VECTOR

MICROSAR E2E

Technical Reference

Version 7.00.00

Authors	virgmi, virsrl, virljs, vireno, virbka, virbmz, virhdr, twagenpfeil
Status	Released

Document Information

History

Author	Date	Version	Remarks
virgmi	2015-07-03	1.00.00	Initial version
virgmi	2015-10-21	1.01.00	Support of JLR E2E profile
virgmi	2016-11-25	1.02.00	Support of E2E profile 7
virgmi	2018-01-15	1.03.00	Update to ASR 4.2.2
virsrl	2019-08-19	1.04.00	MWDG-99: Support of Safety Related Signal Implementation 6.00.00
virsrl	2019-10-22	1.05.00	MWDG-1925: E2ELib shall support AUTOSAR 4.3.0 profile 11
virljs	2020-10-27	2.00.00	MWDG-4079: E2ELib: Add Profile 44
virljs	2021-07-22	3.00.00	MWDG-5455: E2ELib: Add Profile 8
virljs	2021-09-23	4.00.00	MWDG-5625: Document sequences of API invocations
vireno	2021-10-14	5.00.00	MWDG-2527: Support Profile 22
virbka, virbmz	2022-01-24	6.00.00	MWDG-5808: Generate E2E_MemMap.h MWDG-6482: Update E2ELib to new window size attributes. MWDG-6638: Adapt E2ELib to the fixed E2EProtocol R19-11 Specification MWDG-5650: E2E shall implement the SM according to ASR20-11
virhdr	2023-10-16	6.01.00	MEMSLP-8728: Support E2E Profile 11 according to ASR 21-11
twagenpfeil	2024-02-12	7.00.00	MEMSLP-8116: Extend E2E lib with profile for J1939-76

Reference Documents

No.	Source	Title	Version
[1]	AUTOSAR	Specification of SW-C End-to-End Communication Protection Library	V4.2.2
[2]	AUTOSAR	Specification of SW-C End-to-End Communication Protection Library	V4.3.0
[3]	AUTOSAR	Specification of SW-C End-to-End Communication Protection Library	V4.3.1
[4]	AUTOSAR	Specification of SW-C End-to-End Communication Protection Library	R19-11
[5]	AUTOSAR	Specification of SW-C End-to-End Communication Protection Library	R20-11
[6]	AUTOSAR	Specification of SW-C End-to-End Communication Protection Library	R21-11
[7]	AUTOSAR	List of Basic Software Modules	V4.2.1
[8]	SAE J1939 - 76	Functional Safety Communications Protocol	APR2020
[9]	Vector Informatik	MICROSAR MemMap Technical Reference	See delivery

Scope of the Document

This technical reference describes the general use of the E2E library basis software. The E2E Library was developed according to ISO 26262 for use in safety-related items. There are no aspects which are controller specific.

Caution
We have configured the programs in accordance with your specifications in the questionnaire. Whereas the programs do support other configurations than the one specified in your questionnaire, Vector´s release of the programs delivered to your company is expressly restricted to the configuration you have specified in the questionnaire.

Contents

1 Introduction 7
1.1 Architecture Overview 8

2 Functional Description 10
2.1 Features 10
2.2 E2E Communication Protection 10
2.2.1 Communication Faults 10
2.2.2 Fault Model 11
2.2.3 Protection mechanisms 11
2.2.4 Detected Communication Faults 12
2.3 Initialization 12
2.4 States 12
2.5 Main Functions 13
2.6 Error Handling 13
2.7 Intended usage 13

3 Integration 17
3.1 Scope of Delivery 17
3.1.1 Static Files 17
3.1.2 Dynamic Files 17
3.1.3 MemMap 17
3.2 Include Structure 19
3.3 Make 19
3.4 Critical Sections 19
3.5 Invocation of Protect and Check service 19
3.6 Compatibility to AUTOSAR R21-11 and newer 20
3.7 Additional integration information for profile J1939-76 20

4 API Description 22
4.1 Type Definitions 22
4.2 Special Type Definitions for profile J1939-76 23
4.3 Services provided by E2E 24
4.3.1 E2E_PXXProtect 24
4.3.2 E2E_PXXProtectInit 26
4.3.3 E2E_PXXCheck 27
4.3.4 E2E_PXXCheckInit 28
4.3.5 E2E_PXXMapStatusToSM 29
4.3.6 E2E_SMCheck 30

4.3.7 E2E_SMCheckInit 31
4.4 Services used by E2E 32
4.5 Callback Functions 32
4.6 Configurable Interfaces 32

5 Configuration 33

6 Glossary and Abbreviations 34
6.1 Glossary 34
6.2 Abbreviations 34

7 Contact 35

Illustrations

Figure 1-1 AUTOSAR 4.2 Architecture Overview 8
Figure 1-2 Interfaces to adjacent modules of E2E component 9
Figure 2-1 E2E State Machine 13
Figure 2-2 Usage of E2E_PXXCheck 15
Figure 2-3 Usage of E2E_PXXProtect 16
Figure 3-1 Include structure containing all possible E2E profiles 19
Figure 3-2 Structure of the buffer handled by the E2E Library 20
Figure 4-1 Datatypes for profile J1939-76 24

Tables

Table 2-1 Supported AUTOSAR standard conform features 10
Table 2-2 Features provided beyond the AUTOSAR standard 10
Table 2-3 Provided E2E protection mechanisms 12
Table 2-4 Detected communication faults 12
Table 3-1 Static files 17
Table 3-2 Example for setting E2E Profile 1 in E2E_cfg.mak 19
Table 4-1 Type definitions 22
Table 4-2 Type definitions for profile J1939-76 23
Table 4-3 E2E_PXXCheck 27
Table 4-4 E2E_PXXCheckInit 28
Table 4-5 E2E_PXXMapStatusToSM 29
Table 4-6 E2E_SMCheck 30
Table 4-7 E2E_SMCheckInit 31
Table 4-8 Services used by the E2E 32
Table 6-1 Glossary 34
Table 6-2 Abbreviations 34

1 Introduction

This document describes the functionality and API of the AUTOSAR BSW module E2E.

Latest Analyzed AUTOSAR Release:	R21-11	
AUTOSAR Schema Compatibility:	R4.2.2	
Supported Configuration Variants:	pre-compile	
Vendor ID:	E2E_VENDOR_ID	30 decimal (= Vector-Informatik, according to HIS)
Module ID:	E2E_MODULE_ID	207 decimal (according to ref. [7])

* For the detailed functional specification please also refer to the corresponding AUTOSAR SWS.

E2E Library provides mechanisms to protect safety-related data exchange at runtime against the effects of faults within the communication link.

To provide appropriate solution addressing flexibility and standardization, AUTOSAR specifies a set of flexible E2E profiles that implement a combination of E2E protection mechanisms, i.e. Profile 1, 2, 4, 5, 6, 7, 8, 11, 22 and 44. Additionally, an E2E profile for JLR is supported, which is based on AUTOSAR Profile 1. Furthermore, the profile J1939-76 for the SAE standard is supported. Each specified E2E profile has a fixed behavior, but it has some configuration options by function parameters (e.g. the location of the CRC in relation to the data, which are to be protected). This document illustrates the functional principle of E2E without getting too deep into details about any particular E2E profile. For information about the used E2E profile (e.g. data layout, CRC computation), refer to [2] for profile 7, [6] for profile 11, [3] for profile 22, [8] for profile J1939-76, [5] for profile 8 and 44 and [1] for all other profiles.

The E2E protection allows the following:

- It protects the safety-related data elements to be sent over the RTE by attaching control data (E2E Header)
- It verifies the safety-related data elements received from the RTE using the control data (E2E Header)
- It indicates that received safety-related data elements are faulty, so that the caller of E2E library can respond in a proper way

E2E Library provides a state machine to signalize the state of communication link over several message cycles.

The E2E Profile check-functions verify data in one cycle. In contrary, the state machine builds up a state out of several results of check-functions within a reception window, which is then provided to the consumer (RTE/SWC/COM).

E2E State Machine should be interpreted more like an aggregator of several check-results than as a real state machine. For information about the E2E state machine refer to [5].

1.1 Architecture Overview

The following figure shows where the E2E is located in the AUTOSAR architecture.

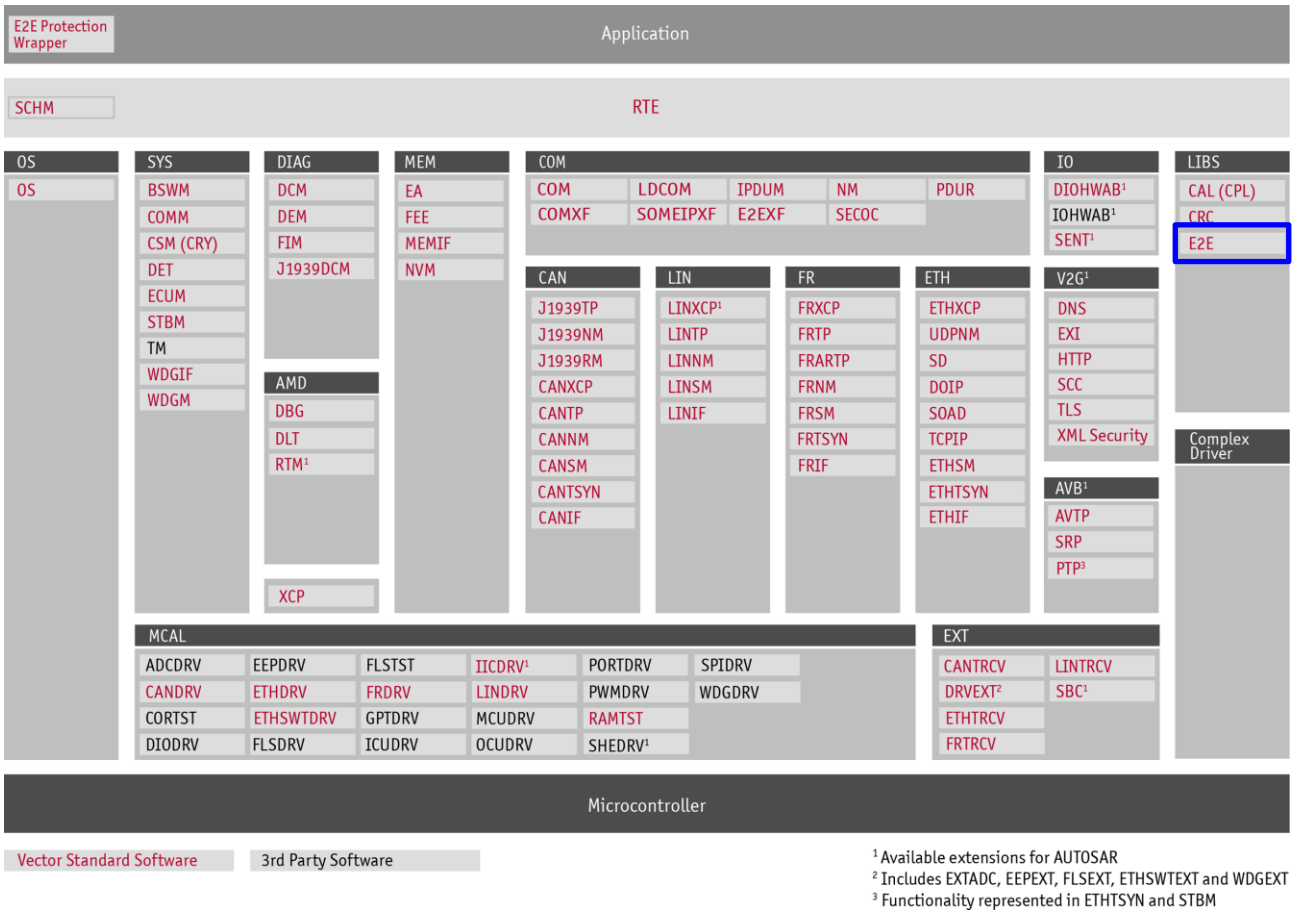

Figure 1-1 AUTOSAR 4.2 Architecture Overview

The next figure shows the interfaces to adjacent modules of the E2E. These interfaces are described in chapter 4.

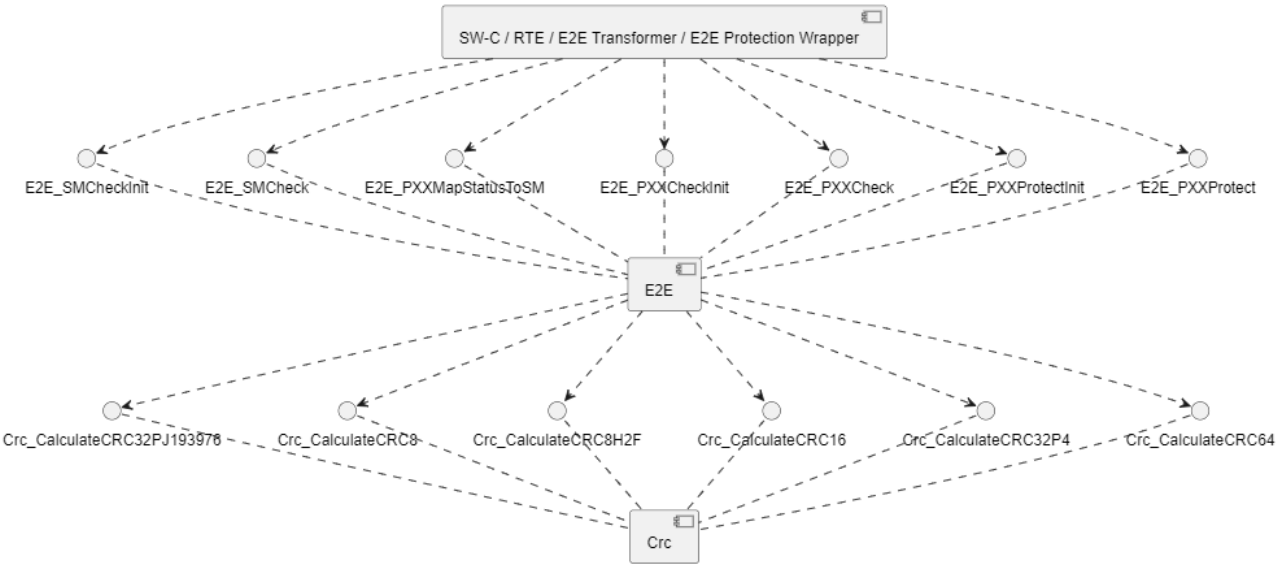

Figure 1-2 Interfaces to adjacent modules of E2E component

PXX is used as placeholder for services, which are implemented in each Profile. For example E2E_**PXX**Protect() develops to the following E2E_**P01**Protect(), E2E_**P02**Protect() and so on.

2 Functional Description

2.1 Features

The features listed in the following tables cover the complete functionality specified for the E2E.

The AUTOSAR standard functionality is specified in [1], the corresponding features are listed in the table

> Table 2-1 Supported AUTOSAR standard conform features

Vector Informatik provides further E2E functionality beyond the AUTOSAR standard. The corresponding features are listed in the table

> Table 2-2 Features provided beyond the AUTOSAR standard

The following features are supported:

Supported AUTOSAR Standard Conform Features
E2E Protection mechanisms via Profile 1, 2, 4, 5, 6, 7, 8, 11, 22 and 44
State machine for communication status within reception window (as specified in [4] and [5])

Table 2-1 Supported AUTOSAR standard conform features

The following features are provided beyond the AUTOSAR standard:

Features Provided Beyond the AUTOSAR Standard
E2E Protection mechanism via Profile JLR
E2E Protection mechanism via Profile J1939-76

Table 2-2 Features provided beyond the AUTOSAR standard

2.2 E2E Communication Protection

This chapter gives an overview over E2E communication protection and provides a short summary of the AUTOSAR architecture for E2E communication protection. Please note that this chapter is not intended to provide a full description (or even a specification) of the AUTOSAR architecture for E2E communication protection.

2.2.1 Communication Faults

To ensure freedom from interference by software partitioning, ISO 26262 takes the following faults into account:

> Loss of peer-to-peer communication

> Unintended message repetition due to the same message being sent again unintentionally

> Message loss during transmission

> Insertion of messages due to receiver unintentionally receiving an additional message, which is interpreted to have correct source and destination addresses

- Re-sequencing due to the order of data being changed during transmission, i.e., the data is not received in the same order as it was sent
- Message corruption due to one or more data bits in the message begin changed during transmission
- Message delay due to the message being received correctly, but not in time
- Blocking access to the data bus due to a faulty node not adhering to the expected patterns of use and making excessive demands for service, thereby reducing its availability to other nodes, e.g. while wrongly waiting for non-existent data
- Constant transmission of messages by a faulty node, thereby compromising the operation of the entire bus

These communication faults may arise due to (systematic) software faults, (random) hardware faults and transient faults caused by external influences.

2.2.2 Fault Model

The following communication faults can be addressed by the E2E communication protection:

- Repetition: The same message is received more than once
- Deletion: The message or parts of it have been removed from the communication stream
- Insertion: An additional message or parts of it have been inserted into the communication stream
- Incorrect sequence: Messages of a communication stream are received in an incorrect order
- Corruption: A message or parts of it are modified
- Timing faults (delay): The timing constraints of a message are violated
- Addressing faults: A message is sent to the wrong destination
- Inconsistency: Communicating nodes have different views of the network status of data being transferred
- Masquerading: The design of a received message with non-authentic content appears authentic, as if sent by the correct sender

The specific features of the E2E communication protection depend on the implementation of the protection mechanisms.

2.2.3 Protection mechanisms

The following mechanisms are provided by E2E profiles. Note that combination and implementation of protection mechanisms differ in each E2E Profile. Table 2-3 summarizes the possible mechanisms. For details about the protection mechanisms in any specific E2E Profile, take a look at the corresponding specification.

E2E Mechanism	Description
Counter	Representing sequence number, which is incremented on every send request. Counter is either transmitted implicitly by including it in the CRC calculation or it is transmitted explicitly as well as covered by CRC calculation.
Timeout monitoring	Timeout is determined by E2E Library by means of evaluation of the Counter, by a non-blocking read at the receiver. Timeout is reported by E2E Library to the caller by means of the status flags in E2E_PXXCheckStatusType.
Data ID	Unique number which is transmitted either implicitly (only included in CRC calculation) or explicitly and covered by CRC calculation.
Length	Length of data being transmitted is sent explicitly by some E2E profiles.
CRC	The CRC routines are provided by CRC Library. Entire E2E Header and data to be transmitted is included in CRC calculation.

Table 2-3 Provided E2E protection mechanisms

2.2.4 Detected Communication Faults

The E2E mechanisms can detect the following faults or effects of faults:

E2E Mechanism	Detected communication faults
Counter	Repetition, Loss, Insertion, Incorrect sequence, Blocking
Transmission on a regular basis and timeout monitoring using E2E Library	Loss, Delay, Blocking
Data ID + CRC	Masquerade and incorrect addressing, Insertion
CRC	Corruption, Asymmetric information

Table 2-4 Detected communication faults

2.3 Initialization

The caller of E2E Library has to provide configuration parameters according to type definitions, which are defined in section 4.1. E2E Protection and check services do require state parameters which are to be initialized before first protect/check cycle. The state machine also requests a state parameter which requires initialization. Therefore, E2E provides several initialization services, which are responsible for resetting state parameters.

> E2E_PXXProtectInit has to be called before first protection

> E2E_PXXCheckInit has to be called before first check

> E2E_SMCheckInit has to be called to initialize the state machine

2.4 States

E2E state machine indicates the communication status over several protect/check cycles.

According to state machine configuration which is passed to E2E_SMCheck service via input parameter, communication link is trustworthy as long as state machine is in state E2E_SM_VALID. Received data can then be used.

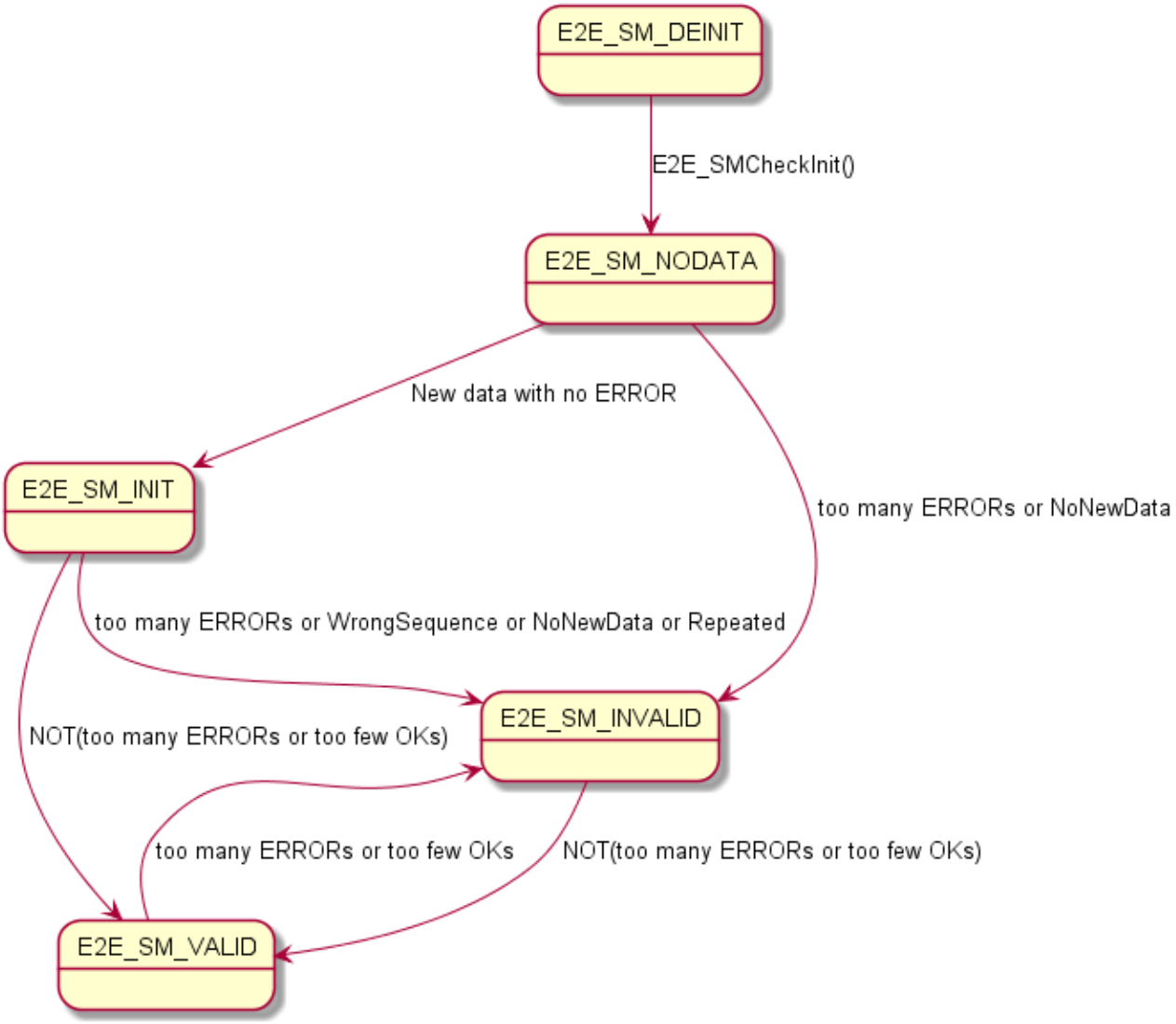

Figure 2-1 E2E State Machine

E2E Profile services are stateless.

2.5 Main Functions

E2E does not contain any scheduled services such as a main function.

2.6 Error Handling

E2E does not support error reporting via DET or DEM modules. Errors are reported synchronously upon call of services via return values and status within state parameters.

2.7 Intended usage

Below are diagrams to show the intended usage of the E2E Library.
For information on how to handle return values of the functions or the resulting states of the E2E_PXXCheck function calls, please refer to the respective specification.

Note
The configurations need to be initialized by the user, only the state will be initialized by the respective initialization functions.

Note
The results of the E2E_PXXCheck function will be within the State that is given to the function. This result needs to be handled by the user, refer to the respective specification for more information about the status.

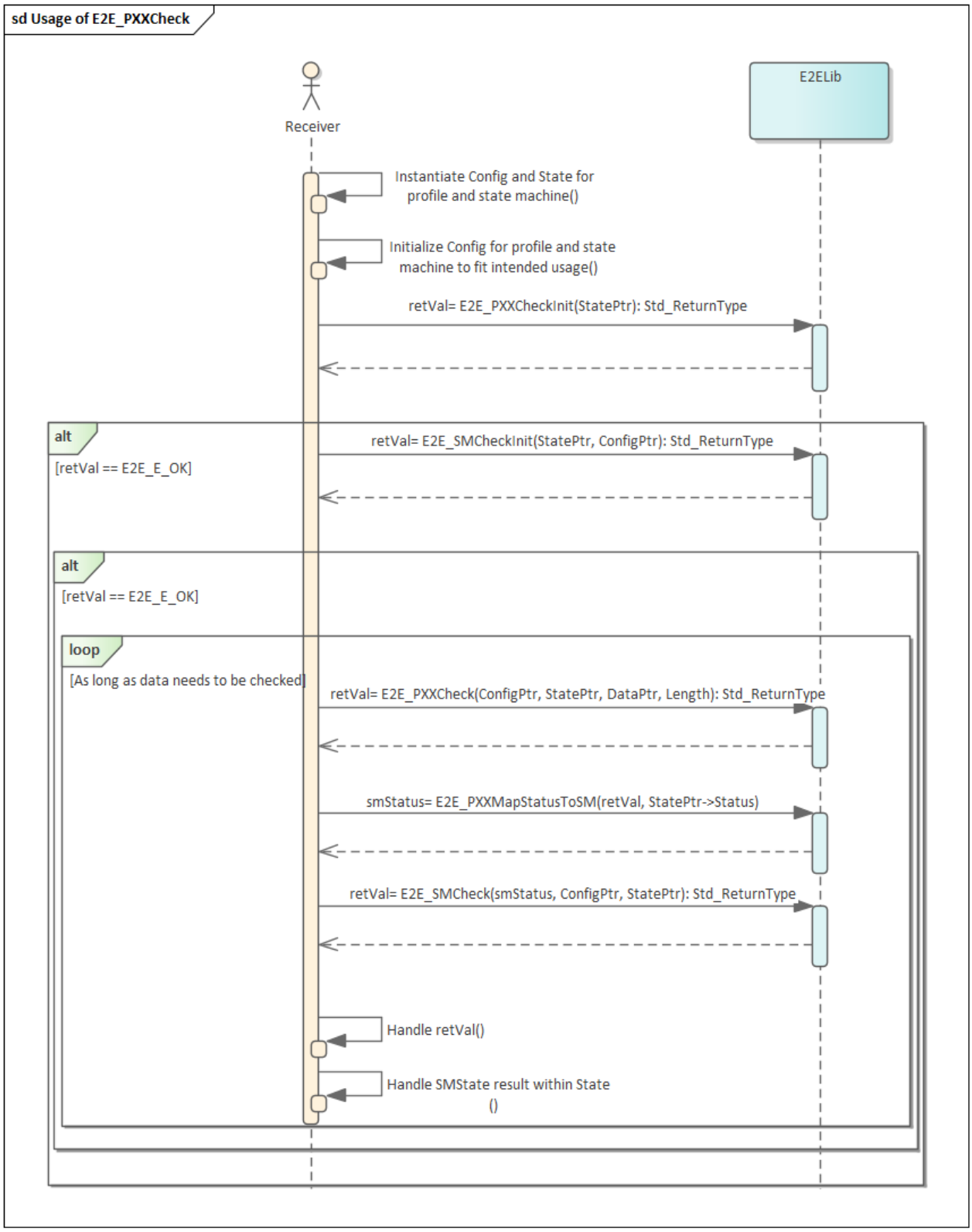

Figure 2-2 Usage of E2E_PXXCheck

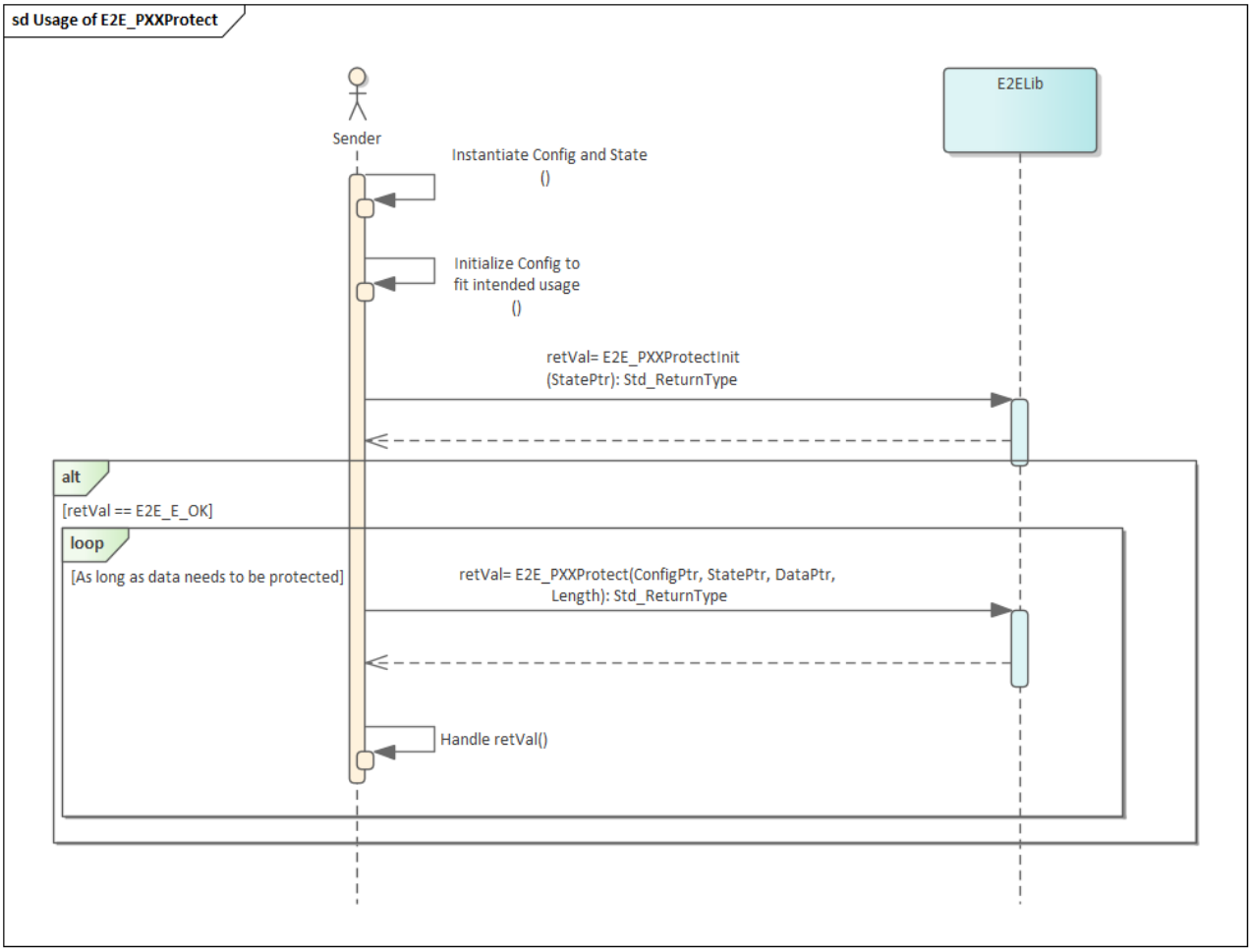

Figure 2-3 Usage of E2E_PXXProtect

3 Integration

This chapter gives necessary information for the integration of the MICROSAR E2E into an application environment of an ECU.

3.1 Scope of Delivery

The delivery of the E2E contains the files which are described in the chapters 3.1.1 and 3.1.2:

3.1.1 Static Files

File Name	Description
E2E.c	This is the common source file of E2E Library.
E2E.h	This is the common header file of E2E Library.
E2E_PXX.c	This is the specific source file for E2E Profile PXX.
E2E_PXX.h	This is the specific header file for E2E Profile PXX.
E2E_SM.c	This is the source file for E2E State Machine.
E2E_SM.h	This is the header file for E2E State Machine.

Table 3-1 Static files

PXX is a placeholder for all supported profiles. Upon delivery of E2E Library only one particular profile is supported.

It is important to note that only the header file E2E_PXX.h has to be included by the caller of E2E Library. See chapter 3.2 for include structure of E2E Library.

Note

For profile J1939-76 the infix of the different files is equal to E2E_PJ193976.

For profile J1939-76 the files Crc.c and Crc.h contain the used function for the CRC calculation.

Actually, the files Crc.c and Crc.h contain the CRC calculation methods of all profiles and belong to the delivery.

3.1.2 Dynamic Files

E2E Library does not contain any generated files.

3.1.3 MemMap

Delivery with MICROSAR Release < R27:

E2E_MemMap.h is available as a static file.

Memory mapping sections of the E2E module are included within the global MemMap.h file (via E2E_MemMap.inc), which will be included by E2E_MemMap.h.

Delivery with MICROSAR Release >= R27:

E2E_MemMap.h is available as a dynamic file. This header file is generated by an own component “MemMap“ (for more information refer to [9]).

Memory mapping sections of the E2E module are directly included within generated E2E_MemMap.h to reduce the build time.

3.2 Include Structure

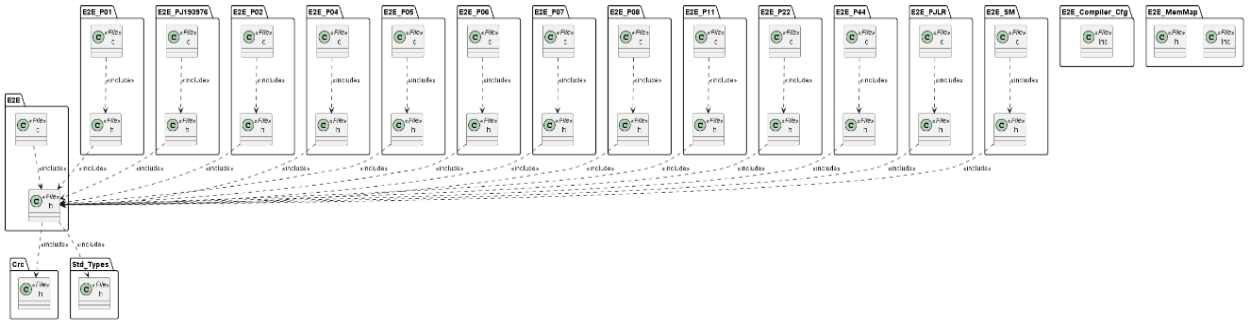

Figure 3-1 Include structure containing all possible E2E profiles

3.3 Make

The content of make sub-package has to be adapted according to the E2E Profile which is used in a delivery. Therefore, set parameter `E2E_USE_PROFILE_XX` to `1` in E2E_cfg.mak file. For example, if E2E Profile 1 is used:

Setting Profile 1 in E2E_cfg.mak

```
#####################################################################
# Use PROFILE 1
#
# Possible Values:
# 0 - No PROFILE1 used
# 1 - PROFILE1 used
#####################################################################
E2E_USE_PROFILE_01 = 1
```

Table 3-2 Example for setting E2E Profile 1 in E2E_cfg.mak

3.4 Critical Sections

E2E Library does not contain critical sections.

3.5 Invocation of Protect and Check service

- The service to protect `E2E_PXXProtect` (e.g. `E2E_P01Protect`) shall be called exactly once before sending the data.
- The service to check `E2E_PXXCheck` (e.g. `E2E_P01Check`) shall be called exactly once after reception of data.

3.6 Compatibility to AUTOSAR R21-11 and newer

Caution
AUTOSAR changed behavior of E2E Profile 11 with release R21-11 [6].

During CRC calculation, the final value shall be XOR'ed with 0xFF (i.e. the value is inverted).

E2E Library remains compatible with older AUTOSAR releases.

To be compatible with R21-11 or newer versions supply the following Define to the C Preprocessor through your build environment:

`E2E_ASR21_11_COMPATIBILITY`

If this Define is set, it will enable inversion of the final CRC value in E2E Profile 11. If the Define is omitted, it will not by default. Thus, you do not have to change anything if you want to remain compatible with older AUTOSAR releases.

3.7 Additional integration information for profile J1939-76

The user of the profile J1939-76 must adhere to the following layout of the transferred data buffer, as the profile assumes exactly this structure.

Byte in buffer	Byte 0								Byte 1								Byte 2								Byte 3								Byte 4							
Byte bit position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Content	U	U	U	S4	S3	S2	S1	S0	C7	C6	C5	C4	C3	C2	C1	C0	C15	C14	C13	C12	C11	C10	C9	C8	C23	C22	C21	C20	C19	C18	C17	C16	C31	C30	C29	C28	C27	C26	C25	C24

Byte in buffer	Byte 5								Byte 6								Byte 7								Byte 8								Byte 9							
Byte bit position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Content	Payload								Payload								Payload								Payload								Payload							

Byte in buffer	Byte 10								Byte 11								Byte 12							
Byte bit position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Content	Payload								Payload								Payload							

U	Unused bits
S0 - S4	Sequence number
C0 - C31	CRC 32 checksum

Figure 3-2 Structure of the buffer handled by the E2E Library

- First byte must be reserved for the sequence number.
- Only 5 bits of this reserved byte are used.
- CRC of 4 bytes is appended directly after the sequence number.
- CRC is only calculated over the payload (8 bytes as specified in the SAE specification)
- The payload is located directly after the CRC.
- Timing constraints and the CAN ID must be treated by the upper layer.
- The provided profile handles exactly one buffer.

> The E2E_P193976Check API declares the first received data as invalid (i.e. E2E_PJ193976CheckStatusType is equal to E2E_PJ193976STATUS_INITIAL), which means that an error status is reported (i.e. E2E_PJ193976CheckStatusType is set to E2E_PJ193976STATUS_ERROR).

Caution
In the SAE specification the length of the payload is specified to be 8 bytes. Nevertheless, the E2E Library deviates from this by allowing the payload to be at most 8 bytes. The CRC Library uses this size for the CRC calculation.

Observe: No padding / filling is performed by the E2E Library.

4 API Description

For an interface overview please see Figure 1-2.

4.1 Type Definitions

The types defined by the E2E are described in this chapter. Content of structure types varies from profile to profile. Hence, profile types are described in general.

Type Name	C-Type	Description
E2E_PXXConfigType	Structure	Configuration of transmitted data. In detail, configuration of E2E Header including protection mechanisms, which are transmitted explicitly. For each transmitted data, there is an instance of this structure.
E2E_PXXProtectStateType	Structure	State of the sender for a data protected with E2E Profile X. In all E2E profiles the Counter is contained in ProtectState structure.
E2E_PXXCheckStateType	Structure	State of the receiver for single data protected with E2E Profile X.
E2E_PXXCheckStatusType	Enumeration	Status of the receiver for single data in one cycle, protected with E2E Profile X. E2E_PXXCheckStatusType is part of E2E_PXXCheckStateType.
E2E_PCheckStatusType	Enumeration	Profile-independent status of the receiver for single data in one cycle. Result and status of most recent check is mapped (by E2E_PXXMapStatusToSM service) to this type, which is compatible to E2E state machine.
E2E_SMConfigType	Structure	Configuration of a communication channel for exchanging data. Number of considered messages (window size) can be configured, which is used to indicate state of communication link over several transmission cycles.
E2E_SMCheckStateType	Structure	State of the protection of a communication channel. State machine state is contained in this type.
E2E_SMStateType	Enumeration	Status of the communication channel exchanging the data. If the status is E2E_SM_VALID, then the data may be used.

Table 4-1 Type definitions

Changes

`E2E_SMStateType` defined according to [4] means that its former member “`WindowSize`” was replaced by “`WindowSizeInit`”, “`WindowSizeInvalid`” and “`WindowSizeValid`”. Additionally, “`ClearToInvalid`” was added.

For more information see [4].

Setting the three new members “`WindowSizeInit`”, “`WindowSizeInvalid`” and "`WindowSizeValid`” to the former “`WindowSize`” value AND setting “`ClearToInvalid`” to FALSE gets the closest to the former internal behavior of the state machine. In this case only the transition from state E2E_SM_INVALID to state E2E_SM_VALID is now stricter as every ERROR message will reset the OK message count.

Current E2E state machine is defined according to [5]. This means that in addition to the changes that were added because of the adjustments to [4], an extension has been added to state machine, which can be disabled with a switch (`TransitToInvalidExtended == FALSE`). This change according to [5] affects two transitions. The transition from E2E_SM_INIT to E2E_SM_INVALID now occurs not only when too many ERRORs are present, but also in case of: “`WrongSequence`” or “`NoNewData`” or “`Repeated`”. Also, there is now a transition from E2E_SM_NODATA to E2E_SM_INVALID in case there are too many errors or “`NoNewData`” has been detected.

4.2 Special Type Definitions for profile J1939-76

Type Name	C-Type	Content
E2E_PJ193976ConfigType	Structure	DataLength: `uint8` MaxDeltaCounter: `uint8`
E2E_PJ193976ProtectStateType	Structure	SequenceNumber: `uint8`
E2E_PJ193976CheckStateType	Structure	Status: `E2E_PJ193976CheckStatusType` LastValidCounter: `uint8`
E2E_PJ193976CheckStatusType	Enumeration	E2E_PJ193976STATUS_OK = 0x00 E2E_PJ193976STATUS_WRONGCRC = 0x07 E2E_PJ193976STATUS_REPEATED = 0x08 E2E_PJ193976STATUS_OKSOMELOST = 0x20 E2E_PJ193976STATUS_WRONGSEQUENCE = 0x40 E2E_PJ193976STATUS_INITIAL = 0xA1 E2E_PJ193976STATUS_ERROR = 0xEE

Table 4-2 Type definitions for profile J1939-76

Figure 4-1 Datatypes for profile J1939-76

4.3 Services provided by E2E

The services provided by E2E library are described in this chapter. Signature of services may vary from profile to profile. Thus, profile specific parameters are marked appropriately.

Observe that for profile J1939-76 the infix of all provided functions is equal to **E2E_PJ193976**.

4.3.1 E2E_PXXProtect

Prototype
`Std_ReturnType` **`E2E_PXXProtect`** `(E2E_PXXConfigType* ConfigPtr, E2E_PXXProtectStateType* StatePtr, uint8* DataPtr, uint16 Length)`
Parameter
`ConfigPtr`	Pointer to profile configuration
`StatePtr`	Pointer to communication state of sender
`DataPtr`	Pointer to data to be protected
`Length`	Length of data in bytes **(only supported in Profile 4, 5, 6, 7, 8, 11, 22 and 44)**. Type of Length parameter in **Profile 7 and 8** is uint32.

Return code	
`Std_ReturnType`	> E2E_E_INPUTERR_NULL: At least one pointer parameter is NULL > E2E_E_INPUTERR_WRONG: One parameter is erroneous > E2E_E_INTERR: Internal library error > E2E_E_OK: Protection successful
Functional Description	
Protects the array/buffer to be transmitted using the E2E Profile X. This includes checksum calculation, handling of counter and Data ID (**Data ID is not supported in profile J1939-76**). Depending on the used profile, also data length can be included in E2E Header.	
Particularities and Limitations	
> Identical ConfigPtr should be used for both protection on sender side and check on receiver side.	
Expected Caller Context	
> There is no specific caller. E2E can be called from anywhere.	

Table 4-3 E2E_PXXProtect

4.3.2 E2E_PXXProtectInit

Prototype	
`Std_ReturnType` **`E2E_PXXProtectInit`** `(E2E_PXXProtectStateType* StatePtr)`	
Parameter	
`StatePtr`	Pointer to communication state of sender
Return code	
`Std_ReturnType`	> E2E_E_INPUTERR_NULL: Pointer parameter is NULL > E2E_E_OK: Initialization successful
Functional Description	
Initializes the protection state by resetting the counter.	
Particularities and Limitations	
> None	
Expected Caller Context	
> There is no specific caller. E2E can be called from anywhere.	

Table 4-4 E2E_PXXProtectInit

4.3.3 E2E_PXXCheck

Prototype	
`Std_ReturnType` **`E2E_PXXCheck`** `(E2E_PXXConfigType* ConfigPtr, E2E_PXXCheckStateType* StatePtr, uint8* DataPtr, uint16 Length)`	
Parameter	
`ConfigPtr`	Pointer to profile configuration
`StatePtr`	Pointer to communication state of receiver
`DataPtr`	Pointer to data to be checked
`Length`	Length of data in bytes **(only supported in Profile 4, 5, 6, 7, 8, 11, 22 and 44)**. Type of Length parameter in **Profile 7 and 8** is uint32.
Return code	
`Std_ReturnType`	> E2E_E_INPUTERR_NULL: At least one pointer parameter is NULL > E2E_E_INPUTERR_WRONG: One parameter is erroneous > E2E_E_INTERR: Internal library error > E2E_E_OK: Check successful
Functional Description	
Checks the received data using the E2E Profile X. This includes CRC calculation, handling of the counter and Data ID (**Data ID is not supported in profile J1939-76**). Depending on the used profile, also data length can be used for checking received data.	
Particularities and Limitations	
> Identical ConfigPtr should be used for both protection on sender side and checking on receiver side.	
Expected Caller Context	
> There is no specific caller. E2E can be called from anywhere.	

Table 4-3 E2E_PXXCheck

4.3.4 E2E_PXXCheckInit

Prototype	
`Std_ReturnType` **`E2E_PXXCheckInit`** `(E2E_PXXCheckStateType* StatePtr)`	
Parameter	
`StatePtr`	Pointer to communication state of sender
Return code	
`Std_ReturnType`	> E2E_E_INPUTERR_NULL: Input pointer parameter is NULL > E2E_E_OK: Initialization successful
Functional Description	
Initializes the check state by resetting the parameters contained in StatePtr.	
Particularities and Limitations	
> None	
Expected Caller Context	
> There is no specific caller. E2E can be called from anywhere.	

Table 4-4 E2E_PXXCheckInit

4.3.5 E2E_PXXMapStatusToSM

Prototype	
`E2E_PCheckStatusType` **`E2E_PXXMapStatusToSM`** `(Std_ReturnType CheckReturn, E2E_PXXCheckStatusType Status, Boolean profileBehavior)`	
Parameter	
`CheckReturn`	Return value of E2E_PXXCheck service
`Status`	Status of received and checked data in single cycle.
`profileBehavior`	> TRUE: profile behavior as introduced in ASR4.2 > FALSE: profile behavior as before ASR4.2 **This parameter is only supported in Profile 1, 2 and JLR.**
Return code	
`E2E_PCheckStatusType`	Standard state value to be used in E2E Library state machine.
Functional Description	
Maps the check status of Profile X to a generic check status, which can be used by E2E state machine check function. The E2E Profile X delivers a more fine-grained status than it is necessary for E2E state machine.	
Particularities and Limitations	
> None	
Expected Caller Context	
> There is no specific caller. E2E can be called from anywhere.	

Table 4-5 E2E_PXXMapStatusToSM

4.3.6 E2E_SMCheck

Prototype	
`Std_ReturnType` **`E2E_SMCheck`** `(E2E_PCheckStatusType ProfileStatus, E2E_SMConfigType* ConfigPtr, E2E_SMCheckStateType* StatePtr)`	
Parameter	
`ProfileStatus`	Profile independent status of the reception on one single data in one cycle. This is the mapped return value of E2E_PXXMapStatusToSM service.
`ConfigPtr`	Pointer to static configuration of E2E State Machine
`StatePtr`	Pointer to state of the communication channel
Return code	
`Std_ReturnType`	> E2E_E_INPUTERR_NULL: At least one pointer parameter is NULL > E2E_E_INPUTERR_WRONG: One parameter is erroneous > E2E_E_INTERR: Internal library error > E2E_E_OK: State machine check successful > E2E_E_WRONGSTATE: Invalid state machine state
Functional Description	
Processes E2E State Machine and checks the communication channel. It determines if the data can be used for safety-related application, based on history of checks performed by a corresponding E2E_PXXCheck function.	
Particularities and Limitations	
> None	
Expected Caller Context	
> There is no specific caller. E2E can be called from anywhere.	

Table 4-6 E2E_SMCheck

4.3.7 E2E_SMCheckInit

Prototype	
`Std_ReturnType` **`E2E_SMCheckInit`** `(E2E_SMCheckStateType* StatePtr, E2E_SMConfigType* ConfigPtr)`	
Parameter	
`StatePtr`	Pointer to state of the communication channel
`ConfigPtr`	Pointer to static configuration of E2E State Machine
Return code	
`Std_ReturnType`	> E2E_E_INPUTERR_NULL: At least one pointer parameter is NULL > E2E_E_OK: Initialization successful
Functional Description	
Initializes state of the communication channel by resetting all parameters in state structure.	
Particularities and Limitations	
> None	
Expected Caller Context	
> There is no specific caller. E2E can be called from anywhere.	

Table 4-7 E2E_SMCheckInit

4.4 Services used by E2E

In the following table services provided by other components, which are used by the E2E are listed. For details about prototype and functionality refer to the documentation of the providing component.

Component	API
CRC	> Crc_CalculateCRC8 > Crc_CalculateCRC8H2F > Crc_CalculateCRC16 > Crc_CalculateCRC32P4 > Crc_CalculateCRC64 > Crc_CalculateCRC32PJ193976

Table 4-8 Services used by the E2E

Note
The Crc_CalculateCRC32PJ193976 is only used in the profile J1939-76.

4.5 Callback Functions

E2E Library does not provide any callback functions.

4.6 Configurable Interfaces

E2E Library does not provide any configurable interfaces.

5 Configuration

E2E Library is non-generated, deterministic software code, where all inputs and settings are passed by function parameters.

Caution
Although the E2E Library cannot be configured it must be added as an own Module into the DaVinci Configurator Project (“dummy” E2E_bswmd.arxml provided therefore).

For both protection on sender side and checking on receiver side a state structure has to be provided to the services E2E_PXXProtect (sender) and E2E_PXXCheck (receiver). Additionally, both sender and receiver require a configuration structure of type E2E_PXXConfigType, which specifies details about the E2E Header and its containing protection mechanisms. The content of configuration structure is static and needs to be specified by the caller of E2E Library.

Caution
Static configuration for data (e.g. DataID, position of CRC in data array) has to be identical on sender and receiver side.

The configuration of E2E State Machine is static as well and has to be provided by caller via function parameter.

Behavior of Profile 11 changed with AUTOSAR Release R21-11. This can be influenced by setting a corresponding C Preprocessor Define. See chapter 3.6 for more details.

6 Glossary and Abbreviations

6.1 Glossary

Term	Description
E2E	End to end protection

Table 6-1 Glossary

6.2 Abbreviations

Abbreviation	Description
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
BSW	Basis Software
CRC	Cyclic Redundancy Check
DEM	Diagnostic Event Manager
DET	Development Error Tracer
EAD	Embedded Architecture Designer
ECU	Electronic Control Unit
HIS	Hersteller Initiative Software
ISR	Interrupt Service Routine
MICROSAR	Microcontroller Open System Architecture (the Vector AUTOSAR solution)
PPORT	Provide Port
RPORT	Require Port
RTE	Runtime Environment
SRS	Software Requirement Specification
SWC	Software Component
SWS	Software Specification

Table 6-2 Abbreviations

7 Contact

Visit our website for more information on

- News
- Products
- Demo software
- Support
- Training data
- Addresses

www.vector.com