

API (C-Code Module) for ES5398.1 Fault Insertion Board Documentation



Copyright

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

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

© **Copyright 2019** ETAS GmbH, Stuttgart

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

V1.0.0 R08 EN - 05.2019

Contents

1	Introduction	5
1.1	About this Manual	6
1.1.1	User Profile	6
1.1.2	Using this Manual	6
2	Integration of the C-Code Module for ES5398.1 into a LABCAR-OPERATOR Project	9
2.1	Building a New Project with ES5398.1	9
3	API Description	14
3.1	Program Sequences	14
3.1.1	Single Errors	14
3.1.2	Multiple Errors in Standalone Applications	15
3.1.3	Single Errors in Master/Slave Applications	16
3.1.4	Multiple Errors in Master/Slave Applications	20
3.2	Detailed Description of the Commands	21
3.2.1	General Command Structure	22
3.2.2	Definitions for all Functions	23
3.2.3	Error Codes	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 Contact Addresses	39
Index	41

1 Introduction

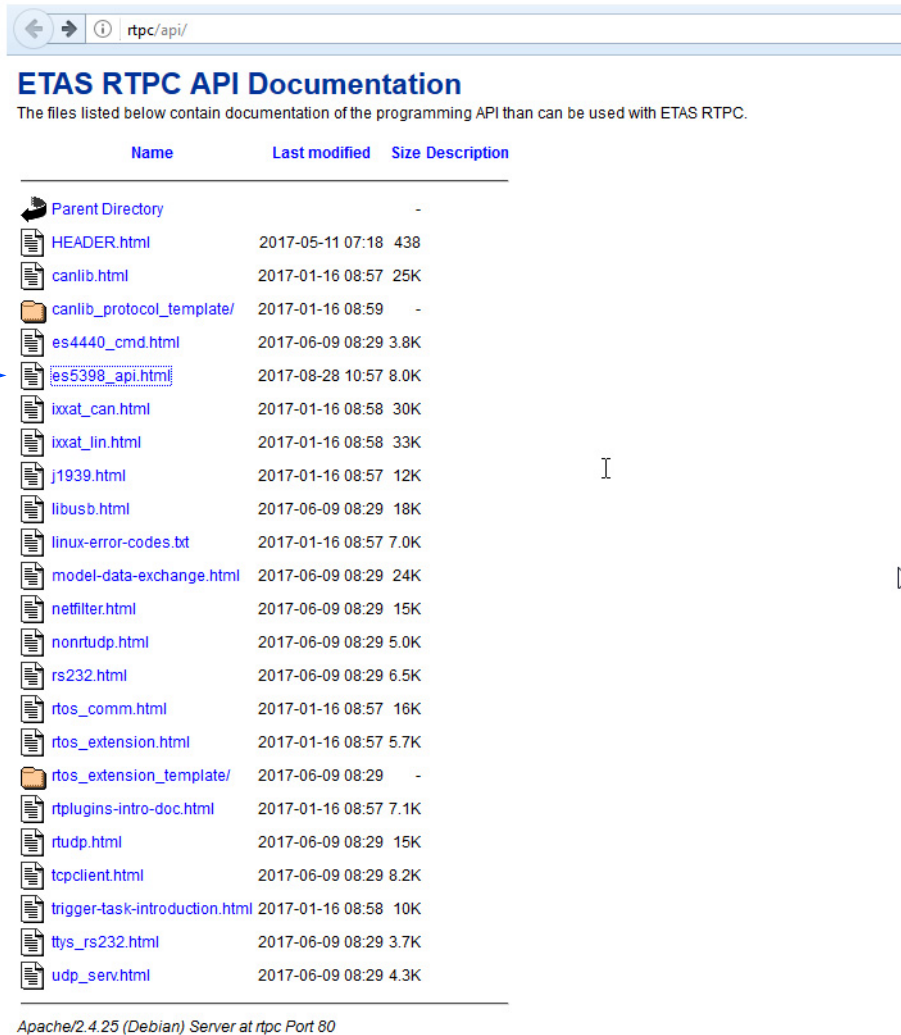
The ES5398.1 is a plug-in board for the ES5300 system housing and is used to reproduce electrical error cases. This simulation of error cases is used to test the diagnostics functions of an ECU.

A C-Code Module for the ES5398.1 is available via LABCAR-OPERATOR. The API of the C-Code Module for the ES5398.1 is described in this manual.

The C-Code Module can be used both for the ES5398.1 and for the ES4440.1/2. The ES5398.1 Error Insertion Board should be connected to the RTPC via an RTOS driver.

Note

You can find the current ES5398 RTOS driver API documentation directly on the RTPC under *rtpc/api* (see screenshot).



1.1 About this Manual

This manual consists of the following chapters:

- "Introduction" on page 5
This chapter
- "Integration of the C-Code Module for ES5398.1 into a LABCAR-OPERATOR Project" on page 9
This chapter explains the integration of a C-Code Module for ES5398.1 into a LABCAR-OPERATOR project. The C-Code Module controls the ES5398.1.
- "API Description" on page 14
This chapter presents possible program sequences for the ES5398.1.

1.1.1 User Profile

This manual is aimed at specialist personnel in the fields of motor vehicle ECU development and testing. Specialist knowledge in the field of measuring and control unit technology is assumed.

1.1.2 Using this Manual

Presentation of Information

All activities to be performed by the user are presented in a so-called "use case" format. This means that the goal to be achieved is first briefly defined in the title line and the respective necessary steps to achieve this goal are then presented in a list. The presentation looks as follows:

Goal definition

Any preliminary information...

1. Step 1
Any explanation of step 1...
2. Step 2
Any explanation of step 2...
3. Step 3
Any explanation of step 3...

Any concluding comments...

Specific example:

Creating a new file

Before creating a new file, no other file may be open.

1. Select **File** → **New**.
The "Create File" dialog box appears.

2. Enter the name of the new file in the "File name" field.

The file name may must not be longer than eight characters.

3. Click on **OK**.

The new file is being created and saved under the name you specified. You can now work with the file.

Typographical conventions

The following typographical conventions are used:

Select File → Open .	Menu commands are displayed in bold/blue.
Click on OK .	Buttons are displayed in bold/blue.
Press <ENTER>.	Keyboard commands are displayed in angle brackets in small capitals.
The "Open file" dialog window appears.	Names of program windows, dialog windows, fields and similar are given in quotation marks.
Select the <code>setup.exe</code> file.	Text in selection lists, program code, as well as path and file names are displayed using the <code>Courier</code> font.
A conversion between logical and arithmetic data types is <i>not</i> possible.	Emphasized text portions and newly introduced terms are printed in <i>italic</i> font face

Important notes for the user are presented as follows:

Note

Important note for the user

2 Integration of the C-Code Module for ES5398.1 into a LABCAR-OPERATOR Project

This chapter explains the integration of a C-Code Module for ES5398.1 into a LABCAR-OPERATOR project. The C-Code Module controls the ES5398.1.

All source files required for configuration and control of the ES5398.1 are made available on the ES5398.1 Software Integration DVD (F-00K-111-293).

Note

The ES5398.1 is supported by LABCAR-OPERATOR V5.4.4 and higher.

The C-Code Module for the ES5398.1 can also be used for the ES4440.1/2.

The necessary source files are in the subfolder `ES5398 Software Integration\ES4440Access`.

2.1 Building a New Project with ES5398.1

To integrate the C-Code Module in your LABCAR-OPERATOR project and configure it for ES5398.1 or ES4440.1/2, follow the steps below.

To integrate the C-Code Module for ES5398.1 / ES4440.1/2

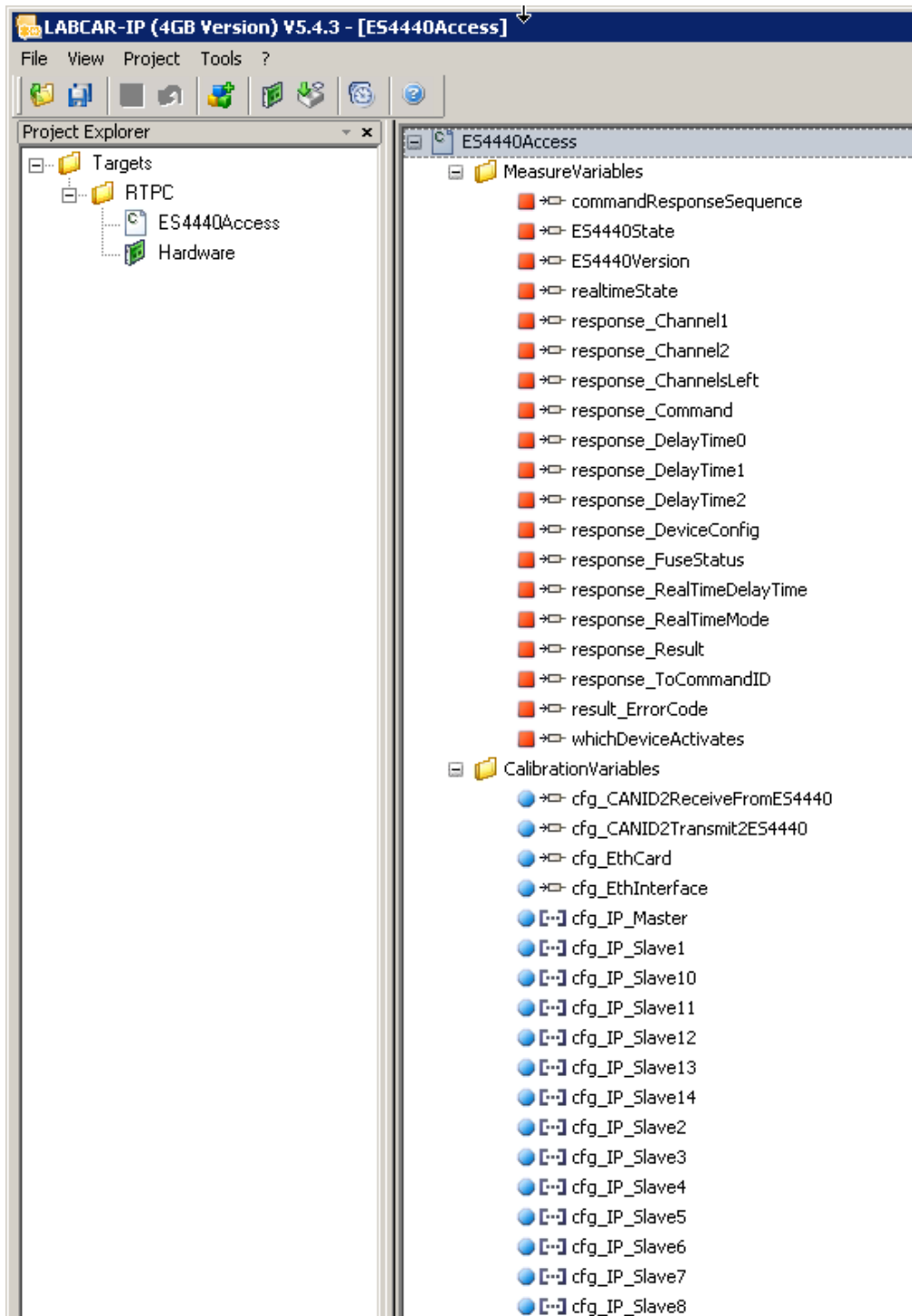
1. Open LABCAR-OPERATOR.
2. Select the **File** → **New Project** menu item.
3. Select a LABCAR-OPERATOR 5.x project with default target name "RTPC" and "No model".
4. Save the new project.
5. Integrate the C-Code Module into the project.
To do this, click the **Add Module** → **Add C-Code Module** → **Use existingModule (will be copied)** button.
6. Click on **Browse**.
7. Select the file `ES4440Access.lmd` from the subfolder `\ES5398 Software Integration\ES4440Access`.

Note

The entire module folder with all subdirectories is copied to the project folder.

To configure the C-Code Module for ES5398.1 / ES4440.1/.2

1. Double-click on the entry **ES4440Access** in the "Project Explorer" and then click on **OK**.
The MeasureVariables and CalibrationVariables of the C-Code Module appear in the "ES4440Access" window:



- Configure the following CalibrationVariables:

cfg_EthInterface
 cfg_IP_Master
 cgf_IP_SlaveX

Therefore, click on the CalibrationVariable, and then, in the "Model" window, click on **Value**. Adjust the value in the column behind according to the following description.

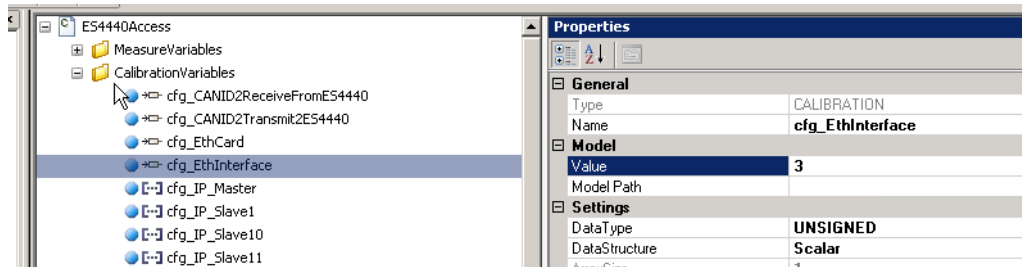
CalibrationVariables	Value	Comment
cfg_EthInterface	3	Activates driver for ES5398.1
	0	Activates driver for ES4440.1/2
cfg_IP_Master	Board ID	Global board ID: "Global Instance" on RTPC Web Interface, 4th value of the IP address
cgf_IP_SlaveX	Board ID	Global board ID: "Global Instance" on RTPC Web Interface, 4th value of the IP address

These variables remain constant while the experiment is being carried out.

- To activate the driver for the ES5398.1 or for the ES4440.1/2, for cfg_EthInterface, set the value to 3 or to 0, respectively.

Example for configuring cfg_EthInterface:

Value = 3 means configuration for ES5398.1, see table above.

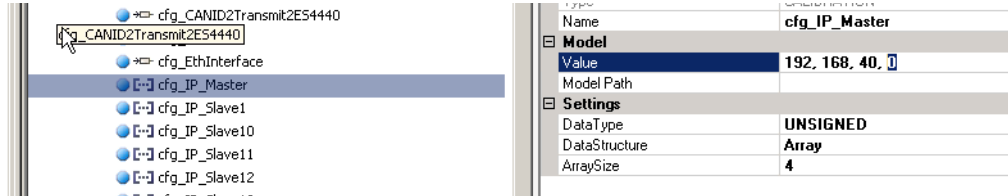


- To assign the global board ID ("Global Instance" on the RTPC Web Interface) to the master and slave devices, set the fourth value of the IP-address according to the following table.

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

Example for configuring cfg_IP_Master:

The fourth value of the IP address is key. "0" in this example corresponds to an ES5398.1 master board with global board ID "0".

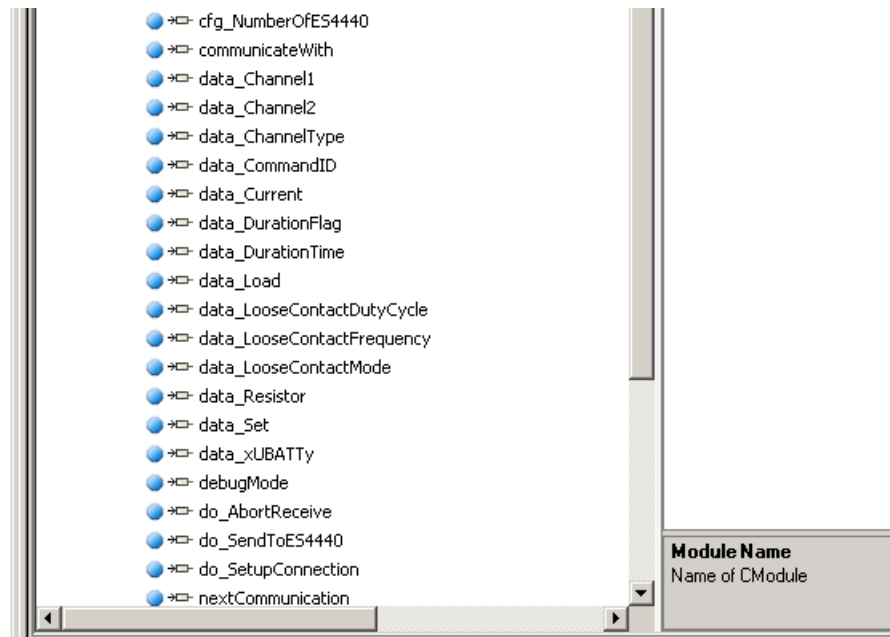


5. Click on **Project → OS Configuration**.
Make sure that the process "cmod_run_ES4440_ES4440Access" is included in the standard cycle.
6. Build the real-time code on the RTPC by clicking **Project → Build**.

Note
Clean intermediate files is recommended.

Integration of the C-Code Module into the LABCAR-OPERATOR project is now complete.

Note
The "ES4440Access" window contains further CalibrationVariables, which need to be set for the ES5398.1 or ES4440.1/2 when programming for automatic operation. How to set the values is explained in the chapter "Detailed Description of the Commands" on page 21.



3 API Description

This chapter outlines possible program sequences and provides a description of the commands for the API.

This chapter contains the following information:

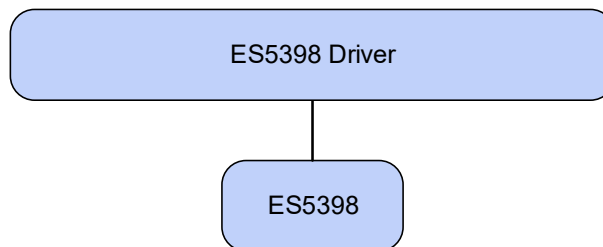
- "Program Sequences" on page 14
- "Detailed Description of the Commands" on page 21

3.1 Program Sequences

This chapter presents possible program sequences for the ES5398.1.

3.1.1 Single Errors

The program sequences described in this chapter apply to single errors, which are executed on a ES5398.1.



Program sequence for errors generated with a relay

1. Error configuration

High-current errors:

- Open_Load
- or
- ShortCut_xUBATTy_20A
- or
- Pin2PinFirstChWithoutLoad/Pin2PinSecondChWithoutLoad

2. Error activation

- Activate_relay

3. Reset the error

- Reset_all_errors

Program sequence for errors implemented with MOSFETs

1. Error configuration

- Open_Load_realtime
- or
- ShortCut_xUBATTy_20A_realtime
- or

- Pin2PinFirstChRelatimeWithLoad
- or
- Pin2PinSecondChRealtimeWithLoad
- or
- RInline_realtime
- or
- Pullup_Pulldown_xUBATTy_20A_realtime

2. Error activation

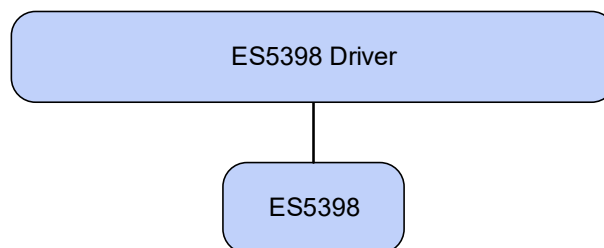
- Activate_realtime_switch

3. Reset the error

- Reset_all_errors

3.1.2 Multiple Errors in Standalone Applications

The program sequences described in this chapter apply to multiple errors, which are executed on a ES5398.1.



Program sequence for errors generated with a relay

1. Error configuration

(max. ten commands are sent to the ES5398.1 in sequence)

- Open_Load
- and/or
- ShortCut_xUBATTy_20A

2. Error activation

- Activate_relay

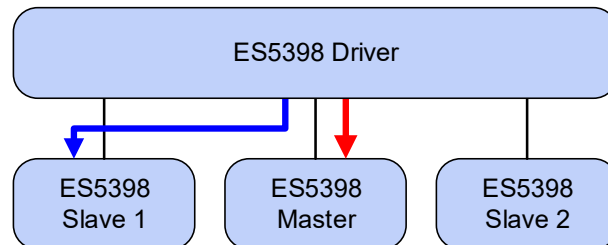
3. Reset the errors

- Reset_all_errors

3.1.3 Single Errors in Master/Slave Applications

The program sequences described in this chapter apply to single errors, which are executed on a ES5398.1 master/slave system.

Program sequence for errors generated with a relay



1. Error configuration (on slave 1)

Max. ten high-current errors:

- `Open_Load`
and/or
- `ShortCut_xUBATTy_20A`

2. Error activation (on master)

- `Activate_relay`

3. Reset the error (on slave 1) *

- `Reset_all_errors`

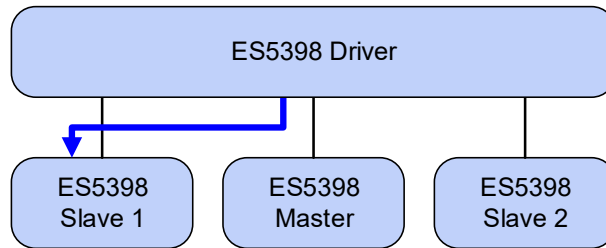
4. Reset the error (on master) **

- `Reset_all_errors`

* The "Reset the error" command on a slave is initially only saved.

** The "Reset the error" command on the master causes synchronous resetting of the error on the master and all slaves that previously saved this command.

Program sequence for errors generated with MOSFETs



1. Error configuration

- `Open_Load_realtime`
or
- `ShortCut_xUBATTy_20A_realtime`

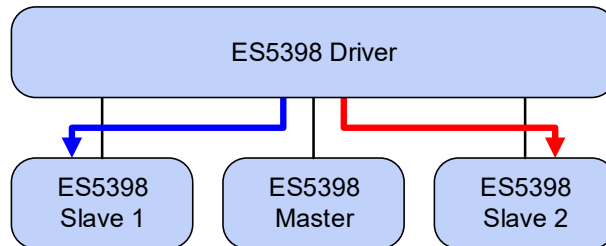
2. Error activation

- `Activate_realtime_switch`

3. Reset the error

- `Reset_all_errors`

Special case: Pin-to-pin error with load



1. Error configuration for first pin (on slave 1)

- `Pin2PinFirstChRelatimeWithLoad`

2. Error configuration for second pin (on slave 2)

- `Pin2PinSecondChRealtimeWithLoad`

3. Error activation (on slave 1)

- `Activate_realtime_switch`

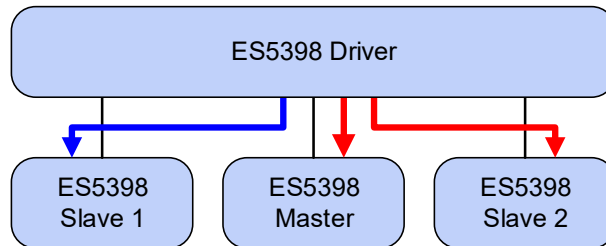
4. Reset the error (on slave 1)

- `Reset_all_errors`

5. Reset the error (on slave 2)

- `Reset_all_errors`

Special case: Pin-to-pin without load



- 1. Error configuration for first pin (on slave 1)**
 - `Pin2PinFirstChWithoutLoad`
- 2. Error configuration for second pin (on slave 2)**
 - `Pin2PinSecondChWithoutLoad`
- 3. Error activation (on master)**
 - `Activate_relay`
- 4. Reset the error (on slave 1)**
 - `Reset_all_errors`
- 5. Reset the error (on slave 2)**
 - `Reset_all_errors`
- 6. Reset the error (on master)**
 - `Reset_all_errors`

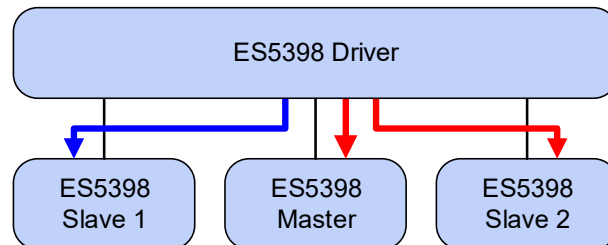
Note

This error is switched via relay – there is no fuse in the error path between pin 1 and pin 2.

3.1.4 Multiple Errors in Master/Slave Applications

The program sequences described in this chapter apply to multiple errors, which are executed on a master/slave system.

Program sequence for errors generated with a relay



1. Error configuration (on slave 1)

Max. ten high-current errors:

- Open_Load
and/or
- ShortCut_xUBATTy_20A

2. Error configuration (on slave 2)

Max. ten high-current errors:

- Open_Load
and/or
- ShortCut_xUBATTy_20A

3. Error activation (on master)

- Activate_relay

4. Reset the error (on slave 1)

- Reset_all_errors

5. Reset the error (on slave 2)

- Reset_all_errors

6. Reset the error (on master)

- Reset_all_errors

3.2 Detailed Description of the Commands

In this section, you can find the complete syntax description of all commands.

Specifically, these are:

- "Open_Load" on page 27
- ".Open_Load_realtime" on page 28
- "ShortCut_xUBATTy_20A" on page 29
- "ShortCut_xUBATTy_20A_realtime" on page 30
- "Pin2PinFirstChWithoutLoad" on page 31
- "Pin2PinSecondChannelWithoutLoad" on page 32
- "Pin2PinFirstChRealtimeWithLoad" on page 33
- "Pin2PinSecondChRealtimeWithLoad" on page 34
- "Reset_all_errors" on page 35
- "Activate_relay" on page 36
- "Activate_realtime_switch" on page 38

3.2.1 General Command Structure

In the following, you can find a description of the general structure of the send and receive messages between the C-Code Module and the ES5398.1 driver.

Send message

COM_command_name(arguments)		
1st byte	Command ID	<i>Value</i>
2nd byte	Parameter 0	<i>Value</i>
3rd byte	Parameter 1	<i>Value</i>
4th byte	Parameter 2	<i>Value</i>
5th byte	Parameter 3	<i>Value</i>
6th byte	Parameter 4	<i>Value</i>
7th byte	Parameter 5	<i>Value</i>
8th byte	Parameter 6	<i>Value</i>

Tab. 3-1 Structure of a send message

Receive message

Response		
1st byte	Command ID	<i>Value</i>
2nd byte	Parameter 0	<i>Value</i>
3rd byte	Parameter 1	<i>Value</i>
4th byte	Parameter 2	<i>Value</i>
5th byte	Parameter 3	<i>Value</i>
6th byte	Parameter 4	<i>Value</i>
7th byte	Parameter 5	<i>Value</i>
8th byte	Parameter 6	<i>Value</i>

Tab. 3-2 Structure of the receive message

Note

The ES5398.1 uses a send/receive protocol. Send messages are confirmed by the system with an accompanying receive message. No new send messages may be sent until this receive message has been received.

3.2.2 Definitions for all Functions

The information in this section applies to all commands.

Note

For the CalibrationVariables (see "ES4440Access" window on page 12), the values to be set are specified at the corresponding points.

Channel numbers

The channels are numbered from 0 to 39.

In the "ES4440Access" window of the C-Code Module, this corresponds to the CalibrationVariables data_Channel11 or data_Channel12.

Definition for parameter 1 of the send message:

Parameter 1 (3rd byte) has the same structure for all functions:

		CalibrationVariable
Bit 0	Load Load = 1: Error simulation with load Load = 0: Error simulation without load	data_Load: Set to 0 or 1
Bit 1	xUBatty xUBatty = 0: +UBatt_A	data_xUBATTy: Set to 0, 1, or 2
Bit 2	xUBatty = 1: -UBatt	
Bit 3	xUBatty = 2: +UBatt_B	
Bit 4	Not Used	
Bit 5	Set Set = 1: Error set Set = 0: Error reset	data_Set: Set to 0 or 1
Bit 6	dura- tion_- flag duration_flag = 0: Error present contin- uously (i.e. until reset). duration_flag = 1: Error duration is specified by "duration_time".	data_Dura- tionFlag: Set to 0 or 1
Bit 7	Not used	

3.2.3 Error Codes

Error codes are transferred in the receive message and appear in the Measure-Variable `response_Command`.

Error codes, previously applicable to ES4440.1, partially applicable to ES5398.1

The error codes in the following table apply to ES4440.1/.2. In some cases, they also apply to ES5398.1.

Error codes that are new and apply exclusively to the ES5398.1 are listed in the subsequent table (see "New error codes for ES5398.1" on page 25).

- White background: For ES5398.1 and ES4440.1/.2
- Gray background: Only for ES4440.1/.2

Result	Meaning
0x0	Command OK
0x21	Incorrect parameter for slave address (> 16)
0x22	Unknown command
0x23	Incorrect data type when writing to flash memory
0x24	Incorrect parameter during LED test
0x25	Number in IP address too large (must be < 256)
0x26	Incorrect parameter for CAN baud rate
0x27	Incorrect parameter for CAN scheduling
0x28	Incorrect parameter for CAN identifier type
0x29	Parameter for cascade channel too high (must be < 15)
0x2a	Incorrect parameter for resistor cascade
0x2b	Not used
0x2c	Incorrect address during flash memory read access (valid values < 513)
0x2d	Incorrect data length during flash memory read access (valid values < 17)
0x2e	Incorrect address during flash memory write access (valid values < 513)
0x2f	Incorrect data length during flash memory write access (valid values < 17)
0x30	PLD error
0x31	EEPROM checksum error
0x32	CAN controller does not respond
0x41	Simulation command returns plausibility error
0x42	Reference relay was not detected
0x43	"duration_time" does not have the value "0xffff", despite the fact that an error is to be present indefinitely.
0x44	Simulation command was not detected

Result	Meaning
0x45	PLD internal hardware error: Command could not be correctly switched
0x46	Value of "duration time" outside the valid range (1 to 6000 or 0xFFFF)
0x47	Last error simulation is still active, delete with <code>Reset_all_errors</code>
0x48	Max. number of relays reached
0x49	Error with MultiErrorFlag
0x4a	Specified channel number outside the valid range
0x4b	Frequency or duty cycle outside the valid range
0x4c	System temperature > 60 °C
0x4d	Resistor cascade temperature > 60 °C
0x4e	MOSFET temperature > 60 °C
0x4f	System temperature sensor defective
0x50	Resistor cascade temperature sensor defective
0x51	MOSFET temperature sensor defective
0x52	Rail voltage not correct (possible short-circuit)
0x53	Invalid resistor value

New error codes for ES5398.1

Several error codes are new and apply exclusively to the ES5398.1. They are listed in the following table:

Result	Meaning	Detail
0x60	ADC overvoltage	" +VBATA exceeds the limit for relay errors "
0x61	ADC overvoltage	" +VBATA exceeds the limit for relay errors "
0x62	ADC overvoltage	" RAIL1 voltage exceeds the limit for relay errors "
0x63	ADC overvoltage	" RAIL2 voltage exceeds the limit for relay errors "
0x64	ADC overvoltage	" Ext. resistor voltage exceeds the limit for relay errors "
0x65	ADC overcurrent	" Current in failure path exceeds limit "
0x66	MOSFET overtemperature	" Realtime switch exceeds limit for temperature "
0x70	"TER error in SPI to load-rail relays"	

Result	Meaning	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

Interrupts a line between ECU and load. This error is switched via relay – up to ten errors can be switched at the same time. The value of "channels left" in the command response specifies how many channels are still available for further errors.

Send message

Open_Load(channel_no, duration_flag, set)			CalibrationVariable
1st byte	Command ID	0x1	data_CommandID
2nd byte	Parameter 0	Channel number	data_Channel11
3rd byte	Parameter 1	Set, duration_flag (see page 23)	data_Set, data_DurationFlag (for values see page 23)
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Not used	

Receive message

Response			MeasureVariable
1st byte	Command ID	0x1	
2nd byte	Parameter 0	Channel number	
3rd byte	Parameter 1	channels left	
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Command result	response_Command

The parameter 6 command result corresponds to the MeasureVariable **responseCommand** in the "ES4440Access" window (see page 10). This also applies in the following.

3.2.5 .Open_Load_realtime

Interrupts a line between ECU and load. This error is switched with MOSFETs and can only be implemented as a single error.

Send message

Open_Load_realtime (channel_no, duration_flag)			CalibrationVariable
1st byte	Command ID	0x2	
2nd byte	Parameter 0	ES5398.1 channel number	
3rd byte	Parameter 1	duration_flag	data_DurationFlag (for values see page 23)
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Not used	

Receive message

Response			MeasureVariable
1st byte	Command ID	0x2	
2nd byte	Parameter 0	ES5398.1 channel number	
3rd byte	Parameter 1	Not used	
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Command result	response_Command

3.2.6 ShortCut_xUBATTy_20A

Generates a line short-circuit to a battery voltage in a high-current channel. This error is switched via relay and can be implemented multiple times.

Send message

ShortCut_xUBATTy_20A (channel_no, load, xUBATTy, duration_flag, set)			CalibrationVariable
1st byte	Command ID	0x3	
2nd byte	Parameter 0	ES5398.1 channel number	
3rd byte	Parameter 1	Load, xUBatty, set, duration_flag)	data_Load, data_xUBATTy, data_Set, data_DurationFlag (for values see page 23)
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Not used	

Receive message

Response			MeasureVariable
1st byte	Command ID	0x3	
2nd byte	Parameter 0	ES5398.1 channel number	
3rd byte	Parameter 1	channels left	
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Command result	responseCommand

3.2.7 ShortCut_xUBATTy_20A_realtime

Generates a line short-circuit to a battery voltage in a high-current channel. This error is switched with MOSFETs and can therefore only be implemented as a single error.

Send message

ShortCut_xUBATTy_20A_realtime (channel_no, load, xUBATTy, duration_flag)			CalibrationValue
1st byte	Command ID	0x4	
2nd byte	Parameter 0	Channel number	
3rd byte	Parameter 1	Load, xUBatty, duration_flag	data_Load, data_xUBATTy, data_DurationFlag (for values see page 23)
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Not used	

Receive message

Response			MeasureVariable
1st byte	Command ID	0x4	
2nd byte	Parameter 0	Channel number	
3rd byte	Parameter 1	Not used	
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Command result	responseCommand

3.2.8 Pin2PinFirstChWithoutLoad

Defines the first line for a short-circuit between two lines – the second line is defined with the "Pin2PinSecondChannelWithoutLoad" command (see page 32).

Note

This error is switched via relay and implemented both without load and without resistor. There is no fuse between pin 1 and pin 2.

Send message

Pin2PinFirstChWithoutLoad (channel_no1, duration_flag)			CalibrationVariable
1st byte	Command ID	0x5	
2nd byte	Parameter 0	ES5398.1 channel number 1	data_Channel1
3rd byte	Parameter 1	duration_flag	data_DurationFlag (for values see page 23)
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Not used	

Receive message

Response			MeasureVariable
1st byte	Command ID	0x5	
2nd byte	Parameter 0	ES5398.1 channel number 1	
3rd byte	Parameter 1	Not used	
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Command result	responseCommand

3.2.9 Pin2PinSecondChannelWithoutLoad

Defines the second line for a short-circuit between two lines.

Note

This error is switched via relay and implemented both without load and without resistor. There is no fuse between pin 1 and pin 2.

Send message

Pin2PinSecondChannelWithoutLoad (channel_no1, duration_flag)			CalibrationVariable
1st byte	Command ID	0x6	
2nd byte	Parameter 0	Channel number 2	data_Channel2
3rd byte	Parameter 1	duration_flag	data_Duration - Flag (for values see page 23)
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Not used	

Receive message

Response			MeasureVariable
1st byte	Command ID	0x6	
2nd byte	Parameter 0	Channel number 2	
3rd byte	Parameter 1	Not used	
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Command result	responseCommand

3.2.10 Pin2PinFirstChRealtimeWithLoad

Defines the first line for a short-circuit between two lines. This error enables the simulation with load and finite resistor between the two lines.

This error is switched with MOSFETs.

Send message

Pin2PinFirstChRealtimeWithLoad (channel_no1, duration_flag)			CalibrationVariable
1st byte	Command ID	0x7	
2nd byte	Parameter 0	Channel number 1	data_Channel1
3rd byte	Parameter 1	Current, duration_flag (see page 23)	data_DurationFlag (for values see page 23)
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Byte 0 resistor	data_Resistor (Use value !=0)
6th byte	Parameter 4	Byte 1 resistor	data_Resistor (Use value !=0)
7th byte	Parameter 5	Byte 2 resistor	data_Resistor (Use value !=0)
8th byte	Parameter 6	Byte 3 resistor	data_Resistor (Use value !=0)

Receive message

Response			MeasureVariable
1st byte	Command ID	0x7	
2nd byte	Parameter 0	channel number 1	
3rd byte	Parameter 1	Not used	
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Command result	responseCommand

3.2.11 Pin2PinSecondChRealtimeWithLoad

Defines the second line for a short-circuit between two lines. This error enables the simulation with load and finite resistor between the two lines.

This error is switched with MOSFETs.

Send message

Pin2PinSecondChRealtimeWithLoad (channel_no2, duration_flag)			CalibrationVariable
1st byte	Command ID	0x8	
2nd byte	Parameter 0	Channel number 2	data_Channel12
3rd byte	Parameter 1	duration_flag	data_Duration_ - Flag (for values see page 23)
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Not used	

Receive message

Response			MeasureVariable
1st byte	Command ID	0x8	
2nd byte	Parameter 0	Channel number 2	
3rd byte	Parameter 1	Not used	
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Not used	

3.2.12 Reset_all_errors

This command is used to reset all errors on a ES5398.1.

Send message

Reset_all_errors ()			CalibrationVariable
1st byte	Command ID	0x10	
2nd byte	Parameter 0	Not used	
3rd byte	Parameter 1	Not used	
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Not used	

Receive message

Response			MeasureVariable
1st byte	Command ID	0x10	
2nd byte	Parameter 0	Not used	
3rd byte	Parameter 1	Not used	
4th byte	Parameter 2	Not used	
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Command result	responseCommand

3.2.13 Activate_relay

This command is used to close the relay for a certain time.

If "duration_flag" (bit 6 in the 3rd byte) is set in the previous error command (= 1), the "duration_time" can selected between 1 ms and 5 s. If "duration_flag" = 0, then "duration_time" = -1 or = 65535 (0xFFFF) must be selected.

Note

When setting multiple errors, the "duration_flag" parameters of all errors must have the same value.

The switching times measured on the reference relay are transferred in the command response.

Send message

Activate_relay (duration_time)			CalibrationVariable
1st byte	Command ID	0x12	
2nd byte	Parameter 0	Not used	
3rd byte	Parameter 1	duration_time in ms (byte 0)	data_Duration-Time (for values see below)
4th byte	Parameter 2	duration_time in ms (byte 1)	data_Duration-Time (for values see below)
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Not used	

Values for **data_DurationTime**:

- If data_DurationFlag was 0, set the value to -1.
- Otherwise, enter the time in ms, as described in the following:
- The parameter "duration_time" can be specified in steps of 20 ms – the lowest possible value is 20 ms, the highest possible value is 5000 ms.
- Looping errors are generated with the value 0xFFFF.

The "channel_type" parameter can assume the following values:

- channel_type = 0: Current channel
- channel_type = 1: Voltage channel

*Receive message***Response**

1st byte	Command ID	0x12
2nd byte	Parameter 0	delay_time0 NO contact* 20 A closes (in 100 µs steps)
3rd byte	Parameter 1	delay_time1 NO contact* 20 A closes (in 100 µs steps)
4th byte	Parameter 2	delay_time0 NC contact** 20 A opens (in 100 µs steps)
5th byte	Parameter 3	delay_time1 NC contact** 20 A opens (in 100 µs steps)
6th byte	Parameter 4	delay_time0 NC contact** 400 V closes (in 100 µs steps)
7th byte	Parameter 5	NC contact** 400 V closes (in 100 µs steps)
8th byte	Parameter 6	Command result (responseCommand)

* NO = normally open contact

** NC= normally closed contact

3.2.14 Activate_realtime_switch

This command is used to close an error switched with MOSFETs for a certain time.

If "duration_flag" (bit 6 in the 3rd byte) is set in the previous error command (= 1), the "duration_time" can selected between 1 ms and 5 s. If "duration_flag" = 0, then "duration_time" = -1 or = 65535 (0xFFFF) must be selected.

Send message

Activate_realtime_switch (mode, duration_time, dutycycle, frequency)			CalibrationVariable
1st byte	Command ID	0x13	
2nd byte	Parameter 0	Mode	
3rd byte	Parameter 1	duration_time (byte 0)	data_Duration-Time
4th byte	Parameter 2	duration_time (byte 1)	data_Duration-Time
5th byte	Parameter 3	Not used	
6th byte	Parameter 4	Not used	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Not used	

The "duration_time" parameter can be specified in steps of 1 ms – the lowest possible value is 1 ms, the highest possible value is 5000 ms. Looping errors are generated with the value 0xFFFF.

The "mode" parameter can assume the following values:

- Mode = 0: Static error, whose duration is defined by "duration_time".

Receive message

Response			MeasureVariable
1st byte	Command ID	0x13	
2nd byte	Parameter 0	Mode	
3rd byte	Parameter 1	duration_time (byte 0)	
4th byte	Parameter 2	duration_time (byte 1)	
5th byte	Parameter 3	duration_time (byte 2)	
6th byte	Parameter 4	duration_time (byte 3)	
7th byte	Parameter 5	Not used	
8th byte	Parameter 6	Command result	responseCommand

4 **ETAS Contact Addresses**

ETAS HQ

ETAS GmbH

Borsigstraße 24
70469 Stuttgart, Germany
Germany

Phone: +49 (0) 711 3423-0
Fax: +49 (0) 711 3423-2106
WWW: www.etas.com

ETAS Subsidiaries and Technical Support

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

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

Index

A

Activate_realtime_switch 38
Activate_relay 36

C

CAN command
 ShortCut_xUBATTy_400V_Ex 34
 Test fuses 38
Channels left 27
CurrentMeasurement 38

E

ETAS contact addresses 39

G

General command structure 22

I

Introduction 5

O

Open_Load 27
Open_Load_realtime 28
Operation
 Conventions 7
 Use case 6

P

Pin2PinFirstChRealtimeWithLoad 33
Pin2PinFirstChWithoutLoad 31

Pin2PinSecondChannelWithoutLoad
 32

Pin2PinSecondChRealtimeWithLoad
 34

R

Reset_all_errors 35
RInline_realtime 34

S

ShortCut_xUBATTy_20A 29
ShortCut_xUBATTy_20A_realtime 30

U

User profile 6

