Elektrobit

EB tresos® E2E Profile 7 documentation

product release 8.8.5

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

Technical support

https://www.elektrobit.com/support

Legal disclaimer

Confidential information.

Table of Contents

1. Overview of EB tresos E2E Profile 7 documentation

Welcome to the EB tresos E2E Profile 7 (E2EP07) product documentation.

This document provides:

- Chapter 2, “E2EP07 release notes“: release notes for the E2EP07 modules
- Chapter 3, “E2EP07 user's guide“: containing background information and instructions
- Chapter 4, “E2EP07 module references“: information about configuration parameters and the application programming interface

2. E2EP07 release notes

2.1. Overview

This chapter provides the E2EP07 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: 28.2.0 b211016-0103

2.2.2. AUTOSAR modules

The following table lists the AUTOSAR modules that are part of this E2EP07 release.

Module name	AUTOSAR version and revision	SWS version and revision	Module version	Supplier
E2E	4.2.1 []	4.2.1 [0000]	2.0.18	Elektrobit Automotive GmbH
E2EP07	4.3.0 []	4.3.0 [0000]	1.0.12	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
No EB modules available		

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`[1]. 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.B520150
- 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

[1] `$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 .

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 contraints 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 contraints 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. E2EP07 module release notes

- AUTOSAR R4.3 Rev 0
- AUTOSAR SWS document version: 4.3.0
- Module version: 1.0.12.B520150
- Supplier: Elektrobit Automotive GmbH

2.3.2.1. Change log

This chapter lists the changes between different versions.

Module version 1.0.12
2021-03-05

- Incorporated RfC AR-88280: In Profile 7 CRC calculation, the ComputedCRC local variable is initialized incorrectly

- Updated preprocessor include guards to be PC-lint compatible

Module version 1.0.11
2020-10-23

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

Module version 1.0.10
2020-06-19

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

Module version 1.0.9
2020-01-24

- Implemented check for Offset to be not set beyond configured DataLength

Module version 1.0.8
2019-10-11

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

Module version 1.0.7
2019-06-14

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

Module version 1.0.6
2019-02-15

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

Module version 1.0.5
2018-10-26

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

Module version 1.0.4
2018-03-02

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

Module version 1.0.3
2017-10-27

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

Module version 1.0.2
2017-09-22

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

Module version 1.0.1
2017-08-25

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

Module version 1.0.0
2017-03-31

- Initial release

2.3.2.2. New features

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

2.3.2.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

- This module provides no EB-specific enhancements.

2.3.2.4. Deviations

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

- Typing error in requirement SWS_E2E_00550

  Description:

  In contrast to SWS_E2E_00550 the parameters `CheckReturn` and `Status` refer to the E2E_P07Check function.

  Rationale:

  Update requirement SWS_E2E_00550: In parameter status replace *Status determined by E2E_P04Check function* with *Status determined by E2E_P07Check function*. For more information, see https://www.autosar.org/bugzilla/show_bug.cgi?id=76737.

  Requirements:

  SWS_E2E_00550

- Discrepancy for minimal data length in Profile 07

  Description:

  In requirement SWS_E2E_00544 the member `MinDataLength` of `E2E_P07ConfigType` is described as *Minimal length of Data, in bits*. E2E checks that length is >= MinDataLength. The value shall be >= 18*8 and <= MaxDataLength. The value 18*8 refers to 18 bytes provided by the header as a minimum value. In contrast, the fixed header layout, as presented by Figure 7-16: E2E Profile 07 header, shows 20 bytes. So, there is a discrepancy of the value 18 used in the description of member `Offset` as well as `MinDataLength` of `E2E_P07ConfigType` and those provided with 20 bytes for the header layout.

  Rationale:

  Update requirement SWS_E2E_00544: `Offset`: Replace *MaxDataLength-(18*8)* with *MaxDataLength-(20*8)*. `MinDataLength`: Replace *shall be >= 18*8 and <= MaxDataLength* by *shall be >= 20*8 and <= MaxDataLength*. For more information, see https://www.autosar.org/bugzilla/show_bug.cgi?id=76738.

  Requirements:

  SWS_E2E_00544

- Return type of Profile 07 API shall be updated.

  Description:

There are conflicts in Profile 07 requirements which are as follows:

1. E2E_P07Protect() - Conflicts in the requirements [SWS_E2E_00546] and [SWS_E2E_00486]

i. [SWS_E2E_00546] : Return value of `E2E_P07Protect()` is void.

ii. [SWS_E2E_00486] : In the diagram, function `E2E_P07Protect()` returns with a value of `E2E_E_INPUTERR_NULL`, `E2E_E_OK` or `E2E_E_INPUTERR_WRONG`.

2. E2E_P07Check () - Conflicts in the requirement of [SWS_E2E_00548] and [SWS_E2E_00495]

i. [SWS_E2E_00548] : Return value of `E2E_P07Check()` is void.

ii. [SWS_E2E_00495] : In the diagram, the function `E2E_P07Check()` returns with a value of `E2E_E_INPUTERR_NULL`, `E2E_E_OK`, `E2E_E_INPUTERR_WRONG`.

3. E2E_P07ProtectInit() - Conflicts in the requirement of [SWS_E2E_00547] and [SWS_E2E_00551] function return value

i. [SWS_E2E_00547] : Return value of `E2E_P07ProtectInit()` is void.

ii. [SWS_E2E_00551] : If `State` is `NULL`, `E2E_P07ProtectInit` shall return immediately with `E2E_E_INPUTERR_NULL`.

4. E2E_P07CheckInit() - Conflicts in the requirement of [SWS_E2E_00549] and [SWS_E2E_00552]

i. [SWS_E2E_00549] : Return value of `E2E_P07CheckInit()` is void.

ii. [SWS_E2E_00552] : If `State` is `NULL`, `E2E_P07CheckInit` shall return immediately with `E2E_E_INPUTERR_NULL`.

5. E2E_P07MapStatusToSM() - Conflicts in the requirement of [SWS_E2E_00550] and [SWS_E2E_00553]

i. [SWS_E2E_00550] : Return value of `E2E_P07MapStatusToSM()` is void.

ii. [SWS_E2E_00553] : If `CheckReturn = E2E_E_OK`, then the function `E2E_P07MapStatusToSM` shall return the values depending on the value of `Status`.

For more information, see https://www.autosar.org/bugzilla/show_bug.cgi?id=76739.

Rationale:

Following functions shall have return type as `Std_ReturnType`:

`E2E_P07Protect`

`E2E_P07Check`

`E2E_P07ProtectInit`

`E2E_P07ProtectInit`

Requirements:

SWS_E2E_00546, SWS_E2E_00548, SWS_E2E_00547, SWS_E2E_00549, SWS_E2E_00550, SWS_E2E_00486, SWS_E2E_00495, SWS_E2E_00551, SWS_E2E_00552, SWS_E2E_00553

- Redundancy of Offset equaling zero in Profile 07

  Description:

  Requirement EB_E2EP07020492 incorporates requirement PRS_E2E_00492 of E2E Protocol Specification Release R19-11. PRS_E2E_00492 uses `Offset` of `E2E_P07ConfigType` for computation of the CRC, which is redundant in case the initial decision `Offset > 0` evaluates to `FALSE`. At this point `Offset` is already known as 0, which makes subtracting it from `Length` meaningless. Furthermore, the usage of `Offset` to access `Crc_DataPtr` at index `[offset+8]` is pointless as well.

  Rationale:

  If `Offset > 0` evaluates to `FALSE`:

  Replace `&Data[offset+8]` with `&Data[8]`.

  Replace `Length-Offset-8` with `Length-8`.

  Requirements:

  EB_E2EP07020492

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 limitations are known.

2.3.2.6. Open-source software

E2EP07 does not use open-source software.

3. E2EP07 user's guide

3.1. Overview

This user's guide describes the E2EP07 module. From this user's guide you learn the basic functionality of the E2EP07. You also learn which related modules are necessary to configure the E2EP07 module. The E2EP07 module reference provides further information on configuring the E2EP07 itself.

Note that this user's guide is intended for readers who have good knowledge of AUTOSAR and about the purpose of the E2EP07. The information provided here helps you to integrate the E2EP07 in your AUTOSAR project.

- Section 3.2, “Background information” provides an overview of the basic functionality of the E2EP07.
- Section 3.3, “Configuring E2EP07” provides information on related modules that are needed in order to configure the E2EP07.
- Section 3.4, “E2EP07 integration notes” provides notes for the integration of the E2EP07 module into your project.
- For details on how to configure the E2EP07 itself, see the parameter descriptions provided in the E2EP07 module reference Chapter 4, “E2EP07 module references“ which is provided with the dependent modules E2E.

3.2. Background information

The general concept of end-to-end communication protection is described in the EB tresos E2E Protection Transformer documentation of the E2EXf module, based on the AUTOSAR E2E transformer specified in [2].

3.2.1. Functional overview

3.2.1.1. Safety mechanisms

This profile is based on E2E Profile 7 specified by AUTOSAR, see [1]. It is called from the virtual functional bus generated by the Rte module together with a previously called serializing transformer, e.g. ComXf, or

SomeIpXf) to add protection information to the serialized data stream for the following communication paradigms:

- Non-blocking queued sender-receiver communication

  E2EP07 provides APIs to add protection information at the sender to the result of a serializing transformer, e.g. ComXf or SomeIpXf. It also provides APIs to cyclically check for communication errors by using this information at the receiver. Its API functions are called by the E2EXf module.

The E2EP07 module uses the following safety mechanisms:

- **Cyclic redundancy check (CRC):** A 64-bit CRC is explicitly sent with polynomial in normal form 0x42F0E1EBA9EA3693 with an initial value 0xFFFFFFFFFFFFFFFF and a final XOR-value 0xFFFFFFFFFFFFFFFF.

  NOTE: CRC implementation is part of E2EP07 module itself. There is no separate module for CRC implementation.
- **Sequence counter/alive counter:** A 32-bit sequence number is explicitly sent and incremented at every transmission request.
- **Data ID:** A system-wide unique 32-bit data ID is explicitly sent for every port data element.
- **Length:** A 32-bit number to support dynamic-size data.

The header of AUTOSAR E2E Profile 7 can be placed at a specific location in the protected data, by configuring the offset of the entire E2E header. Figure 3.1, “Header layout of AUTOSAR E2E Profile 7.” shows the header layout with a header offset equal to 0. The individual control data fields are encoded in Big Endian with the most significant byte first.

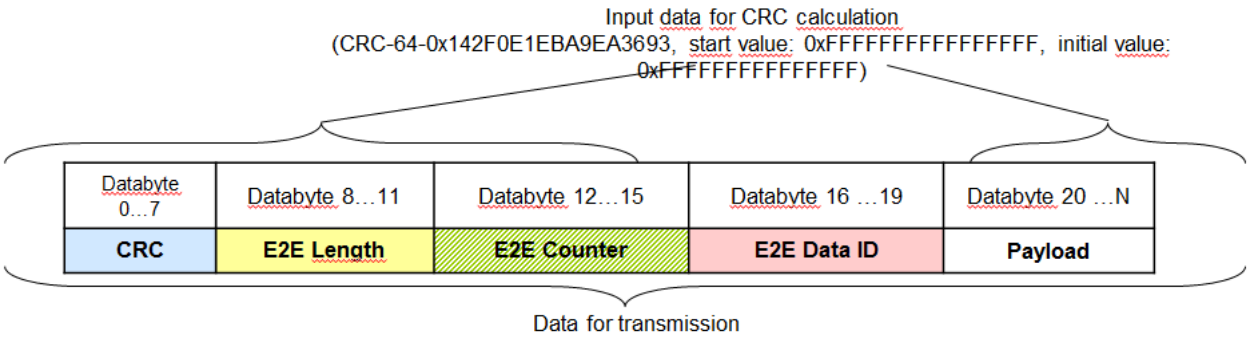

Figure 3.1. Header layout of AUTOSAR E2E Profile 7.

3.2.1.2. Failure modes and required safety mechanisms

The Table 3.1, “Failure modes detection matrix for E2E Profile 7” shows the failure modes and the required safety mechanisms of E2E Profile 7 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.

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

Table 3.1. Failure modes detection matrix for E2E Profile 7

3.3. Configuring E2EP07

To configure the `E2EP07` 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 `E2EP07` module, you must configure additional modules as outlined below:

- The `E2EP07` module requires API functions and data types from the `E2E` library module. This module does not provide any configuration parameters.
- The `E2EP07` module provides API functions and data types required from the `E2ESM` library module and from the `E2EXf` module. For more information on the `E2ESM` and `E2EXf` modules, see [1] and [2]).

3.4. E2EP07 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, “E2EP07 module references“ sub-section `Integration notes` in each module.

4. E2EP07 module references

4.1. Overview

This chapter provides module references for the E2EP07 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 E2EP07 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**
CommonPublishedInformation	1..1	**Label:** Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
PublishedInformation	1..1	**Label:** EB Published Information Additional published parameters not covered by CommonPublishedInformation container.

4.2.1.1. CommonPublishedInformation

Parameters included	
Parameter name	**Multiplicity**
ArMajorVersion	1..1
ArMinorVersion	1..1
ArPatchVersion	1..1
SwMajorVersion	1..1
SwMinorVersion	1..1
SwPatchVersion	1..1
ModuleId	1..1
VendorId	1..1
VendorApiInfix	1..1
Release	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
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**
Label	Software Minor Version
Description	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	0
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

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

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

Parameter Name	**VendorId**
Label	Vendor ID

Description	Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**VendorApiInfix**
Multiplicity	1..1
Type	STRING_LABEL

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

4.2.1.2. PublishedInformation

Parameters included	
Parameter name	**Multiplicity**
PbcfgMSupport	1..1

Parameter Name	**PbcfgMSupport**
Label	PbcfgM support
Description	Specifies whether or not the E2E can use the PbcfgM module for post-build support.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**PublishedInformation:**

Origin	Elektrobit Automotive GmbH

4.2.2. Application programming interface (API)

4.2.2.1. Type definitions

4.2.2.1.1. E2E_PCheckStatusType

Purpose	Status type for E2E State Machine return values.
Type	`uint8`

4.2.2.2. Macro constants

4.2.2.2.1. E2E_AR_MAJOR_VERSION

Purpose	AUTOSAR specification major version.
Value	4U

4.2.2.2.2. E2E_AR_MINOR_VERSION

Purpose	AUTOSAR specification minor version.
Value	2U

4.2.2.2.3. E2E_AR_PATCH_VERSION

Purpose	AUTOSAR specification patch version.
Value	1U

4.2.2.2.4. E2E_AR_RELEASE_MAJOR_VERSION

Purpose	AUTOSAR release major version.
Value	4U

4.2.2.2.5. E2E_AR_RELEASE_MINOR_VERSION

Purpose	AUTOSAR release minor version.
Value	2U

4.2.2.2.6. E2E_AR_RELEASE_REVISION_VERSION

Purpose	AUTOSAR release revision version.
Value	1U

4.2.2.2.7. E2E_E_INPUTERR_NULL

Purpose	At least one pointer parameter is a NULL pointer.
Value	0x13U

4.2.2.2.8. E2E_E_INPUTERR_WRONG

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

4.2.2.2.9. E2E_E_INTERR

Purpose	An internal library error has occurred.
Value	0x19U
Description	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

Purpose	Invalid value passed to function.
Value	0xFFU

4.2.2.2.11. E2E_E_OK

Purpose	Function completed successfully.
Value	0U

4.2.2.2.12. E2E_E_WRONGSTATE

Purpose	Function executed in wrong state.
Value	0x1AU

4.2.2.2.13. E2E_MODULE_ID

Purpose	AUTOSAR module identification.
Value	207U

4.2.2.2.14. E2E_P_ERROR

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

4.2.2.2.15. E2E_P_NONEWDATA

Purpose	No new Data is available since the last call of the check function.
Value	0x05U

Description	Note: related to E2E_PCheckStatusType

4.2.2.2.16. E2E_P_NOTAVAILABLE

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

4.2.2.2.17. E2E_P_OK

Purpose	The checks of the Data in this cycle were successful (including counter check).
Value	0x00U
Description	Note: related to E2E_PCheckStatusType

4.2.2.2.18. E2E_P_REPEATED

Purpose	Either no new data is available, or the new data has a repeated counter.
Value	0x01U
Description	Note: related to E2E_PCheckStatusType

4.2.2.2.19. E2E_P_WRONGSEQUENCE

Purpose	The checks of the Data in this cycle were successful, with the exception of counter jump, which changed more than allowed delta.
Value	0x02U
Description	Note: related to E2E_PCheckStatusType

4.2.2.2.20. E2E_SW_MAJOR_VERSION

Purpose	AUTOSAR module major version.
Value	2U

4.2.2.2.21. E2E_SW_MINOR_VERSION

Purpose	AUTOSAR module minor version.
Value	0U

4.2.2.2.22. E2E_SW_PATCH_VERSION

Purpose	AUTOSAR module patch version.
Value	18U

4.2.2.2.23. E2E_VENDOR_ID

Purpose	AUTOSAR vendor identification: Elektrobit Automotive GmbH.
Value	1U

4.2.2.3. Functions

4.2.2.3.1. E2E_GetVersionInfo

Purpose	Return the modules version information.	
Synopsis	`void` **`E2E_GetVersionInfo`** `( Std_VersionInfoType * VersionInfo );`	
Service ID	0x14	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	`VersionInfo`	Pointer where to store the version information of this module
Description	This service returns the version information of this module. The version information includes: ► Module Id ► Vendor Id ► Vendor specific version numbers	

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. E2EP07

4.3.1. Configuration parameters

Containers included		
Container name	**Multiplicity**	**Description**
CommonPublishedInformation	1..1	**Label:** Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
PublishedInformation	1..1	**Label:** EB Published Information Additional published parameters not covered by CommonPublishedInformation container.

4.3.1.1. CommonPublishedInformation

Parameters included	
Parameter name	**Multiplicity**
ArMajorVersion	1..1
ArMinorVersion	1..1
ArPatchVersion	1..1
SwMajorVersion	1..1
SwMinorVersion	1..1
SwPatchVersion	1..1
ModuleId	1..1
VendorId	1..1
VendorApiInfix	1..1
Release	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	3
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	0
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	1
Configuration class	**PublishedInformation:**

Origin	Elektrobit Automotive GmbH

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

Parameter Name	**SwPatchVersion**
Label	Software Patch Version
Description	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	12
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

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

Parameter Name	**VendorId**
Label	Vendor ID
Description	Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list

Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**VendorApiInfix**
Multiplicity	1..1
Type	STRING_LABEL

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

4.3.1.2. PublishedInformation

Parameters included	
Parameter name	**Multiplicity**
PbcfgMSupport	1..1

Parameter Name	**PbcfgMSupport**
Label	PbcfgM support
Description	Specifies whether or not the E2EP07 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_P07CheckStateType

Purpose	Definition of E2E Profile 7 receiver state type.	
Type	struct	
Members	`E2E_P07CheckStatusType Status`	Result of the verification of the Data, determined by the Check function.
	`uint32 Counter`	Counter of the data in previous cycle.
Description	State of the sender for a Data protected with E2E Profile 7.	

4.3.2.1.2. E2E_P07CheckStatusType

Purpose	Definition of E2E Profile 7 receiver status type.	
Type	enum	
Constants	E2E_P07STATUS_OK	New data has been correctly received.
	E2E_P07STATUS_NONEWDATA	The Check function has been invoked but new Data is not available since the last call.
	E2E_P07STATUS_ERROR	The data has been received according to communication medium, but the CRC is incorrect.
	E2E_P07STATUS_REPEATED	New data has been correctly received, but the Counter is identical to the most recent Data
	E2E_P07STATUS_OKSOMELOST	New data has been correctly received, but some data in the sequence have been probably lost.
	E2E_P07STATUS_WRONGSEQUENCE	The new data has been correctly received, but the Counter Delta is too big (DeltaCounter > MaxDeltaCounter)

Description	Result of the verification of the Data in E2E Profile 7, determined by the Check function.

4.3.2.1.3. E2E_P07ConfigType

Purpose	Non-modifiable configuration of the data element sent over an RTE port, for E2E Profile.	
Type	struct	
Members	`uint32 DataID`	A system-unique identifier of the Data.
	`uint32 Offset`	Bit offset of the E2E header from the beginning of the Data Array. The offset shall be a multiple of 8 and 0<=Offset<=MaxDataLength-(20*8). Example: If Offset equals 8, then the high byte of the E2E Length (32 bit) is written to Byte 1, the low Byte is written to Byte 2. Additionally, Offset + size of header (160 bits) shall be <= Length*8
	`uint32 MinDataLength`	Minimum length of Data in bits. The value shall be >= 20*8 and <= MaxDataLength. NOTE: In general, the MinDataLength shall at least include the E2EP07 header (20 bytes) itself without any additional data. However, it is configurable for specific needs, if it is known that certain data will be transmitted in any case, e.g. the SomeIp header if SomeIpXf is used as a serializer or a length field for certain data types. Example: E2EP07 (20 bytes E2E header) used in conjunction with SomeIpXf (8 bytes header) and a variable-size array is transmitted (even if the current size is 0, at least one length field with a minimum configurable size of 1 byte must be transmitted). In this case, the MinDataLength results in 20 bytes + 8 bytes + 1 byte = 29 bytes.
	`uint32 MaxDataLength`	Maximum length of data in bits. The value shall be >= MinDataLength and <= 4096*8*1024 (4 MB). NOTE: MaxDataL-

		ength is up to 4 MB (system dependent) and includes the E2EP07 header and the payload. Any data to be protected by E2E and which are different from the E2EP07 header, is considered as payload. Example: For E2EP07 the SomeIpXf header and also the length field for variable size arrays are considered as payload.
	`uint32 MaxDeltaCounter`	Maximum allowed gap between two counter values of two consecutively received valid Data.
Description	The position of the counter and CRC is not configurable in Profile 7.	

4.3.2.1.4. E2E_P07ProtectStateType

Purpose	State of the sender for a Data protected with E2E Profile 7.	
Type	struct	
Members	`uint32 Counter`	Counter to be used for protecting the next Data.

4.3.2.2. Macro constants

4.3.2.2.1. E2EP07_AR_RELEASE_MAJOR_VERSION

Purpose	AUTOSAR release major version.
Value	4U

4.3.2.2.2. E2EP07_AR_RELEASE_MINOR_VERSION

Purpose	AUTOSAR release minor version.
Value	3U

4.3.2.2.3. E2EP07_AR_RELEASE_REVISION_VERSION

Purpose	AUTOSAR release revision version.

Value	0U

4.3.2.2.4. E2EP07_SW_MAJOR_VERSION

Purpose	AUTOSAR module major version.
Value	1U

4.3.2.2.5. E2EP07_SW_MINOR_VERSION

Purpose	AUTOSAR module minor version.
Value	0U

4.3.2.2.6. E2EP07_SW_PATCH_VERSION

Purpose	AUTOSAR module patch version.
Value	12U

4.3.2.2.7. E2EP07_VENDOR_ID

Purpose	AUTOSAR vendor identification: Elektrobit Automotive GmbH.
Value	1U

4.3.2.3. Functions

4.3.2.3.1. E2E_P07Check

Purpose	Check the received Data using the E2E Profile 7.
Synopsis	`Std_ReturnType` **`E2E_P07Check`** `( const E2E_P07ConfigType * ConfigPtr , E2E_P07CheckStateType * StatePtr , const uint8 * DataPtr , uint32 Length );`
Service ID	0x23

Sync/Async	Synchronous	
Reentrancy	Reentrant for different communication data / states	
Parameters (in)	`ConfigPtr`	Pointer to static configuration.
	`DataPtr`	Pointer to received Data.
	`Length`	Length of the Data.
Parameters (in,out)	`StatePtr`	Pointer to port/data communication state.
Return Value	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.
Description	Checks the Data received using the E2E Profile 7. This includes CRC calculation, handling of Sequence Counter and Data ID.	

4.3.2.3.2. E2E_P07CheckInit

Purpose	Initializes the check state.	
Synopsis	`Std_ReturnType` **`E2E_P07CheckInit`** `( E2E_P07CheckStateType * StatePtr );`	
Service ID	0x24	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different states	
Parameters (out)	`StatePtr`	Pointer to port/data communication state.
Return Value	Function execution success status	
	`E2E_E_INPUTERR_NULL`	NULL pointer passed.
	`E2E_E_OK`	Function completed successfully.
Description	Initializes the state structure by setting the Counter to 0xFFFFFFFF and the State to E2E_P07STATUS_ERROR.	

4.3.2.3.3. E2E_P07MapStatusToSM

Purpose	Maps the check status of Profile 7 to a generic check status.

Synopsis	`E2E_PCheckStatusType` **`E2E_P07MapStatusToSM`** `( Std_ReturnType CheckReturn , E2E_P07CheckStatusType Status );`	
Service ID	0x25	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different status types	
Parameters (in)	`CheckReturn`	Return value of the E2E_P07Check function.
	`Status`	Status determined by E2E_P07Check function.
Return Value	Profile-independent status of the reception on one single Data in one cycle.	
	`E2E_P_OK`	CheckReturn is E2E_E_OK and Status is E2E_P07STATUS_OK or E2E_P07STATUS_OKSOMELOST.
	`E2E_P_ERROR`	CheckReturn is E2E_E_OK and Status is E2E_P07STATUS_ERROR or CheckReturn is different than E2E_E_OK or Status is undefined.
	`E2E_P_REPEATED`	CheckReturn is E2E_E_OK and Status is E2E_P07STATUS_REPEATED
	`E2E_P_NONEWDATA`	CheckReturn is E2E_E_OK and Status is E2E_P07STATUS_NONEWDATA.
	`E2E_P_WRONGSEQUENCE`	CheckReturn is E2E_E_OK and Status is E2E_P07STATUS_WRONGSEQUENCE.
Description	The function maps the check status of Profile 7 to a generic check status, which can be used by E2E state machine check function. The E2E Profile 7 delivers a more fine-granular status, but this is not relevant for the E2E state machine.	

4.3.2.3.4. E2E_P07Protect

Purpose	Protects the array/buffer to be transmitted using the E2E Profile 7.
Synopsis	`Std_ReturnType` **`E2E_P07Protect`** `( const E2E_P07ConfigType * ConfigPtr , E2E_P07ProtectStateType * StatePtr , uint8 * DataPtr , uint32 Length );`
Service ID	0x21
Sync/Async	Synchronous
Reentrancy	Reentrant for different communication data / states

Parameters (in)	`ConfigPtr`	Pointer to static configuration.
	`Length`	Length of the data in bytes.
Parameters (in,out)	`StatePtr`	Pointer to port/data communication state.
	`DataPtr`	Pointer to Data to be transmitted.
Return Value	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.
Description	Protects the array/buffer to be transmitted using the E2E Profile 7. This includes checksum calculation, handling of Sequence counter and Data ID.	

4.3.2.3.5. E2E_P07ProtectInit

Purpose	Initializes the protection state.	
Synopsis	`Std_ReturnType` **`E2E_P07ProtectInit`** `( E2E_P07ProtectStateType * StatePtr );`	
Service ID	0x22	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different states	
Parameters (out)	`StatePtr`	Pointer to port/data communication state.
Return Value	Function execution success status	
	`E2E_E_INPUTERR_NULL`	NULL pointer passed.
	`E2E_E_OK`	Function completed successfully.
Description	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 `E2EP07` module.

4.3.3.2. Production errors

Production errors are not reported by the `E2EP07` 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 E2EP07 module.

5. Bibliography

Bibliography

[1] *AUTOSAR Specification of SW-C End-to-End Communication Protection Library*, Issue AUTOSAR 4.3.0, Publisher: AUTOSAR

[2] *AUTOSAR Specification of Module E2E Transformer*, Issue AUTOSAR 4.3.0, Publisher: AUTOSAR