



Elektrobit

# EB tresos<sup>®</sup> E2E Profile 1 documentation

product release 8.8.3



Elektrobit Automotive GmbH  
Am Wolfsmantel 46  
91058 Erlangen, Germany  
Phone: +49 9131 7701 0  
Fax: +49 9131 7701 6333  
Email: [info.automotive@elektrobit.com](mailto:info.automotive@elektrobit.com)

## Technical support

<https://www.elektrobit.com/support>

## Legal disclaimer

Confidential information.

ALL RIGHTS RESERVED. No part of this publication may be copied in any form, by photocopy, microfilm, retrieval system, or by any other means now known or hereafter invented without the prior written permission of Elektrobit Automotive GmbH.

All brand names, trademarks, and registered trademarks are property of their rightful owners and are used only for description.

Copyright 2021, Elektrobit Automotive GmbH.

# Table of Contents

1. Overview of EB tresos E2E Profile 1 documentation .....	7
2. E2EP01 release notes .....	8
2.1. Overview .....	8
2.2. Scope of the release .....	8
2.2.1. Configuration tool .....	8
2.2.2. AUTOSAR modules .....	8
2.2.3. EB (Elektrobit) modules .....	8
2.2.4. MCAL modules and EB tresos AutoCore OS .....	9
2.3. Module release notes .....	9
2.3.1. E2E module release notes .....	9
2.3.1.1. Change log .....	9
2.3.1.2. New features .....	13
2.3.1.3. EB-specific enhancements .....	13
2.3.1.4. Deviations .....	14
2.3.1.5. Limitations .....	14
2.3.1.6. Open-source software .....	15
2.3.2. E2EP01 module release notes .....	15
2.3.2.1. Change log .....	15
2.3.2.2. New features .....	19
2.3.2.3. EB-specific enhancements .....	20
2.3.2.4. Deviations .....	21
2.3.2.5. Limitations .....	23
2.3.2.6. Open-source software .....	23
2.3.3. SCrc module release notes .....	23
2.3.3.1. Change log .....	23
2.3.3.2. New features .....	26
2.3.3.3. EB-specific enhancements .....	27
2.3.3.4. Deviations .....	27
2.3.3.5. Limitations .....	27
2.3.3.6. Open-source software .....	27
3. E2EP01 user's guide .....	28
3.1. Overview .....	28
3.2. Background information .....	28
3.2.1. Functional overview .....	28
3.2.1.1. Safety mechanisms .....	28
3.2.1.2. Failure modes and required safety mechanisms .....	30
3.3. Configuring E2EP01 .....	31
3.4. E2EP01 integration notes .....	32
4. E2EP01 module references .....	33

4.1. Overview .....	33
4.1.1. Notation in EB module references .....	33
4.1.1.1. Default value of configuration parameters .....	33
4.1.1.2. Range information of configuration parameters .....	33
4.2. E2E .....	34
4.2.1. Configuration parameters .....	34
4.2.1.1. CommonPublishedInformation .....	34
4.2.1.2. PublishedInformation .....	37
4.2.2. Application programming interface (API) .....	37
4.2.2.1. Type definitions .....	38
4.2.2.1.1. E2E_PCheckStatusType .....	38
4.2.2.2. Macro constants .....	38
4.2.2.2.1. E2E_AR_MAJOR_VERSION .....	38
4.2.2.2.2. E2E_AR_MINOR_VERSION .....	38
4.2.2.2.3. E2E_AR_PATCH_VERSION .....	38
4.2.2.2.4. E2E_AR_RELEASE_MAJOR_VERSION .....	38
4.2.2.2.5. E2E_AR_RELEASE_MINOR_VERSION .....	38
4.2.2.2.6. E2E_AR_RELEASE_REVISION_VERSION .....	39
4.2.2.2.7. E2E_E_INPUTERR_NULL .....	39
4.2.2.2.8. E2E_E_INPUTERR_WRONG .....	39
4.2.2.2.9. E2E_E_INTERR .....	39
4.2.2.2.10. E2E_E_INVALID .....	39
4.2.2.2.11. E2E_E_OK .....	39
4.2.2.2.12. E2E_E_WRONGSTATE .....	40
4.2.2.2.13. E2E_MODULE_ID .....	40
4.2.2.2.14. E2E_P_ERROR .....	40
4.2.2.2.15. E2E_P_NONEWDATA .....	40
4.2.2.2.16. E2E_P_NOTAVAILABLE .....	40
4.2.2.2.17. E2E_P_OK .....	40
4.2.2.2.18. E2E_P_REPEATED .....	41
4.2.2.2.19. E2E_P_WRONGSEQUENCE .....	41
4.2.2.2.20. E2E_SW_MAJOR_VERSION .....	41
4.2.2.2.21. E2E_SW_MINOR_VERSION .....	41
4.2.2.2.22. E2E_SW_PATCH_VERSION .....	41
4.2.2.2.23. E2E_VENDOR_ID .....	41
4.2.2.3. Functions .....	42
4.2.2.3.1. E2E_GetVersionInfo .....	42
4.2.3. Integration notes .....	42
4.2.3.1. Exclusive areas .....	42
4.2.3.2. Production errors .....	42
4.2.3.3. Memory mapping .....	42
4.2.3.4. Integration requirements .....	43

4.3. E2EP01 .....	43
4.3.1. Configuration parameters .....	43
4.3.1.1. CommonPublishedInformation .....	43
4.3.1.2. PublishedInformation .....	47
4.3.2. Application programming interface (API) .....	47
4.3.2.1. Type definitions .....	47
4.3.2.1.1. E2E_P01CheckStateType .....	47
4.3.2.1.2. E2E_P01CheckStatusType .....	47
4.3.2.1.3. E2E_P01ConfigType .....	48
4.3.2.1.4. E2E_P01DataIDMode .....	49
4.3.2.1.5. E2E_P01ProtectStateType .....	49
4.3.2.1.6. E2E_P01ReceiverStateType .....	49
4.3.2.1.7. E2E_P01ReceiverStatusType .....	50
4.3.2.1.8. E2E_P01SenderStateType .....	51
4.3.2.2. Macro constants .....	51
4.3.2.2.1. E2EP01_AR_MAJOR_VERSION .....	51
4.3.2.2.2. E2EP01_AR_MINOR_VERSION .....	51
4.3.2.2.3. E2EP01_AR_PATCH_VERSION .....	52
4.3.2.2.4. E2EP01_AR_RELEASE_MAJOR_VERSION .....	52
4.3.2.2.5. E2EP01_AR_RELEASE_MINOR_VERSION .....	52
4.3.2.2.6. E2EP01_AR_RELEASE_REVISION_VERSION .....	52
4.3.2.2.7. E2EP01_SW_MAJOR_VERSION .....	52
4.3.2.2.8. E2EP01_SW_MINOR_VERSION .....	52
4.3.2.2.9. E2EP01_SW_PATCH_VERSION .....	52
4.3.2.2.10. E2EP01_VENDOR_ID .....	53
4.3.2.3. Functions .....	53
4.3.2.3.1. E2E_P01Check .....	53
4.3.2.3.2. E2E_P01CheckInit .....	53
4.3.2.3.3. E2E_P01MapStatusToSM .....	54
4.3.2.3.4. E2E_P01Protect .....	55
4.3.2.3.5. E2E_P01ProtectInit .....	56
4.3.3. Integration notes .....	56
4.3.3.1. Exclusive areas .....	56
4.3.3.2. Production errors .....	56
4.3.3.3. Memory mapping .....	57
4.3.3.4. Integration requirements .....	57
4.4. SCrc .....	57
4.4.1. Configuration parameters .....	57
4.4.1.1. CommonPublishedInformation .....	58
4.4.1.2. PublishedInformation .....	61
4.4.1.3. SCrcPublishedInformation .....	61
4.4.2. Application programming interface (API) .....	61

4.4.2.1. Macro constants .....	62
4.4.2.1.1. SCRC_FUNCENABLED_SCRC_CRC16 .....	62
4.4.2.1.2. SCRC_FUNCENABLED_SCRC_CRC32P4 .....	62
4.4.2.1.3. SCRC_FUNCENABLED_SCRC_CRC8 .....	62
4.4.2.1.4. SCRC_FUNCENABLED_SCRC_CRC8H2F .....	62
4.4.2.1.5. SCRC_SW_MAJOR_VERSION .....	62
4.4.2.1.6. SCRC_SW_MINOR_VERSION .....	63
4.4.2.1.7. SCRC_SW_PATCH_VERSION .....	63
4.4.2.1.8. SCRC_VENDOR_ID .....	63
4.4.2.2. Functions .....	63
4.4.2.2.1. SCrc_CalculateCRC16 .....	63
4.4.2.2.2. SCrc_CalculateCRC32P4 .....	64
4.4.2.2.3. SCrc_CalculateCRC8 .....	64
4.4.2.2.4. SCrc_CalculateCRC8H2F .....	65
4.4.3. Integration notes .....	66
4.4.3.1. Exclusive areas .....	66
4.4.3.2. Production errors .....	66
4.4.3.3. Memory mapping .....	66
4.4.3.4. Integration requirements .....	66
5. Bibliography .....	67



# 1. Overview of EB tresos E2E Profile 1 documentation

Welcome to the EB tresos E2E Profile 1 (E2EP01) product documentation.

This document provides:

- ▶ [Chapter 2, “E2EP01 release notes”](#): release notes for the E2EP01 modules
- ▶ [Chapter 3, “E2EP01 user's guide”](#): containing background information and instructions
- ▶ [Chapter 4, “E2EP01 module references”](#): information about configuration parameters and the application programming interface

## 2. E2EP01 release notes

### 2.1. Overview

This chapter provides the E2EP01 product specific release notes. General release notes that are applicable to all products are provided in the EB tresos AutoCore Generic documentation. Refer to the general release notes in addition to the product release notes documented here.

### 2.2. Scope of the release

#### 2.2.1. Configuration tool

Your release of EB tresos AutoCore is compatible with the release of the EB tresos Studio configuration tool:

- ▶ EB tresos Studio: 28.1.0 b210701-0227

#### 2.2.2. AUTOSAR modules

The following table lists the AUTOSAR modules that are part of this E2EP01 release.

Module name	AUTOSAR version and revision	SWS version and revision	Module version	Supplier
<a href="#">E2E</a>	4.2.1 []	4.2.1 [0000]	2.0.18	Elektrobit Automotive GmbH
<a href="#">E2EP01</a>	4.2.1 []	4.2.1 [0000]	2.2.17	Elektrobit Automotive GmbH

Table 2.1. Hardware-Independent Modules specified by the AUTOSAR standard

#### 2.2.3. EB (Elektrobit) modules

The following table lists all modules which are part of this release but are not specified by the AUTOSAR standard. These modules include tooling developed by EB or they may hold files shared by all other modules.



Module name	Module version	Supplier
<a href="#">SCrc</a>	2.0.11	Elektrobit Automotive GmbH

Table 2.2. Modules not specified by the AUTOSAR standard

## 2.2.4. MCAL modules and EB tresos AutoCore OS

For information about MCAL modules and OS, refer to the respective documentation, which is available as PDF at `$TRESOS_BASE/doc/3.0_EB_tresos_AutoCore_OS` and `$TRESOS_BASE/doc/5.0_MCAL_modules`<sup>1</sup>. It is also available in the online help in EB tresos Studio. Browse to the folders `EB tresos AutoCore OS` and `MCAL modules`.

## 2.3. Module release notes

### 2.3.1. E2E module release notes

- ▶ AUTOSAR R4.2 Rev 1
- ▶ AUTOSAR SWS document version: 4.2.1
- ▶ Module version: 2.0.18.B439717
- ▶ Supplier: Elektrobit Automotive GmbH

#### 2.3.1.1. Change log

This chapter lists the changes between different versions.

##### Module version 2.0.18

2021-03-05

- ▶ Updated preprocessor include guards to be PC-lint compatible

##### Module version 2.0.17

2020-10-23

---

<sup>1</sup>`$TRESOS_BASE` is the location at which you installed EB tresos Studio.

- ▶ Internal module improvement. This module version update does not affect module functionality
- ▶ Implemented independency of E2E\_Mem.h from Base module

#### **Module version 2.0.16**

2020-06-19

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.0.15**

2020-02-21

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.0.14**

2020-01-24

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.0.13**

2019-10-11

- ▶ Improved implementation of signal unpack macros
- ▶ Improved range check implementation of signal pack macros
- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.0.12**

2019-07-05

- ▶ ASCE2E-766 Fixed known issue: Wrong safety related ComXf serialization of 64bit ISignalGroup members
- ▶ ASCE2E-771, ASCCOMXF-492 Fixed known issue: Invalid safety-related ComXf support for XfrmBuffer-LengthType configured to UINT32 (Note: requires also ComXf module update)

#### **Module version 2.0.11**

2019-06-14

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.0.10**

2019-02-15

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.0.9**

2018-10-26

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.0.8**

2018-05-29

- ▶ ASCE2E-626 Fixed known issue: Undefined behavior for safety related ComXf serialization

#### **Module version 2.0.7**

2018-03-16

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.0.6**

2018-02-16

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.0.5**

2017-10-27

- ▶ Implemented support for configurable type of BufferLength
- ▶ Implemented profile specific parts from E2E transformer in E2E profiles and library

#### **Module version 2.0.4**

2017-09-22

- ▶ Improved implementation of signal pack and unpack macros
- ▶ Implemented 64 bit signal pack and unpack macros
- ▶ Switch from MISRA-C:2004 to MISRA-C:2012

### **Module version 2.0.3**

2017-06-30

- ▶ Implemented signal pack and unpack macros

### **Module version 2.0.2**

2015-06-19

- ▶ Added additional return values required for E2E transformer concept
- ▶ Implemented deterministic start-up behavior for E2E State Machine according to Bugzilla #67553

### **Module version 2.0.1**

2013-02-15

- ▶ Provide a Basic Software Module Description that specifies the Memory Mappings

### **Module version 2.0.0**

2012-06-22

- ▶ Update to ASR 4.0 Rev3

### **Module version 1.0.4**

2011-09-16

- ▶ Internal module improvement. This module version update does not affect module functionality

### **Module version 1.0.3**

2011-05-20

- ▶ Update of version macros and Info
- ▶ Usage of unix line endings

### **Module version 1.0.2**

2010-12-23

- ▶ Each profile is implemented as an individual module

- ▶ The E2E library shall use the SCrc module instead of the AUTOSAR Crc module

#### **Module version 1.0.1**

2010-11-30

- ▶ Enumeration E2E\_P02ReceiverStateType must have defined values
- ▶ Removed dead code in E2E Library
- ▶ Input parameter is not correctly checked against Null pointer

#### **Module version 1.0.0**

2010-11-12

- ▶ Initial release

### **2.3.1.2. New features**

- ▶ No new features have been added since the last release.

### **2.3.1.3. EB-specific enhancements**

This chapter lists the enhancements provided by the module.

- ▶ Support for the usage with AUTOSAR E2E Transformer

Description:

In addition to AUTOSAR version R4.0 Rev 3, the following additional return values according to AUTOSAR release 4.2.1 are provided:

- ▶ E2E\_P\_OK
  - ▶ E2E\_P\_REPEATED
  - ▶ E2E\_P\_WRONGSEQUENCE
  - ▶ E2E\_P\_ERROR
- ▶ Deterministic initialization behavior for E2E State Machine

Description:

In addition to AUTOSAR release 4.2.1, Bugzilla RfC 67553 is incorporated. That is, a new generic profile status type E2E\_P\_NONEWDATA is introduced in case no new data was received in the actual receive cycle. See also [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=67553](http://www.autosar.org/bugzilla/show_bug.cgi?id=67553).

### 2.3.1.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

- ▶ [ASCE2E-10] No support for building customized E2E Profiles

Description:

The following functions are not supported:

- ▶ E2E\_CRC8\*
- ▶ E2E\_UpdateCounter

Rationale:

The generic E2E library is only used in combination with specific E2E profiles.

Requirements:

E2E0106, E2E0107, E2E0092, E2E0091, E2E0094, E2E0095, E2E0096, E2E0276, E2E0097, E2E0098, E2E0099

### 2.3.1.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- ▶ Packing of 64 bit signal types

Description:

In order to serialize 64-bit data types the following constraints apply to the configuration of a signal within a PDU:

- ▶ signal shall be byte aligned
- ▶ bit length shall be a multiple of 8
- ▶ the bitlength parameter value shall correlate with the value of the nbytes parameter

- ▶ Unpacking of 64 bit signal types

Description:

In order to deserialize 64-bit data types the following constraints apply to the configuration of a signal within a PDU:

- ▶ signal shall be byte aligned
- ▶ bitlength shall be a multiple of 8

- ▶ the bitlength parameter value shall correlate with the value of the nbytes parameter
- ▶ Range checks of 64 bit signal types

Description:

- ▶ Range checks for float signal types are not supported.
- ▶ Range checks for 64bit signal types are only supported on byte level.

### 2.3.1.6. Open-source software

E2E does not use open-source software.

## 2.3.2. E2EP01 module release notes

- ▶ AUTOSAR R4.2 Rev 1
- ▶ AUTOSAR SWS document version: 4.2.1
- ▶ Module version: 2.2.17.B439717
- ▶ Supplier: Elektrobit Automotive GmbH

### 2.3.2.1. Change log

This chapter lists the changes between different versions.

#### Module version 2.2.17

2021-04-09

- ▶ Implemented E2E Profile variant 1B (DataIDMode: E2E\_P01\_DATAID\_ALT)

#### Module version 2.2.16

2021-03-05

- ▶ Updated preprocessor include guards to be PC-lint compatible

#### Module version 2.2.15

2020-06-19

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.2.14**

2020-01-24

- ▶ Implemented extension of range in data validity check for ArrayLength up to 2048 bits

#### **Module version 2.2.13**

2019-10-11

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.2.12**

2019-06-14

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.2.11**

2018-12-21

- ▶ Implemented configurable initial value of the WaitForFirstData parameter

#### **Module version 2.2.10**

2018-10-26

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.2.9**

2018-03-02

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.2.8**

2017-10-27

- ▶ Implemented support for configurable type of BufferLength



- ▶ Implemented profile specific parts from E2E transformer in E2E profiles and library

#### **Module version 2.2.7**

2017-09-22

- ▶ Switch from MISRA-C:2004 to MISRA-C:2012

#### **Module version 2.2.6**

2017-08-25

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.2.5**

2015-11-06

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.2.4**

2015-06-19

- ▶ Implemented a deterministic start-up behavior for the E2E State Machine according to Bugzilla #67553

#### **Module version 2.2.3**

2014-04-07

- ▶ Internal module improvement. This module version update does not affect module functionality
- ▶ Removed the maximum data length check (depends on the application scenario)

#### **Module version 2.2.2**

2013-11-29

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.2.1**

2013-08-16

- ▶ Implemented support for Profile 01 variant C according to AUTOSAR 4.1 Rev 2 (RfC #59003)

#### **Module version 2.2.0**

2013-04-15

- ▶ Implemented support for Profile 01 variant C according to AUTOSAR 4.1 Rev 1

#### **Module version 2.1.3**

2013-02-15

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.1.2**

2012-12-12

- ▶ Implemented support for profile variant A and different CRC and counter offsets

#### **Module version 2.1.1**

2012-10-24

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.1.0**

2012-07-27

- ▶ Implemented a re-synchronization handling if the profile detects an unexpected behavior of the received counter according to AUTOSAR 3.2 Rev 2

#### **Module version 2.0.0**

2012-06-22

- ▶ Updated to AUTOSAR 4.0 Rev 3

#### **Module version 1.0.5**

2011-09-16

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 1.0.4**

2011-05-20

- ▶ Corrected a parameter check regarding `MaxDeltaCounterInit`
- ▶ Corrected the wording from `INITAL` to `INITIAL` of the receiver states
- ▶ Internal module improvement. This module version update does not affect module functionality
- ▶ Updated version macros
- ▶ Implemented usage of Unix line endings

#### **Module version 1.0.3**

2011-03-18

- ▶ Implemented code improvements regarding explicit casts

#### **Module version 1.0.2**

2011-02-28

- ▶ Removed unnecessary input parameter checks

#### **Module version 1.0.1**

2011-01-28

- ▶ Implemented an evaluation by the receiver if all bits in the payload data are set to 1 ("Ungültigkeitskennzeichnung" of application CRC)

#### **Module version 1.0.0**

2010-12-23

- ▶ Initial release

### **2.3.2.2. New features**

- ▶ E2EP01B: Implemented E2E Profile variant 1B (`DataIDMode: E2E_P01_DATAID_ALT`). Depending on parity of the counter (alternating ID configuration) the high or the low byte is included. For even counter values the low byte is included and for odd counter values the high byte is included.

### 2.3.2.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

- ▶ [ASCE2E-41] The receiver evaluates if all bits in the payload data are set to 1 (“Ungültigkeitskennzeichnung” of application CRC)

Description:

For more information, see the *Deviations* chapter.

- ▶ [ASCE2ESE-138] Re-synchronization handling in case the profile detects an unexpected behavior of the received counter

Description:

After a detected sequence error, e.g. due to a message loss or a reset of the sender, the library enters a synchronization state until the communication is error-free in order to provide a deterministic behavior. This behavior is configurable according to the *Specification of SW-C End-to-End Communication Protection Library, V3.0.0, R4.1 Rev 1*. For more information, see [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=52362](http://www.autosar.org/bugzilla/show_bug.cgi?id=52362).

Requirements:

E2E0018, E2E0022, E2E0196, E2E0021

- ▶ [ASCE2E-204] Support for Profile 01 variant C

Description:

Provision of a new profile variant 1C which uses a 12-bit DataId where the higher 4 bits are explicitly transmitted and checked by the receiver according to the *Specification of SW-C End-to-End Communication Protection Library, V3.0.0, R4.1 Rev 1*. For more information, see [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=53619](http://www.autosar.org/bugzilla/show_bug.cgi?id=53619)) and [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=59003](http://www.autosar.org/bugzilla/show_bug.cgi?id=59003).

Rationale:

Existing Profile 01 variants do not have sufficient error detection for masquerading failures if more than 255 values are used in a given system.

Requirements:

E2E0018, E2E0196, E2E0085, E2E0163, E2E0200, E2E0195

- ▶ Support for the usage with AUTOSAR E2E Transformer

Description:

In addition to AUTOSAR version R4.0 Rev 3, the following APIs are provided according to AUTOSAR release 4.2.1 in a backward compatible way:

- ▶ `E2E_P01Protect()`
- ▶ `E2E_P01Check()`
- ▶ `E2E_P01ProtectInit()`
- ▶ `E2E_P01CheckInit()`
- ▶ `E2E_P01MapStatusToSM()`

▶ **Deterministic initialization behavior for E2E State Machine**

**Description:**

In addition to AUTOSAR release 4.2.1, Bugzilla RfC 67553 is incorporated. That is, a new generic profile status type `E2E_P_NONEWDATA` is introduced if no new data is received in the actual receive cycle. For more information, see [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=67553](http://www.autosar.org/bugzilla/show_bug.cgi?id=67553).

▶ **[ASCE2E-246] Support of 256 bytes in Profile 01**

**Description:**

Provision of the possibility to configure `E2E_P01ConfigType DataLength` to up to 2048 bits (including control fields).

**Rationale:**

There may be use cases for a `DataLength` of more than 32 Bytes. Therefore, up to 2048 Bits are supported, such as in P02. ASR UC\_E2E\_00051 recommends a maximum length of 32 Bytes for the E2E Profile 01. The User is responsible if the `DataLength` configuration exceeds 32 Bytes.

**Requirements:**

E2E0018

▶ **[ASCE2E-894] Support for Profile 01 variant B**

**Description:**

Provision of a new profile variant 1B which uses a 16-bit `DataId` where depending on parity of the counter (alternating ID configuration) the high and the low byte is included. For even counter values the low byte is included and for odd counter values the high byte is included.

**Requirements:**

E2E0018, E2E0196, E2E0085, E2E0163, E2E0200, E2E0195

## 2.3.2.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

- ▶ [ASCE2E-34] The E2E Library uses the Safety CRC module `SCrc` instead of the AUTOSAR `Crc` module

Description:

Instead of the standard AUTOSAR `Crc` module, a Safety CRC module named `SCrc` is used. That is, instead of `Crc_CalculateCRC8()` and `Crc_CalculateCRC8H2F()`, the functions `SCrc_CalculateCRC8()` and `SCrc_CalculateCRC8H2F()` of the `SCrc` module are called.

Rationale:

The implementation of the `SCrc` library complies with the requirements for the development of safety-related software for the automotive domain.

Requirements:

E2E0114, E2E0190

- ▶ [ASCE2E-41] The receiver evaluates if all bits in the payload data are set to 1 (“Ungültigkeitskennzeichnung” of application CRC)

Description:

The functional behavior of the `E2E_P01Check()` routine is stated in requirement E2E0196 (AUTOSAR SWS of the E2E Library) and does not consider *data invalid* recognition. This behavior is modified in a way that the evaluation of the payload data without CRC is done as follows:

- ▶ 1) The receiver checks the payload data to contain completely binary ones.
- ▶ 2a) If so, the data is invalid and no further checks are performed.
- ▶ 2b) If the data does not contain completely binary ones, the data is checked as specified by the Profile 01.

As a consequence, the application is informed if *data invalid* is recognized.

Rationale:

This is a requirement stated in the communication specification document (*Lastenheft Requirements Specification: Bordnetzkommunikation; BMW Group; Requirement Id: BN\_KOM\_30\_569; 11.11.2009*).

Requirements:

E2E0196

- ▶ [ASCE2E-8] Configuration pointer is a pointer to `const`

Description:

In contrast to E2E0166 and E2E0158, the first argument of `E2E_P01Protect()` and the first and third argument of `E2E_P01Check()` are pointers to `const`:

```
Std_ReturnType E2E_P01Protect(const E2E_P01ConfigType* Config, E2E_P01SenderStateType*State, uint8*Data);  
  
Std_ReturnType E2E_P01Check(const E2E_P01ConfigType* Config, E2E_P01ReceiverStateType*State, const uint8*Data);
```

See also [https://www.autosar.org/bugzilla/show\\_bug.cgi?id=68903](https://www.autosar.org/bugzilla/show_bug.cgi?id=68903).

Rationale:

The E2E configuration is always constant and no write access is performed on it. Also the data must not be modified by `E2E_P01Check()`, so no write access is performed on it.

Requirements:

E2E0166, E2E0158

### 2.3.2.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

For this module no constraints and limitations are known.

### 2.3.2.6. Open-source software

E2EP01 does not use open-source software.

## 2.3.3. SCrc module release notes

- ▶ Module version: 2.0.11.B439717
- ▶ Supplier: Elektrobit Automotive GmbH

### 2.3.3.1. Change log

This chapter lists the changes between different versions.

**Module version 2.0.11**

2021-03-05

- ▶ Updated preprocessor include guards to be PC-lint compatible
- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.0.10**

2020-06-19

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.0.9**

2019-10-11

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.0.8**

2019-06-14

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.0.7**

2019-02-15

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.0.6**

2018-10-26

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.0.5**

2018-02-01

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 2.0.4**

2017-09-22



- ▶ Switch from MISRA-C:2004 to MISRA-C:2012

#### **Module version 2.0.3**

2015-06-19

- ▶ Added CRC routines required for new E2E Profiles 4, 5, 6

#### **Module version 2.0.2**

2013-11-29

- ▶ Non-functional code improvements (Inspection findings, Consistent use of static qualifier).
- ▶ ASCSCRC-74 Fixed known issue: The SCrc module used a compiler abstraction macro of the Crc module.

#### **Module version 2.0.1**

2013-02-15

- ▶ Provide a Basic Software Module Description that specifies the Memory Mappings.

#### **Module version 2.0.0**

2012-06-22

- ▶ Update to ASR 4.0 Rev3 (New parameter isFirstCall in CalculateCRC APIs).

#### **Module version 1.0.8**

2011-09-16

- ▶ Non-functional code improvements.

#### **Module version 1.0.7**

2011-05-20

- ▶ Usage of unix line endings.

#### **Module version 1.0.6**

2011-04-20

- ▶ Correction of common published information.
- ▶ Correction of compiler abstraction.

#### **Module version 1.0.5**

2011-03-18

- ▶ Macro definitions for switching between Crc implementation of SCrc module or external Crc module.

#### **Module version 1.0.4**

2011-02-28

- ▶ The GetVersionInfoApi() removed from the SCrc module.

#### **Module version 1.0.3**

2011-02-14

- ▶ Elimination of misra violations.

#### **Module version 1.0.2**

2010-12-23

- ▶ Table-based 8-bit Crc routine according to SAE J1850 CRC.

#### **Module version 1.0.1**

2010-11-30

- ▶ Removed unintended header file in Crc.c.

#### **Module version 1.0.0**

2010-11-12

- ▶ Table-based 8-bit Crc routine with generator polynomial 0x2F.

### **2.3.3.2. New features**

- ▶ No new features have been added since the last release.

### 2.3.3.3. EB-specific enhancements

This module is not part of the AUTOSAR specification.

### 2.3.3.4. Deviations

This module is not part of the AUTOSAR specification.

### 2.3.3.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- For this module no limitations are known.

### 2.3.3.6. Open-source software

Open-source software information is not available for this module.

## 3. E2EP01 user's guide

### 3.1. Overview

This user's guide describes the E2EP01 module. From this user's guide you learn the basic functionality of the E2EP01. You also learn which related modules are necessary to configure the E2EP01 module. The E2EP01 module reference provides further information on how to configure the E2EP01 itself.

Note that this user's guide is intended for readers who have good knowledge of AUTOSAR and about the purpose of the E2EP01. The information provided here helps you to integrate the E2EP01 in your AUTOSAR project.

- ▶ [Section 3.2, “Background information”](#) provides an overview of the basic functionality of the E2EP01.
- ▶ [Section 3.3, “Configuring E2EP01”](#) provides information on related modules that are needed in order to configure the E2EP01.
- ▶ [Section 3.4, “E2EP01 integration notes”](#) provides notes for the integration of the E2EP01 module into your project.
- ▶ For details on how to configure the E2EP01 itself, see the parameter descriptions provided in the E2EP01 module reference [Chapter 4, “E2EP01 module references”](#), which is provided together with the dependent modules E2E and SCrc.

### 3.2. Background information

The general concept of end-to-end communication protection is described in the EB tresos E2E Protection Wrapper documentation of the E2EPW module.

#### 3.2.1. Functional overview

##### 3.2.1.1. Safety mechanisms

This profile is based on E2E Profile 1 specified by AUTOSAR, see [\[1\]](#) and [\[2\]](#). E2EPW uses the following safety mechanisms:

- ▶ **Cyclic redundancy check (CRC):** An 8-bit CRC is explicitly sent with polynomial  $0x1D$  with an initial value  $0x00$  and a final XOR-value  $0x00$ . The bit-offset of the CRC value within a transmitted signal group is configurable but must be byte-aligned.
- ▶ **Sequence counter/alive counter:** A 4-bit sequence number with a counter that represents numbers from 0 to 14 is explicitly sent and incremented at every transmission request. The bit-offset of the sequence counter/alive counter value within a transmitted signal group is configurable but must be aligned with respect to nibbles.
- ▶ **System-wide unique 16-bit data ID for every port data element sent over a port:** The following data ID inclusion modes can be configured:
  - ▶ **Both bytes** (`dataIdMode=0`): Both bytes of the 16-bit data ID are attached to the safety data for CRC calculation, but not explicitly sent.
  - ▶ **Low byte only** (`dataIdMode=2`): Only the low byte of the 16-bit data ID is attached to the safety data for CRC calculation, but not explicitly sent, while the high byte is set to 0.
  - ▶ **Explicit transmission of data ID nibble** (`dataIdMode=3`): Both bytes of the 16-bit data ID are attached to the safety data for CRC calculation, but the low nibble of the high byte of the data ID is explicitly transmitted. Only 12 bits are used in this 16-bit data ID and the high nibble of the high byte is set to 0. The bit-offset of the data ID nibble value within a transmitted signal group is configurable, but must be aligned with respect to nibbles. To be able to use this data ID inclusion mode together with the data ID inclusion mode **Both bytes**, the CRC is calculated over the low byte of the data ID and the high byte which is set to 0. For more information on explicit transmission of data ID nibbles, see [Figure 3.4, “Layout of the protected message including control data \(CRC, SEQ\) with explicit transmission of data ID nibble \(`dataIdMode=3`\)”](#). The constraints specified in ASR\_E2EPW020377 must be adhered to.

Figure 3.1, “[Layout of the protected message including control data \(CRC, SEQ\) with 2-byte data ID \(`dataIdMode=0`\)](#)” shows the layout of the AUTOSAR E2E Profile 1 with a CRC offset of 0 bits and a sequence counter/alive counter offset of 8 bits for `dataIdMode=0`.

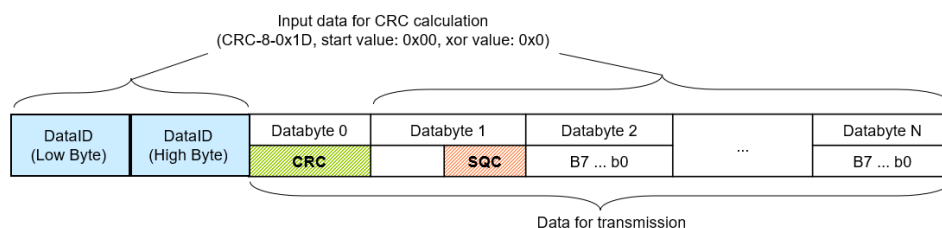


Figure 3.1. Layout of the protected message including control data (CRC, SEQ) with 2-byte data ID (`dataIdMode=0`)

Figure 3.2, “[Layout of the protected message including control data \(CRC, SEQ\) with 2-byte data ID \(`dataIdMode=1`\)](#)” shows the layout of the AUTOSAR E2E Profile 1 with a CRC offset of 0 bits and a sequence counter/alive counter offset of 8 bits for `dataIdMode=1`.

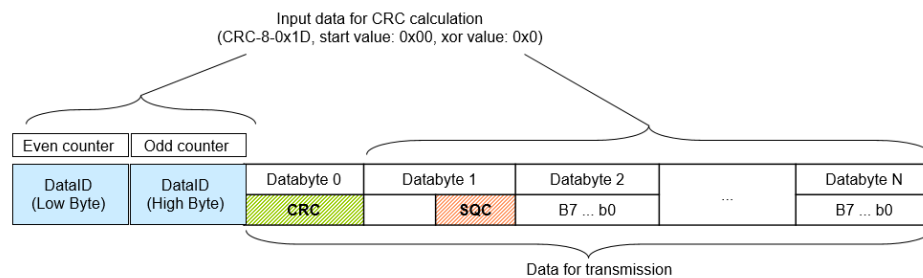


Figure 3.2. Layout of the protected message including control data (CRC, SEQ) with 2-byte data ID (`dataIdMode=1`)

Figure 3.3, “Layout of the protected message including control data (CRC, SEQ) with 1-byte data ID (`dataIdMode=2`)” shows the layout of the AUTOSAR E2E Profile 1 with a CRC offset of 0 bits and a sequence counter/alive counter offset of 8 bits for `dataIdMode=2`.

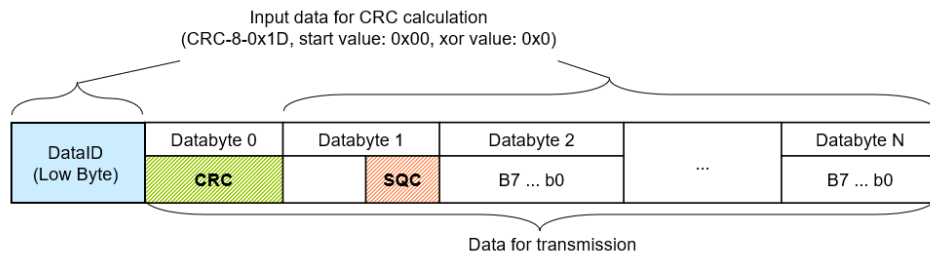


Figure 3.3. Layout of the protected message including control data (CRC, SEQ) with 1-byte data ID (`dataIdMode=2`)

Figure 3.4, “Layout of the protected message including control data (CRC, SEQ) with explicit transmission of data ID nibble (`dataIdMode=3`)” shows the layout of the AUTOSAR E2E Profile 1 with a CRC offset of 0 bits and a sequence counter/alive counter offset of 8 bits and a data ID nibble offset of 12 bits as used for `dataIdMode=3`.

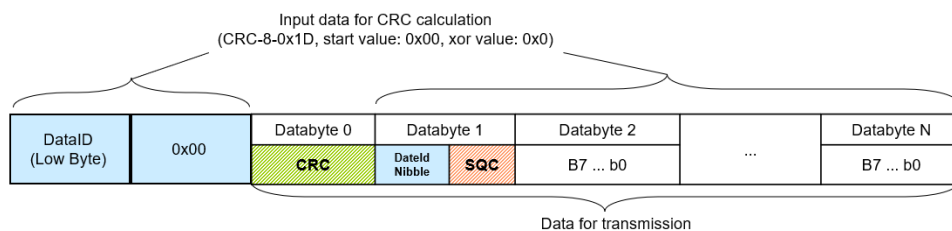


Figure 3.4. Layout of the protected message including control data (CRC, SEQ) with explicit transmission of data ID nibble (`dataIdMode=3`)

### 3.2.1.2. Failure modes and required safety mechanisms

The serialization of the application data is equal to the layout of the corresponding signal group for all variants. The [Table 3.1, “Failure modes detection matrix for E2E Profile 1”](#) shows the failure modes and the required safety mechanisms of E2E Profile 1 with the different data ID variants for detection of the failure mode.

#### NOTE



#### Different data ID inclusion modes

The different data ID inclusion modes only limits the applicable range of data IDs which can be used to detect masquerading.

An **X** specifies that the failure mode can be detected by the safety mechanism implemented in the E2E Profile.

An **(X)** specifies a safety mechanism which is only required to implement another safety mechanism.

An **A** specifies that the failure mode can be detected by a safety mechanism implemented in the data sink.

A **W** specifies that the failure mode can be detected by the safety mechanism implemented in the product EB tresos E2E Protection Wrapper. If this module is used directly in the Software Component or is used together with a different product, e.g. EB tresos E2E Protection Transformer, ensure that either this safety mechanism is not required or already implemented in a different module, e.g. `Rte`.

Failure mode/safety mechanism	Sequence counter	CRC	Data ID	Timeout detection	Range check
Unintended message repetition	X				
Message loss	X			A	
Insertion of message	X	(X)	X		
Resequencing	X				
Message corruption		X			W
Delayed reception				A	
Addressing faults	(X)	(X)	X		
Masquerading	(X)	(X)	X		

Table 3.1. Failure modes detection matrix for E2E Profile 1

## 3.3. Configuring E2EP01

To configure the `E2EP01` module, add the module to your project using EB tresos Studio. This module does not provide any configuration parameters except a common published information. You find this information in

the module references section of this document. You also find these in the parameter description in EB tresos Studio.

To use the `E2EP01` module, you must configure additional modules as outlined below:

- ▶ The `E2EP01` module requires API functions and data types from the `E2E` library module. This module does not provide any configuration parameters.
- ▶ The `E2EP01` module requires API functions and data types from the `SCrc` library module. This module does not provide any configuration parameters.
- ▶ The `E2EP01` module provides API functions and data types required either from the `E2EPW` module or from the `E2ESM` and `E2EXf` module. For more information on the `E2ESM` and `E2EXf` modules, see [\[2\]](#) and [\[3\]](#).

## 3.4. E2EP01 integration notes

You find general integration information in the EB tresos AutoCore Generic documentation.

In addition, you find module-specific information about exclusive areas, production errors and memory mapping in the module-specific integration notes. You find the module-specific integration notes in the module references chapter of this document. See [Chapter 4, “E2EP01 module references”](#) sub-section `Integration notes` in each module.



## 4. E2EP01 module references

### 4.1. Overview

This chapter provides module references for the E2EP01 product modules. These include a detailed description of all configuration parameters. Furthermore this chapter lists the application programming interface with all data types, constants and functions.

The content of the sections is sorted alphabetically according the EB tresos AutoCore Generic module names.

For further information on the functional behavior of these modules, refer to the chapter E2EP01 user's guide.

#### 4.1.1. Notation in EB module references

EB notation may differ from the AUTOSAR standard notation in the software specification documents (SWS). This section describes the notation of *default value* and *range* fields in the EB module references.

##### 4.1.1.1. Default value of configuration parameters

If there is no default value specified for a parameter, the default value field is omitted to prevent ambiguity with parameters that have -- as default values.

Example: The parameter `BswMCompuConstText` of the `BswM` module of EB tresos AutoCore Generic 8 Mode Management has no default value field, therefore it is omitted.

##### 4.1.1.2. Range information of configuration parameters

The range of a configuration parameter contains an upper and a lower boundary. However, in special cases the range of allowed values can be computed by means of an XPath function that is evaluated at configuration time. An XPath function can either be a standard `xpath:<function>()` or a custom `cxpath:<function>()` function. The range of a configuration parameter may be computed based on other configuration parameters that are referenced from the XPath function. For more information on custom XPath functions, see section *Custom XPath Functions API* of the EB tresos Studio developer's guide.

Example: The parameter `BswMCompuConstText` of the `BswM` module of EB tresos AutoCore Generic 8 Mode Management has the custom XPath function `cxpath:getCompuMethodsVT()` in the range field which provides the allowed values.

## 4.2. E2E

### 4.2.1. Configuration parameters

Containers included		
Container name	Multiplicity	Description
<a href="#">CommonPublishedInformation</a>	1..1	<b>Label:</b> Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
<a href="#">PublishedInformation</a>	1..1	<b>Label:</b> EB Published Information Additional published parameters not covered by Common-PublishedInformation container.

#### 4.2.1.1. CommonPublishedInformation

Parameters included	
Parameter name	Multiplicity
<a href="#">ArMajorVersion</a>	1..1
<a href="#">ArMinorVersion</a>	1..1
<a href="#">ArPatchVersion</a>	1..1
<a href="#">SwMajorVersion</a>	1..1
<a href="#">SwMinorVersion</a>	1..1
<a href="#">SwPatchVersion</a>	1..1
<a href="#">ModuleId</a>	1..1
<a href="#">VendorId</a>	1..1
<a href="#">VendorApiInfix</a>	1..1
<a href="#">Release</a>	1..1

Parameter Name	ArMajorVersion
<b>Label</b>	AUTOSAR Major Version
<b>Description</b>	Major version number of AUTOSAR specification on which the appropriate implementation is based on.
<b>Multiplicity</b>	1..1

Type	INTEGER_LABEL
Default value	4
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ArMinorVersion
Label	AUTOSAR Minor Version
Description	Minor version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	2
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ArPatchVersion
Label	AUTOSAR Patch Version
Description	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMajorVersion
Label	Software Major Version
Description	Major version number of the vendor specific implementation of the module.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	2
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMinorVersion
----------------	----------------

<b>Label</b>	Software Minor Version	
<b>Description</b>	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER_LABEL	
<b>Default value</b>	0	
<b>Configuration class</b>	<b>PublishedInformation:</b>	
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>SwPatchVersion</b>	
<b>Label</b>	Software Patch Version	
<b>Description</b>	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER_LABEL	
<b>Default value</b>	18	
<b>Configuration class</b>	<b>PublishedInformation:</b>	
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>ModuleId</b>	
<b>Label</b>	Numeric Module ID	
<b>Description</b>	Module ID of this module from Module List	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER_LABEL	
<b>Default value</b>	207	
<b>Configuration class</b>	<b>PublishedInformation:</b>	
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>VendorId</b>	
<b>Label</b>	Vendor ID	
<b>Description</b>	Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER_LABEL	

<b>Default value</b>	1
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>VendorApilnfix</b>
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING_LABEL

<b>Parameter Name</b>	<b>Release</b>
<b>Label</b>	Release Information
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING_LABEL
<b>Default value</b>	
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

#### 4.2.1.2. PublishedInformation

Parameters included	
Parameter name	Multiplicity
<a href="#">PbcfgMSupport</a>	1..1

<b>Parameter Name</b>	<b>PbcfgMSupport</b>
<b>Label</b>	PbcfgM support
<b>Description</b>	Specifies whether or not the E2E can use the PbcfgM module for post-build support.
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

#### 4.2.2. Application programming interface (API)

### 4.2.2.1. Type definitions

#### 4.2.2.1.1. E2E\_PCheckStatusType

<b>Purpose</b>	Status type for E2E State Machine return values.
<b>Type</b>	uint8

### 4.2.2.2. Macro constants

#### 4.2.2.2.1. E2E\_AR\_MAJOR\_VERSION

<b>Purpose</b>	AUTOSAR specification major version.
<b>Value</b>	4U

#### 4.2.2.2.2. E2E\_AR\_MINOR\_VERSION

<b>Purpose</b>	AUTOSAR specification minor version.
<b>Value</b>	2U

#### 4.2.2.2.3. E2E\_AR\_PATCH\_VERSION

<b>Purpose</b>	AUTOSAR specification patch version.
<b>Value</b>	1U

#### 4.2.2.2.4. E2E\_AR\_RELEASE\_MAJOR\_VERSION

<b>Purpose</b>	AUTOSAR release major version.
<b>Value</b>	4U

#### 4.2.2.2.5. E2E\_AR\_RELEASE\_MINOR\_VERSION

<b>Purpose</b>	AUTOSAR release minor version.
----------------	--------------------------------

<b>Value</b>	2U
--------------	----

#### 4.2.2.2.6. E2E\_AR\_RELEASE\_REVISION\_VERSION

<b>Purpose</b>	AUTOSAR release revision version.
<b>Value</b>	1U

#### 4.2.2.2.7. E2E\_E\_INPUTERR\_NULL

<b>Purpose</b>	At least one pointer parameter is a NULL pointer.
<b>Value</b>	0x13U

#### 4.2.2.2.8. E2E\_E\_INPUTERR\_WRONG

<b>Purpose</b>	At least one input parameter is erroneous (e.g. out of range).
<b>Value</b>	0x17U

#### 4.2.2.2.9. E2E\_E\_INTERR

<b>Purpose</b>	An internal library error has occurred.
<b>Value</b>	0x19U
<b>Description</b>	An internal library error has occurred. (e.g. error detected by program flow monitoring, violated invariant or postcondition)

#### 4.2.2.2.10. E2E\_E\_INVALID

<b>Purpose</b>	Invalid value passed to function.
<b>Value</b>	0xFFU

#### 4.2.2.2.11. E2E\_E\_OK

<b>Purpose</b>	Function completed successfully.
<b>Value</b>	0U

#### 4.2.2.2.12. E2E\_E\_WRONGSTATE

<b>Purpose</b>	Function executed in wrong state.
<b>Value</b>	0x1AU

#### 4.2.2.2.13. E2E\_MODULE\_ID

<b>Purpose</b>	AUTOSAR module identification.
<b>Value</b>	207U

#### 4.2.2.2.14. E2E\_P\_ERROR

<b>Purpose</b>	Error not related to counters occurred (e.g. wrong crc, wrong length, wrong DataID) or the return of the check function was not OK.
<b>Value</b>	0x03U
<b>Description</b>	Note: related to E2E_PCheckStatusType

#### 4.2.2.2.15. E2E\_P\_NONEWDATA

<b>Purpose</b>	No new Data is available since the last call of the check function.
<b>Value</b>	0x05U
<b>Description</b>	Note: related to E2E_PCheckStatusType

#### 4.2.2.2.16. E2E\_P\_NOTAVAILABLE

<b>Purpose</b>	No value has been received yet. This is used as the initialization value for the buffer, it is not returned by any E2E profile.
<b>Value</b>	0x04U
<b>Description</b>	Note: related to E2E_PCheckStatusType

#### 4.2.2.2.17. E2E\_P\_OK

<b>Purpose</b>	The checks of the Data in this cycle were successful (including counter check).
<b>Value</b>	0x00U



<b>Description</b>	Note: related to E2E_PCCheckStatusType
--------------------	--

#### 4.2.2.2.18. E2E\_P\_REPEATED

<b>Purpose</b>	Either no new data is available, or the new data has a repeated counter.
<b>Value</b>	0x01U
<b>Description</b>	Note: related to E2E_PCCheckStatusType

#### 4.2.2.2.19. E2E\_P\_WRONGSEQUENCE

<b>Purpose</b>	The checks of the Data in this cycle were successful, with the exception of counter jump, which changed more than allowed delta.
<b>Value</b>	0x02U
<b>Description</b>	Note: related to E2E_PCCheckStatusType

#### 4.2.2.2.20. E2E\_SW\_MAJOR\_VERSION

<b>Purpose</b>	AUTOSAR module major version.
<b>Value</b>	2U

#### 4.2.2.2.21. E2E\_SW\_MINOR\_VERSION

<b>Purpose</b>	AUTOSAR module minor version.
<b>Value</b>	0U

#### 4.2.2.2.22. E2E\_SW\_PATCH\_VERSION

<b>Purpose</b>	AUTOSAR module patch version.
<b>Value</b>	18U

#### 4.2.2.2.23. E2E\_VENDOR\_ID

<b>Purpose</b>	AUTOSAR vendor identification: Elektrobit Automotive GmbH.
<b>Value</b>	1U

### 4.2.2.3. Functions

#### 4.2.2.3.1. E2E\_GetVersionInfo

<b>Purpose</b>	Return the modules version information.	
<b>Synopsis</b>	<code>void E2E_GetVersionInfo ( Std_VersionInfoType * VersionInfo );</code>	
<b>Service ID</b>	0x14	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (out)</b>	VersionInfo	Pointer where to store the version information of this module
<b>Description</b>	<p>This service returns the version information of this module. The version information includes:</p> <ul style="list-style-type: none"><li>▶ Module Id</li><li>▶ Vendor Id</li><li>▶ Vendor specific version numbers</li></ul>	

### 4.2.3. Integration notes

#### 4.2.3.1. Exclusive areas

Exclusive areas are not used by the E2E module.

#### 4.2.3.2. Production errors

Production errors are not reported by the E2E module.

#### 4.2.3.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section `Memory mapping and compiler abstraction` in the `Integration notes` section for details.

The following table provides the list of sections that may be mapped for this module:

Memory section
CODE
CONST_8
CONST_16
CONST_32
CONST_UNSPECIFIED

#### 4.2.3.4. Integration requirements

##### WARNING



##### Integration requirements list is not exhaustive

The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

Integration requirements are not listed for the E2E module.

## 4.3. E2EP01

### 4.3.1. Configuration parameters

Containers included		
Container name	Multiplicity	Description
<a href="#">CommonPublishedInformation</a>	1..1	<b>Label:</b> Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
<a href="#">PublishedInformation</a>	1..1	<b>Label:</b> EB Published Information Additional published parameters not covered by Common-PublishedInformation container.

#### 4.3.1.1. CommonPublishedInformation

Parameters included	
Parameter name	Multiplicity

Parameters included	
<a href="#">ArMajorVersion</a>	1..1
<a href="#">ArMinorVersion</a>	1..1
<a href="#">ArPatchVersion</a>	1..1
<a href="#">SwMajorVersion</a>	1..1
<a href="#">SwMinorVersion</a>	1..1
<a href="#">SwPatchVersion</a>	1..1
<a href="#">ModuleId</a>	1..1
<a href="#">VendorId</a>	1..1
<a href="#">VendorApilInfix</a>	1..1
<a href="#">Release</a>	1..1

Parameter Name	ArMajorVersion	
Label	AUTOSAR Major Version	
Description	Major version number of AUTOSAR specification on which the appropriate implementation is based on.	
Multiplicity	1..1	
Type	INTEGER_LABEL	
Default value	4	
Configuration class	PublishedInformation:	
Origin	Elektrobit Automotive GmbH	

Parameter Name	ArMinorVersion	
Label	AUTOSAR Minor Version	
Description	Minor version number of AUTOSAR specification on which the appropriate implementation is based on.	
Multiplicity	1..1	
Type	INTEGER_LABEL	
Default value	2	
Configuration class	PublishedInformation:	
Origin	Elektrobit Automotive GmbH	

Parameter Name	ArPatchVersion	
Label	AUTOSAR Patch Version	

<b>Description</b>	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER_LABEL	
<b>Default value</b>	1	
<b>Configuration class</b>	<b>PublishedInformation:</b>	
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>SwMajorVersion</b>	
<b>Label</b>	Software Major Version	
<b>Description</b>	Major version number of the vendor specific implementation of the module.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER_LABEL	
<b>Default value</b>	2	
<b>Configuration class</b>	<b>PublishedInformation:</b>	
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>SwMinorVersion</b>	
<b>Label</b>	Software Minor Version	
<b>Description</b>	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER_LABEL	
<b>Default value</b>	2	
<b>Configuration class</b>	<b>PublishedInformation:</b>	
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>SwPatchVersion</b>	
<b>Label</b>	Software Patch Version	
<b>Description</b>	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER_LABEL	
<b>Default value</b>	17	

<b>Configuration class</b>	<b>PublishedInformation:</b>	
<b>Origin</b>	Elektrobit Automotive GmbH	

Parameter Name	ModuleId	
<b>Label</b>	Numeric Module ID	
<b>Description</b>	Module ID of this module from Module List	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER_LABEL	
<b>Default value</b>	0	
<b>Configuration class</b>	<b>PublishedInformation:</b>	
<b>Origin</b>	Elektrobit Automotive GmbH	

Parameter Name	VendorId	
<b>Label</b>	Vendor ID	
<b>Description</b>	Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER_LABEL	
<b>Default value</b>	1	
<b>Configuration class</b>	<b>PublishedInformation:</b>	
<b>Origin</b>	Elektrobit Automotive GmbH	

Parameter Name	VendorApilInfix	
<b>Multiplicity</b>	1..1	
<b>Type</b>	STRING_LABEL	

Parameter Name	Release	
<b>Label</b>	Release Information	
<b>Multiplicity</b>	1..1	
<b>Type</b>	STRING_LABEL	
<b>Default value</b>		
<b>Configuration class</b>	<b>PublishedInformation:</b>	
<b>Origin</b>	Elektrobit Automotive GmbH	

#### 4.3.1.2. PublishedInformation

Parameters included	
Parameter name	Multiplicity
<a href="#">PbcfgMSupport</a>	1..1

Parameter Name	PbcfgMSupport
Label	PbcfgM support
Description	Specifies whether or not the E2EP01 can use the PbcfgM module for post-build support.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

### 4.3.2. Application programming interface (API)

#### 4.3.2.1. Type definitions

##### 4.3.2.1.1. E2E\_P01CheckStateType

<b>Purpose</b>	Definition of E2E Profile 1 check state type.
<b>Type</b>	<a href="#">E2E_P01ReceiverStateType</a>
<b>Description</b>	Type-definition for backward-compatibility due to type renaming done in AUTOSAR version 4.2.1.

##### 4.3.2.1.2. E2E\_P01CheckStatusType

<b>Purpose</b>	Definition of E2E Profile 1 check status type.
<b>Type</b>	<a href="#">E2E_P01ReceiverStatusType</a>
<b>Description</b>	Type-definition for backward-compatibility due to type renaming done in AUTOSAR version 4.2.1.

#### 4.3.2.1.3. E2E\_P01ConfigType

<b>Purpose</b>	Configuration of transmitted Data for E2E Profile 1.	
<b>Type</b>	struct	
<b>Members</b>	uint16 CounterOffset	Bit offset of Counter (Least Significant Bit) from the beginning of the Byte-Array (bit numbering: bit 0 is the least important). The offset shall be a multiple of 4. For example, offset 8 means that the Counter will take the low nibble of the byte 1, i.e. bits 8..11.
	uint16 CRCOffset	Bit offset of CRC (Least Significant Bit) from the beginning of the Byte-Array (bit numbering: bit 0 is the least important). The offset shall be a multiple of 8. For example, offset 8 means that the CRC will take the byte 1, i.e. bits 8..15.
	uint16 DataID	A unique identifier for protection against masquerading.
	uint16 DataIDNibbleOffset	Bit offset of the low nibble (Least Significant Bit) of the high byte of Data ID from the beginning of the Byte-Array (bit numbering: bit 0 is the least important). The offset shall be a multiple of 4. For example, offset 8 means that the DataIDNibble will take the low nibble of the byte 1, i.e. bits 8..11.
	E2E_P01DataIDMode DataIDMode	Inclusion mode of ID in CRC computation.
	uint16 DataLength	Length of the data in bits. The value shall be a multiple of 8 and shall be <= 2048.
	uint8 MaxDeltaCounterInit	Initial maximum allowed gap between two counter values of two consecutively received valid Data.
	uint8 MaxNoNewOrRepeatedData	Maximum amount of missing or repeated Data which the receiver does not expect to exceed under normal communication conditions.
	uint8 SyncCounterInit	Number of Data required for validating the consistency of the counter that must be received with a valid counter after the de-



		tection of an unexpected behavior of a received counter.
<b>Description</b>	Configuration of transmitted Data (Data Element or I-PDU), for E2E Profile 1. For each transmitted Data, there is an instance of this typedef.	

#### 4.3.2.1.4. E2E\_P01DataIDMode

<b>Purpose</b>	Inclusion modes of Data ID for E2E Profile 1.	
<b>Type</b>	enum	
<b>Constants</b>	E2E_P01_DATAID_BOTH	Two bytes are included in the CRC .
	E2E_P01_DATAID_ALT	One of the two bytes is included, alternating high and low byte.
	E2E_P01_DATAID_LOW	Only low byte is included, high byte is never used.
	E2E_P01_DATAID_NIBBLE	The low byte is included in the implicit CRC calculation, the low nibble of the high byte is explicitly transmitted along with the data, the high nibble of the high byte is not used.
<b>Description</b>	The Data ID is two bytes long in E2E Profile 1. There are three inclusion modes how the implicit two-byte Data ID is included in the one-byte CRC.	

#### 4.3.2.1.5. E2E\_P01ProtectStateType

<b>Purpose</b>	Definition of E2E Profile 1 protect state type.	
<b>Type</b>	<a href="#">E2E_P01SenderStateType</a>	
<b>Description</b>	Type-definition for backward-compatibility due to type renaming done in AUTOSAR version 4.2.1.	

#### 4.3.2.1.6. E2E\_P01ReceiverStateType

<b>Purpose</b>	Definition of E2E Profile 1 receiver state type.	
<b>Type</b>	struct	
<b>Members</b>	E2E_P01ReceiverStatusType Status	Result of the verification of the Data, determined by the Check function.
	uint8 LastValidCounter	Counter of last valid received message.

	<code>uint8 MaxDeltaCounter</code>	MaxDeltaCounter specifies the maximum allowed difference between two counter values of consecutively received valid messages.
	<code>boolean WaitForFirstData</code>	If true means that no correct data (with correct Data ID and CRC) has been yet received after the receiver (re-)initialization.
	<code>boolean NewDataAvailable</code>	Indicates to E2E Library that a new data is available for the Library to be checked. This attribute is set by the E2E Library caller and not by the E2E Library.
	<code>uint8 LostData</code>	Number of data (messages) lost since reception of last valid one.
	<code>uint8 NoNewOrRepeatedData-Counter</code>	Amount of consecutive reception cycles in which either (1) there was no new data, or (2) when the data was repeated.
	<code>uint8 SyncCounter</code>	Number of consecutively received valid Data required for determining the consistency of the counter.
<b>Description</b>	State of the sender for a Data protected with E2E Profile 1.	

#### 4.3.2.1.7. E2E\_P01ReceiverStatusType

<b>Purpose</b>	Definition of E2E Profile 1 receiver status type.	
<b>Type</b>	enum	
<b>Constants</b>	<code>E2E_P01STATUS_OK</code>	New data has been correctly received.
	<code>E2E_P01STATUS_NONEWDATA</code>	The Check function has been invoked but new Data is not available since the last call.
	<code>E2E_P01STATUS_WRONGCRC</code>	The data has been received according to communication medium, but the CRC is incorrect.
	<code>E2E_P01STATUS_SYNC</code>	New data has been received after detection of an unexpected behavior of counter.
	<code>E2E_P01STATUS_INITIAL</code>	New data has been correctly received, but this is the first Data since the receiver's (re-)initialization.

	E2E_P01STATUS_REPEATED	New data has been correctly received, but the Counter is identical to the most recent Data received with Status _INITIAL, _OK, or _OKSOMELOST.
	E2E_P01STATUS_OKSOMELOST	New data has been correctly received, but some data in the sequence have been probably lost.
	E2E_P01STATUS_WRONGSEQUENCE	The new data has been correctly received, but the Counter Delta is too big (DeltaCounter > MaxDeltaCounter)
	E2E_P01STATUS_DATAINVALID	All bits in the received data except for byte 0 (CRC) are set to one.
<b>Description</b>	Result of the verification of the Data in E2E Profile 1, determined by the Check function.	

#### 4.3.2.1.8. E2E\_P01SenderStateType

<b>Purpose</b>	State of the sender for a Data protected with E2E Profile 1.	
<b>Type</b>	struct	
<b>Members</b>	uint8 Counter	Counter to be used for protecting the Data. The counter is incremented modulo 15

#### 4.3.2.2. Macro constants

##### 4.3.2.2.1. E2EP01\_AR\_MAJOR\_VERSION

<b>Purpose</b>	AUTOSAR specification major version.
<b>Value</b>	4U

##### 4.3.2.2.2. E2EP01\_AR\_MINOR\_VERSION

<b>Purpose</b>	AUTOSAR specification minor version.
<b>Value</b>	2U

#### 4.3.2.2.3. E2EP01\_AR\_PATCH\_VERSION

<b>Purpose</b>	AUTOSAR specification patch version.
<b>Value</b>	1U

#### 4.3.2.2.4. E2EP01\_AR\_RELEASE\_MAJOR\_VERSION

<b>Purpose</b>	AUTOSAR release major version.
<b>Value</b>	4U

#### 4.3.2.2.5. E2EP01\_AR\_RELEASE\_MINOR\_VERSION

<b>Purpose</b>	AUTOSAR release minor version.
<b>Value</b>	2U

#### 4.3.2.2.6. E2EP01\_AR\_RELEASE\_REVISION\_VERSION

<b>Purpose</b>	AUTOSAR release revision version.
<b>Value</b>	1U

#### 4.3.2.2.7. E2EP01\_SW\_MAJOR\_VERSION

<b>Purpose</b>	AUTOSAR module major version.
<b>Value</b>	2U

#### 4.3.2.2.8. E2EP01\_SW\_MINOR\_VERSION

<b>Purpose</b>	AUTOSAR module minor version.
<b>Value</b>	2U

#### 4.3.2.2.9. E2EP01\_SW\_PATCH\_VERSION

<b>Purpose</b>	AUTOSAR module patch version.
----------------	-------------------------------

<b>Value</b>	17U
--------------	-----

#### 4.3.2.2.10. E2EP01\_VENDOR\_ID

<b>Purpose</b>	AUTOSAR vendor identification: Elektrobit Automotive GmbH.
<b>Value</b>	1U

### 4.3.2.3. Functions

#### 4.3.2.3.1. E2E\_P01Check

<b>Purpose</b>	Check the received Data using the E2E Profile P01.	
<b>Synopsis</b>	Std_ReturnType <b>E2E_P01Check</b> ( const E2E_P01ConfigType * Config , E2E_P01CheckStateType * State , const uint8 * Data );	
<b>Service ID</b>	0x02	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different communication data / states	
<b>Parameters (in)</b>	Config	Pointer to static configuration.
	Data	Pointer to received Data.
<b>Parameters (in,out)</b>	State	Pointer to port/data communication state.
<b>Return Value</b>	Function execution success status	
	E2E_E_INPUTERR_NULL	At least one pointer parameter is a NULL pointer.
	E2E_E_INPUTERR_WRONG	At least one input parameter is erroneous.
	E2E_E_OK	Function completed successfully.
<b>Description</b>	Checks the Data received using the E2E Profile P01. This includes CRC calculation, handling of Counter and Data ID.	

#### 4.3.2.3.2. E2E\_P01CheckInit

<b>Purpose</b>	Initializes the check state.
----------------	------------------------------

<b>Synopsis</b>	Std_ReturnType <b>E2E_P01CheckInit</b> ( E2E_P01CheckStateType * StatePtr );	
<b>Service ID</b>	0x1c	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different states	
<b>Parameters (out)</b>	StatePtr	Pointer to port/data communication state.
<b>Return Value</b>	Function execution success status	
	E2E_E_INPUTERR_NULL	NULL pointer passed.
	E2E_E_OK	Function completed successfully.
<b>Description</b>	Initializes the state structure by setting: LastValidCounter = 0 MaxDeltaCounter = 0 WaitForFirstData = E2EXF_WAIT_FOR_FIRST_DATA_INIT_VALUE NewDataAvailable = TRUE LostData = 0 Status = E2E_P01STATUS_NONEWDATA NoNewOrRepeatedDataCounter = 0 SyncCounter = 0.	

#### 4.3.2.3.3. E2E\_P01MapStatusToSM

<b>Purpose</b>	Maps the check status of Profile 1 to a generic check status.	
<b>Synopsis</b>	E2E_PCheckStatusType <b>E2E_P01MapStatusToSM</b> ( Std_ReturnType CheckReturn , E2E_P01CheckStatusType Status , boolean profileBehavior );	
<b>Service ID</b>	0x1d	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different status types	
<b>Parameters (in)</b>	CheckReturn	Return value of the E2E_P01Check function.
	Status	Status determined by E2E_P01Check function.
	profileBehavior	FALSE: check has the legacy behavior, before R4.2 TRUE: check behaves like new P4/P5/P6 profiles introduced in R4.2
<b>Return Value</b>	Profile-independent status of the reception on one single Data in one cycle.	
	E2E_P_OK	CheckReturn is E2E_E_OK and profileBehavior is TRUE and Status is E2E_P01STATUS_OK or E2E_P01STATUS_OKSOMELOST or E2E_P01STATUS_

		SYNC, or CheckReturn is E2E_E_OK and profileBehavior is FALSE and Status is E2E_P01STATUS_OK or E2E_P01STATUS_OKSOMELOST or E2E_P01STATUS_INITIAL.
	E2E_P_ERROR	CheckReturn is E2E_E_OK and Status is E2E_P01STATUS_WRONGCRC, or CheckReturn is E2E_E_OK and Status is a not defined status, or CheckReturn does not equal E2E_E_OK.
	E2E_P_REPEATED	CheckReturn is E2E_E_OK and Status is E2E_P01STATUS_REPEATED.
	E2E_P_NONEWDATA	CheckReturn is E2E_E_OK and Status is E2E_P01STATUS_NONEWDATA.
	E2E_P_WRONGSEQUENCE	CheckReturn is E2E_E_OK and profileBehavior is TRUE and Status is E2E_P01STATUS_WRONGSEQUENCE or E2E_P01STATUS_INITIAL, or CheckReturn is E2E_E_OK and profileBehavior is FALSE and Status is E2E_P01STATUS_WRONGSEQUENCE or E2E_P01STATUS_SYNC.
<b>Description</b>	The function maps the check status of Profile 1 to a generic check status, which can be used by E2E state machine check function. The E2E Profile 1 delivers a more fine-granular status, but this is not relevant for the E2E state machine.	

#### 4.3.2.3.4. E2E\_P01Protect

<b>Purpose</b>	Protects the array/buffer to be transmitted using the E2E Profile P01.	
<b>Synopsis</b>	Std_ReturnType <b>E2E_P01Protect</b> ( const E2E_P01ConfigType * Config , E2E_P01ProtectStateType * State , uint8 * Data );	
<b>Service ID</b>	0x01	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different communication data / states	
<b>Parameters (in)</b>	Config	Pointer to static configuration.
<b>Parameters (in,out)</b>	State	Pointer to port/data communication state.
	Data	Pointer to Data to be protected.

<b>Return Value</b>	Function execution success status	
	E2E_E_INPUTERR_NULL	At least one pointer parameter is a NULL pointer.
	E2E_E_INPUTERR_WRONG	At least one input parameter is erroneous.
	E2E_E_OK	Function completed successfully.
<b>Description</b>	Protects the array/buffer to be transmitted using the E2E Profile P01. This includes checksum calculation, handling of counter and Data ID.	

#### 4.3.2.3.5. E2E\_P01ProtectInit

<b>Purpose</b>	Initializes the protection state.	
<b>Synopsis</b>	Std_ReturnType <b>E2E_P01ProtectInit</b> ( E2E_P01ProtectStateType * StatePtr );	
<b>Service ID</b>	0x1b	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different states	
<b>Parameters (out)</b>	StatePtr	Pointer to port/data communication state.
<b>Return Value</b>	Function execution success status	
	E2E_E_INPUTERR_NULL	NULL pointer passed.
	E2E_E_OK	Function completed successfully.
<b>Description</b>	Initializes the state structure by setting the Counter to 0.	

### 4.3.3. Integration notes

#### 4.3.3.1. Exclusive areas

Exclusive areas are not used by the E2EP01 module.

#### 4.3.3.2. Production errors

Production errors are not reported by the E2EP01 module.



### 4.3.3.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section `Memory mapping and compiler abstraction` in the `Integration notes` section for details.

The following table provides the list of sections that may be mapped for this module:

Memory section
CODE

### 4.3.3.4. Integration requirements

#### WARNING



#### Integration requirements list is not exhaustive

The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

Integration requirements are not listed for the E2EP01 module.

## 4.4. SCrc

### 4.4.1. Configuration parameters

Containers included		
Container name	Multiplicity	Description
<a href="#">CommonPublishedInformation</a>	1..1	<b>Label:</b> Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
<a href="#">PublishedInformation</a>	1..1	<b>Label:</b> EB Published Information Additional published parameters not covered by Common-PublishedInformation container.
<a href="#">SCrcPublishedInformation</a>	1..1	Additional published parameters not covered by Common-PublishedInformation container.

Containers included		
		Note that these parameters do not have any configuration class setting, since they are published information.

Parameters included	
Parameter name	Multiplicity
<a href="#">IMPLEMENTATION_CONFIG_VARIANT</a>	1..1

Parameter Name	IMPLEMENTATION_CONFIG_VARIANT
Label	Config Variant
Multiplicity	1..1
Type	ENUMERATION
Default value	VariantPreCompile
Range	VariantPreCompile

#### 4.4.1.1. CommonPublishedInformation

Parameters included	
Parameter name	Multiplicity
<a href="#">ArMajorVersion</a>	1..1
<a href="#">ArMinorVersion</a>	1..1
<a href="#">ArPatchVersion</a>	1..1
<a href="#">SwMajorVersion</a>	1..1
<a href="#">SwMinorVersion</a>	1..1
<a href="#">SwPatchVersion</a>	1..1
<a href="#">ModuleId</a>	1..1
<a href="#">VendorId</a>	1..1
<a href="#">Release</a>	1..1

Parameter Name	ArMajorVersion
Label	AUTOSAR Major Version
Description	Major version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL

<b>Default value</b>	0
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>ArMinorVersion</b>
<b>Label</b>	AUTOSAR Minor Version
<b>Description</b>	Minor version number of AUTOSAR specification on which the appropriate implementation is based on.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	0
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>ArPatchVersion</b>
<b>Label</b>	AUTOSAR Patch Version
<b>Description</b>	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	0
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SwMajorVersion</b>
<b>Label</b>	Software Major Version
<b>Description</b>	Major version number of the vendor specific implementation of the module.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	2
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SwMinorVersion</b>
-----------------------	-----------------------

<b>Label</b>	Software Minor Version	
<b>Description</b>	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER_LABEL	
<b>Default value</b>	0	
<b>Configuration class</b>	<b>PublishedInformation:</b>	
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>SwPatchVersion</b>	
<b>Label</b>	Software Patch Version	
<b>Description</b>	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER_LABEL	
<b>Default value</b>	11	
<b>Configuration class</b>	<b>PublishedInformation:</b>	
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>ModuleId</b>	
<b>Label</b>	Numeric Module ID	
<b>Description</b>	Module ID of this module from Module List	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER_LABEL	
<b>Default value</b>	0	
<b>Configuration class</b>	<b>PublishedInformation:</b>	
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>VendorId</b>	
<b>Label</b>	Vendor ID	
<b>Description</b>	Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER_LABEL	

<b>Default value</b>	1
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>Release</b>
<b>Label</b>	Release Information
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING_LABEL
<b>Default value</b>	
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

#### 4.4.1.2. PublishedInformation

Parameters included	
Parameter name	Multiplicity
<a href="#">PbcfgMSupport</a>	1..1

<b>Parameter Name</b>	<b>PbcfgMSupport</b>
<b>Label</b>	PbcfgM support
<b>Description</b>	Specifies whether or not the SCrc can use the PbcfgM module for post-build support.
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

#### 4.4.1.3. SCrcPublishedInformation

### 4.4.2. Application programming interface (API)

#### 4.4.2.1. Macro constants

##### 4.4.2.1.1. SCRC\_FUNCENABLED\_SCRC\_CRC16

<b>Purpose</b>	Defines whether or not SCrc_CalculateCRC16 shall map to <a href="#">SCrc_CalculateCRC16()</a> of this SCrc module (STD_ON) or shall map to an external function Crc_CalculateCRC16() declared in Crc.h (STD_OFF).
<b>Value</b>	STD_ON

##### 4.4.2.1.2. SCRC\_FUNCENABLED\_SCRC\_CRC32P4

<b>Purpose</b>	Defines whether or not SCrc_CalculateCRC32P4 shall map to <a href="#">SCrc_CalculateCRC32P4()</a> of this SCrc module (STD_ON) or shall map to an external function Crc_CalculateCRC32P4() declared in Crc.h (STD_OFF).
<b>Value</b>	STD_ON

##### 4.4.2.1.3. SCRC\_FUNCENABLED\_SCRC\_CRC8

<b>Purpose</b>	Defines whether or not SCrc_CalculateCRC8 shall map to <a href="#">SCrc_CalculateCRC8()</a> of this SCrc module (STD_ON) or shall map to an external function Crc_CalculateCRC8() declared in Crc.h (STD_OFF).
<b>Value</b>	STD_ON

##### 4.4.2.1.4. SCRC\_FUNCENABLED\_SCRC\_CRC8H2F

<b>Purpose</b>	Defines whether or not SCrc_CalculateCRC8H2F shall map to <a href="#">SCrc_CalculateCRC8H2F()</a> of this SCrc module (STD_ON) or shall map to an external function Crc_CalculateCRC8H2F() declared in Crc.h (STD_OFF).
<b>Value</b>	STD_ON

##### 4.4.2.1.5. SCRC\_SW\_MAJOR\_VERSION

<b>Purpose</b>	AUTOSAR module major version.
----------------	-------------------------------

<b>Value</b>	2U
--------------	----

#### 4.4.2.1.6. SCRC\_SW\_MINOR\_VERSION

<b>Purpose</b>	AUTOSAR module minor version.
<b>Value</b>	0U

#### 4.4.2.1.7. SCRC\_SW\_PATCH\_VERSION

<b>Purpose</b>	AUTOSAR module patch version.
<b>Value</b>	11U

#### 4.4.2.1.8. SCRC\_VENDOR\_ID

<b>Purpose</b>	AUTOSAR vendor identification: Elektrobit Automotive GmbH.
<b>Value</b>	1U

### 4.4.2.2. Functions

#### 4.4.2.2.1. SCrc\_CalculateCRC16

<b>Purpose</b>	Calculation of CRC16.	
<b>Synopsis</b>	<pre>uint16 SCrc_CalculateCRC16 ( const uint8 * SCrc_DataPtr , uint32 SCrc_Length , uint16 SCrc_StartValue16 , boolean SCrc_ IsFirstCall );</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SCrc_DataPtr	Valid pointer to start address of data block
	SCrc_Length	Length of data block in bytes
	SCrc_StartValue16	Initial Value
	SCrc_IsFirstCall	TRUE: First call in a sequence or individual CRC calculation; start from initial

		value, ignore Crc_StartValue16. FALSE: Subsequent call in a call sequence; Crc_StartValue16 is interpreted to be the return value of the previous function call.
<b>Return Value</b>	calculated CRC16 value	
<b>Description</b>	This function performs the calculation of a 16-bit CRC value over the memory block referenced by SCrc_DataPtr of byte length SCrc_Length.	

#### 4.4.2.2.2. SCrc\_CalculateCRC32P4

<b>Purpose</b>	Calculation of CRC32P4.	
<b>Synopsis</b>	<pre>uint32 SCrc_CalculateCRC32P4 ( const uint8 * SCrc_DataPtr , uint32 SCrc_Length , uint32 SCrc_StartValue32P4 , boolean SCrc_IsFirstCall );</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SCrc_DataPtr	Valid pointer to start address of data block
	SCrc_Length	Length of data block in bytes
	SCrc_StartValue32P4	Initial Value
	SCrc_IsFirstCall	TRUE: First call in a sequence or individual CRC calculation; start from initial value, ignore Crc_StartValue8. FALSE: Subsequent call in a call sequence; Crc_StartValue32P4 is interpreted to be the return value of the previous function call.
<b>Return Value</b>	calculated CRC32 value	
<b>Description</b>	This function performs the calculation of a 32-bit CRC value for Ethernet over the memory block referenced by SCrc_DataPtr of byte length SCrc_Length.	

#### 4.4.2.2.3. SCrc\_CalculateCRC8

<b>Purpose</b>	Calculation of CRC8.	
<b>Synopsis</b>	<pre>uint8 SCrc_CalculateCRC8 ( const uint8 * SCrc_DataPtr , uint32 SCrc_Length , uint8 SCrc_StartValue8 , boolean SCrc_IsFirstCall );</pre>	



<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SCrc_DataPtr	Valid pointer to start address of data block
	SCrc_Length	Length of data block in bytes
	SCrc_StartValue8	Initial Value
	SCrc_IsFirstCall	TRUE: First call in a sequence or individual CRC calculation; start from initial value, ignore Crc_StartValue8. FALSE: Subsequent call in a call sequence; Crc_StartValue8 is interpreted to be the return value of the previous function call.
<b>Return Value</b>	calculated CRC8 value	
<b>Description</b>	This function performs the calculation of a 8-bit SAE J1850 CRC value over the memory block referenced by SCrc_DataPtr of byte length SCrc_Length.	

#### 4.4.2.2.4. SCrc\_CalculateCRC8H2F

<b>Purpose</b>	Calculation of CRC8 with the Polynomial 0x2F.	
<b>Synopsis</b>	<pre>uint8 SCrc_CalculateCRC8H2F ( const uint8 * SCrc_DataPtr , uint32 SCrc_Length , uint8 SCrc_StartValue8H2F , boolean SCrc_IsFirstCall );</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SCrc_DataPtr	Valid pointer to start address of data block
	SCrc_Length	Length of data block in bytes
	SCrc_StartValue8	Initial Value
	SCrc_IsFirstCall	TRUE: First call in a sequence or individual CRC calculation; start from initial value, ignore Crc_StartValue8. FALSE: Subsequent call in a call sequence; Crc_StartValue8 is interpreted to be the return value of the previous function call.
<b>Return Value</b>	calculated CRC8 value	
<b>Description</b>	This function performs the calculation of a 8-bit CRC value with the Polynomial 0x2F over the memory block referenced by SCrc_DataPtr of byte length SCrc_Length.	

## 4.4.3. Integration notes

### 4.4.3.1. Exclusive areas

Exclusive areas are not used by the `SCrc` module.

### 4.4.3.2. Production errors

Production errors are not reported by the `SCrc` module.

### 4.4.3.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section `Memory mapping and compiler abstraction` in the `Integration notes` section for details.

The following table provides the list of sections that may be mapped for this module:

Memory section
CODE
CONST_8
CONST_16
CONST_32

### 4.4.3.4. Integration requirements

#### WARNING



#### Integration requirements list is not exhaustive

The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

Integration requirements are not listed for the `SCrc` module.

## 5. Bibliography

### Bibliography

- [1] *AUTOSAR Specification of SW-C End-to-End Communication Protection Library*, Issue Version 2.0.0, Release 4.0.3, Revision 0003, Publisher: AUTOSAR
- [2] *AUTOSAR Specification of SW-C End-to-End Communication Protection Library*, Issue AUTOSAR 4.2.1, Publisher: AUTOSAR
- [3] *AUTOSAR Specification of Module E2E Transformer*, Issue AUTOSAR 4.2.1 , Publisher: AUTOSAR