seite 23)
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	nicht verwendet	

Receive-Botschaft

Antwort			MeasureVariable
1. Byte	Command ID	0x3	
2. Byte	Parameter 0	ES5398.1 channel number	
3. Byte	Parameter 1	channels left	
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	command result	responseCommand

3.2.7 ShortCut_xUBATTy_20A_realtime

Erzeugt bei einem Hochstromkanal einen Kurzschluss einer Leitung gegen eine Batteriespannung. Dieser Fehler wird mit MOSFETs geschaltet und ist daher nur einfach realisierbar.

Send-Botschaft

ShortCut_xUBATTy_20A_realtime (channel_nr, load, xUBATTy, duration_flag)			CalibrationValue
1. Byte	Command ID	0x4	
2. Byte	Parameter 0	channel number	
3. Byte	Parameter 1	load, xUBATTy, duration_flag	data_Load, data_xUBATTy, data_Duration-Flag (Werte siehe Seite 23)
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	nicht verwendet	

Receive-Botschaft

Antwort			MeasureVariable
1. Byte	Command ID	0x4	
2. Byte	Parameter 0	channel number	
3. Byte	Parameter 1	nicht verwendet	
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	command result	responseCommand

3.2.8 Pin2PinFirstChWithoutLoad

Definiert die erste Leitung für einen Kurzschluss zwischen zwei Leitungen - die zweite Leitung wird mit dem Befehl „Pin2PinSecondChannelWithoutLoad“ definiert (siehe Seite 32).

Hinweis

Dieser Fehler wird mit Relais geschaltet und ohne Last und ohne Widerstand realisiert. Es befindet sich keine Sicherung zwischen Pin 1 und Pin 2!

Send-Botschaft

Pin2PinFirstChWithoutLoad (channel_nr1, duration_flag)			CalibrationVariable
1. Byte	Command ID	0x5	
2. Byte	Parameter 0	ES5398.1 channel number 1	data_Channel1
3. Byte	Parameter 1	duration_flag	data_Duration-Flag (Werte siehe Seite 23)
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	nicht verwendet	

Receive-Botschaft

Antwort			MeasureVariable
1. Byte	Command ID	0x5	
2. Byte	Parameter 0	ES5398.1 channel number 1	
3. Byte	Parameter 1	nicht verwendet	
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	command result	responseCommand

3.2.9 Pin2PinSecondChannelWithoutLoad

Definiert die zweite Leitung für einen Kurzschluss zwischen zwei Leitungen.

Hinweis

Dieser Fehler wird mit Relais geschaltet und ohne Last und ohne Widerstand realisiert. Es befindet sich keine Sicherung zwischen Pin 1 und Pin 2!

Send-Botschaft

Pin2PinSecondChannelWithoutLoad (channel_nr1, duration_flag)			CalibrationVariable
1. Byte	Command ID	0x6	
2. Byte	Parameter 0	channel number 2	data_Channel2
3. Byte	Parameter 1	duration_flag	data_Duration_Flag (Werte siehe Seite 23)
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	nicht verwendet	

Receive-Botschaft

Antwort			MeasureVariable
1. Byte	Command ID	0x6	
2. Byte	Parameter 0	channel number 2	
3. Byte	Parameter 1	nicht verwendet	
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	command result	responseCommand

3.2.10 Pin2PinFirstChRealtimeWithLoad

Definiert die erste Leitung für einen Kurzschluss zwischen zwei Leitungen. Dieser Fehler ermöglicht die Simulation mit Last und endlichem Widerstand zwischen den beiden Leitungen.

Dieser Fehler wird mit MOSFETs geschaltet.

Send-Botschaft

Pin2PinFirstChRealtimeWithLoad (channel_nr1, duration_flag)			CalibrationVariable
1. Byte	Command ID	0x7	
2. Byte	Parameter 0	channel number 1	data_Channel1
3. Byte	Parameter 1	current, duration_flag (siehe Seite 23)	data_Duration-Flag (Werte siehe Seite 23)
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	Widerstand Byte 0	data_Resistor (Value !=0 verwenden)
6. Byte	Parameter 4	Widerstand Byte 1	data_Resistor (Value !=0 verwenden)
7. Byte	Parameter 5	Widerstand Byte 2	data_Resistor (Value !=0 verwenden)
8. Byte	Parameter 6	Widerstand Byte 3	data_Resistor (Value !=0 verwenden)

Receive-Botschaft

Antwort			MeasureVariable
1. Byte	Command ID	0x7	
2. Byte	Parameter 0	channel number 1	
3. Byte	Parameter 1	nicht verwendet	
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	command result	responseCommand

3.2.11 Pin2PinSecondChRealtimeWithLoad

Definiert die zweite Leitung für einen Kurzschluss zwischen zwei Leitungen. Dieser Fehler ermöglicht die Simulation mit Last und endlichem Widerstand zwischen den beiden Leitungen.

Dieser Fehler wird mit MOSFETs geschaltet.

Send-Botschaft

Pin2PinSecondChRealtimeWithLoad (channel_nr2, duration_flag)			CalibrationVariable
1. Byte	Command ID	0x8	
2. Byte	Parameter 0	channel number 2	data_Channel12
3. Byte	Parameter 1	duration_flag	data_Duration_Flag (Werte siehe Seite 23)
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	nicht verwendet	

Receive-Botschaft

Antwort			MeasureVariable
1. Byte	Command ID	0x8	
2. Byte	Parameter 0	channel number 2	
3. Byte	Parameter 1	nicht verwendet	
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	nicht verwendet	

3.2.12 `Reset_all_errors`

Mit diesem Befehl werden alle Fehler auf einem ES5398.1 zurückgesetzt.

Send-Botschaft

Reset_all_errors ()			CalibrationVariable
1. Byte	Command ID	0x10	
2. Byte	Parameter 0	nicht verwendet	
3. Byte	Parameter 1	nicht verwendet	
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	nicht verwendet	

Receive-Botschaft

Antwort			MeasureVariable
1. Byte	Command ID	0x10	
2. Byte	Parameter 0	nicht verwendet	
3. Byte	Parameter 1	nicht verwendet	
4. Byte	Parameter 2	nicht verwendet	
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	command result	responseCommand

3.2.13 Activate_relay

Mit diesem Befehl wird das Relais für eine bestimmte Zeit geschlossen.

Wenn im vorhergehenden Fehlerbefehl „duration_flag“ (Bit 6 im 3. Byte) gesetzt ist (= 1), kann „duration_time“ zwischen 1 ms und 5 s gewählt werden. Ist „duration_flag“ = 0, muss „duration_time“ = -1 oder = 65535 (0xFFFF) gewählt werden.

Hinweis

Beim Setzen von Mehrfachfehlern müssen die Parameter „duration_flag“ aller Fehler denselben Wert besitzen!

In der Befehlsantwort werden die am Referenzrelais gemessenen Schaltzeiten übermittelt.

Send-Botschaft

Activate_relay (duration_time)			CalibrationVariable
1. Byte	Command ID	0x12	
2. Byte	Parameter 0	nicht verwendet	
3. Byte	Parameter 1	duration_time in ms (Byte 0)	data_Duration- Time (Werte siehe unten)
4. Byte	Parameter 2	duration_time in ms (Byte 1)	data_Duration- Time (Werte siehe unten)
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	nicht verwendet	

Werte für **data_DurationTime**:

- Wenn data_DurationFlag gleich 0 war, setzen Sie den Wert auf -1.
- Sonst Geben Sie die Zeit in ms an, wie im Folgenden beschrieben:
- Der Parameter „duration_time“ kann in Schritten von 20 ms spezifiziert werden - der kleinstmögliche Wert beträgt 20 ms, der größtmögliche 5000 ms.
- Endlosfehler werden mit dem Wert 0xFFFF erzeugt.

Der Parameter „channel_type“ kann folgende Werte annehmen:

- channel_type = 0: Stromkanal
- channel_type = 1: Spannungskanal

*Receive-Botschaft***Antwort**

1. Byte	Command ID	0x12
2. Byte	Parameter 0	delay_time0 NO-Kontakt* 20 A schliesst (in 100 µs Schritten)
3. Byte	Parameter 1	delay_time1 NO-Kontakt* 20 A schliesst (in 100 µs Schritten)
4. Byte	Parameter 2	delay_time0 NC-Kontakt** 20 A öffnet (in 100 µs Schritten)
5. Byte	Parameter 3	delay_time1 NC-Kontakt** 20 A öffnet (in 100 µs Schritten)
6. Byte	Parameter 4	delay_time0 NC-Kontakt** 400 V schliesst (in 100 µs Schritten)
7. Byte	Parameter 5	NC-Kontakt** 400 V schliesst (in 100 µs Schritten)
8. Byte	Parameter 6	command result (responseCommand)

* NO = normally open = Schließer

** NC= normally closed = Öffner

3.2.14 Activate_realtime_switch

Mit diesem Befehl wird ein mit MOSFETs geschalteter Fehler für eine bestimmte Zeit geschlossen.

Wenn im vorhergehenden Fehlerbefehl „duration_flag“ (Bit 6 im 3. Byte) gesetzt ist (= 1), kann „duration_time“ zwischen 1 ms und 5 s gewählt werden. Ist „duration_flag“ = 0, muss „duration_time“ = -1 oder = 65535 (0xFFFF) gewählt werden.

Send-Botschaft

Activate_realtime_switch (mode, duration_time, dutycycle, frequency)			CalibrationVariable
1. Byte	Command ID	0x13	
2. Byte	Parameter 0	mode	
3. Byte	Parameter 1	duration_time (Byte 0)	data_Duration-Time
4. Byte	Parameter 2	duration_time (Byte 1)	data_Duration-Time
5. Byte	Parameter 3	nicht verwendet	
6. Byte	Parameter 4	nicht verwendet	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	nicht verwendet	

Der Parameter „duration_time“ kann in Schritten von 1 ms spezifiziert werden - der kleinstmögliche Wert beträgt 1 ms, der grösstmögliche 5000 ms. Endlosfehler werden mit dem Wert 0xFFFF erzeugt.

Der Parameter „mode“ kann folgenden Werte annehmen:

- mode = 0: Statischer Fehler, dessen Dauer durch „duration_time“ definiert ist.

Receive-Botschaft

Antwort			MeasureVariable
1. Byte	Command ID	0x13	
2. Byte	Parameter 0	mode	
3. Byte	Parameter 1	duration_time (Byte 0)	
4. Byte	Parameter 2	duration_time (Byte 1)	
5. Byte	Parameter 3	duration_time (Byte 2)	
6. Byte	Parameter 4	duration_time (Byte 3)	
7. Byte	Parameter 5	nicht verwendet	
8. Byte	Parameter 6	command result	responseCommand

4 **ETAS Kontaktinformation**

ETAS Hauptsitz

ETAS GmbH

Borsigstraße 24

70469 Stuttgart

Deutschland

Telefon: +49 711 3423-0

Telefax: +49 711 3423-2106

WWW: www.etas.com

ETAS Regionalgesellschaften und Technischer Support

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

ETAS Regionalgesellschaften WWW: www.etas.com/de/contact.php

ETAS Technischer Support WWW: www.etas.com/de/hotlines.php

Index

A

Activate_realtime_switch 38
Activate_relay 36
Allgemeine Befehlsstruktur 22

B

Bedienung
 Konventionen 7
 Use-Case 6
Benutzerprofil 6

C

channels left 27
CurrentMeasurement 38

E

Einführung 5
ETAS Kontaktinformation 39

O

Open_Load 27
Open_Load_realtime 28

P

Pin2PinFirstChRealtimeWithLoad 33
Pin2PinFirstChWithoutLoad 31
Pin2PinSecondChannelWithoutLoad
 32
Pin2PinSecondChRealtimeWithLoad
 34

Pullup_Pulldown_xUBATTy_20A_real-
 time 34

R

Reset_all_errors 35

S

ShortCut_xUBATTy_20A 29
ShortCut_xUBATTy_20A_realtime 30

