

Document Title	Specification of Ethernet Transceiver Driver
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	431
Document Classification	Standard
Document Version	1.4.0
Document Status	Final
Part of Release	4.1
Revision	2

Document Change History

Date	Version	Changed by	Change Description
31.10.2013	1.4.0	AUTOSAR Release Management	<ul style="list-style-type: none"> • Introduction of Eth_GeneralTypes.h • Support of API deviation for asynchronous implementation • Editorial changes • Removed chapter(s) on change documentation
05.03.2013	1.3.10	AUTOSAR Administration	<ul style="list-style-type: none"> • 1000 kbit Ethernet Support
09.12.2011	1.2.0	AUTOSAR Administration	<ul style="list-style-type: none"> • EthTrcv_GetVersionInfo revised
26.10.2010	1.1.0	AUTOSAR Administration	<ul style="list-style-type: none"> • Further post-build configurable parameters • Configuration enhanced by additional parameter EthTrcvWaitCount • 'Instance ID' removed from Version Info (concerns EthTrcv_GetVersionInfo API) • Additional development error in EthTrcv_GetVersionInfo API • Improved description of 'XxxCtrlIdx' semantics • Specification of behaviour for state switch into already active state
30.11.2009	1.0.0	AUTOSAR Administration	Initial Release

Disclaimer

This specification and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the specification.

The material contained in this specification is protected by copyright and other types of Intellectual Property Rights. The commercial exploitation of the material contained in this specification requires a license to such Intellectual Property Rights.

This specification may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only.

For any other purpose, no part of the specification may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The AUTOSAR specifications have been developed for automotive applications only. They have neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Advice for users

AUTOSAR specifications may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the specifications for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such specifications, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

Table of Contents

1	Introduction and functional overview	6
2	Acronyms and abbreviations	8
3	Related documentation.....	9
3.1	Input documents.....	9
3.2	Related standards and norms	10
3.3	Related specification	10
4	Constraints and assumptions	11
4.1	Limitations	11
4.2	Applicability to car domains	11
5	Dependencies to other modules.....	12
5.1	File structure.....	13
5.1.1	Header file structure.....	13
6	Requirements traceability	14
7	Functional specification	17
7.1	Ethernet BSW stack	17
7.1.1	Indexing scheme.....	17
7.1.2	Requirements.....	18
7.1.3	Configuration description	19
7.2	Development Errors.....	20
7.3	Production Errors.....	20
7.4	Extended Production Errors	20
8	API specification.....	21
8.1	Imported types.....	21
8.2	Type definitions	21
8.2.1	EthTrcv_ConfigType	21
8.2.2	EthTrcv_ModeType.....	22
8.2.3	EthTrcv_LinkStateType.....	22
8.2.4	EthTrcv_StateType	22
8.2.5	EthTrcv_BaudRateType.....	22
8.2.6	EthTrcv_DuplexModeType	23
8.3	Function definitions.....	23
8.3.1	EthTrcv_Init.....	23
8.3.2	EthTrcv_TransceiverInit	24
8.3.3	EthTrcv_SetTransceiverMode.....	25
8.3.4	EthTrcv_GetTransceiverMode	26
8.3.5	EthTrcv_StartAutoNegotiation.....	27
8.3.6	EthTrcv_GetLinkState	28
8.3.7	EthTrcv_GetBaudRate	29
8.3.8	EthTrcv_GetDuplexMode.....	30
8.3.9	EthTrcv_GetVersionInfo.....	32
8.4	Callback notifications.....	32

8.5	Interrupt service routines	32
8.6	Scheduled functions	32
8.7	Expected Interfaces.....	32
8.7.1	Mandatory Interfaces	33
8.7.2	Optional Interfaces.....	33
8.7.3	Configurable interfaces	33
9	Sequence diagrams	34
10	Configuration specification.....	35
10.1	Containers and configuration parameters	36
10.1.1	Variants	37
10.1.2	EthTrcv	38
10.1.3	EthTrcvConfigSet	38
10.1.4	EthTrcvConfig.....	38
10.1.5	EthTrcvDemEventParameterRefs	40
10.1.6	EthTrcvGeneral	40
11	Not applicable requirements	44

Known Limitations

Currently, chapter 5 Dependencies to other modules does not describe the versions of dependent modules. Thus, a version check will extend the chapter.

1 Introduction and functional overview

This specification specifies the functionality, API and the configuration of the AUTOSAR Basic Software module Ethernet Transceiver Driver.

In the AUTOSAR Layered Software Architecture, the Ethernet Transceiver Driver belongs to the *Microcontroller Abstraction Layer*, or more precisely, to the *Communication Drivers*.

This indicates the main task of the Ethernet Transceiver Driver:
Provide to the upper layer (Ethernet Interface) a hardware independent interface comprising multiple equal transceivers. This interface shall be uniform for all transceivers. Thus, the upper layer (Ethernet Interface) may access the underlying bus system in a uniform manner. The configuration of the Ethernet Transceiver Driver however is bus specific, since it takes into account the specific features of the communication transceiver.

A single Ethernet Transceiver Driver module supports only one type of transceiver hardware, but several transceivers of the same type. The Ethernet Transceiver Driver's prefix requires a unique namespace. The Ethernet Interface can access different Ethernet controller types using different Ethernet Transceiver Drivers using this prefix. The decision which driver to use to access a particular transceiver is a configuration parameter of the Ethernet Interface.

Figure 1.1 depicts the lower part of the Ethernet stack. One Ethernet Interface accesses several transceivers using one or several Ethernet Transceiver Drivers.

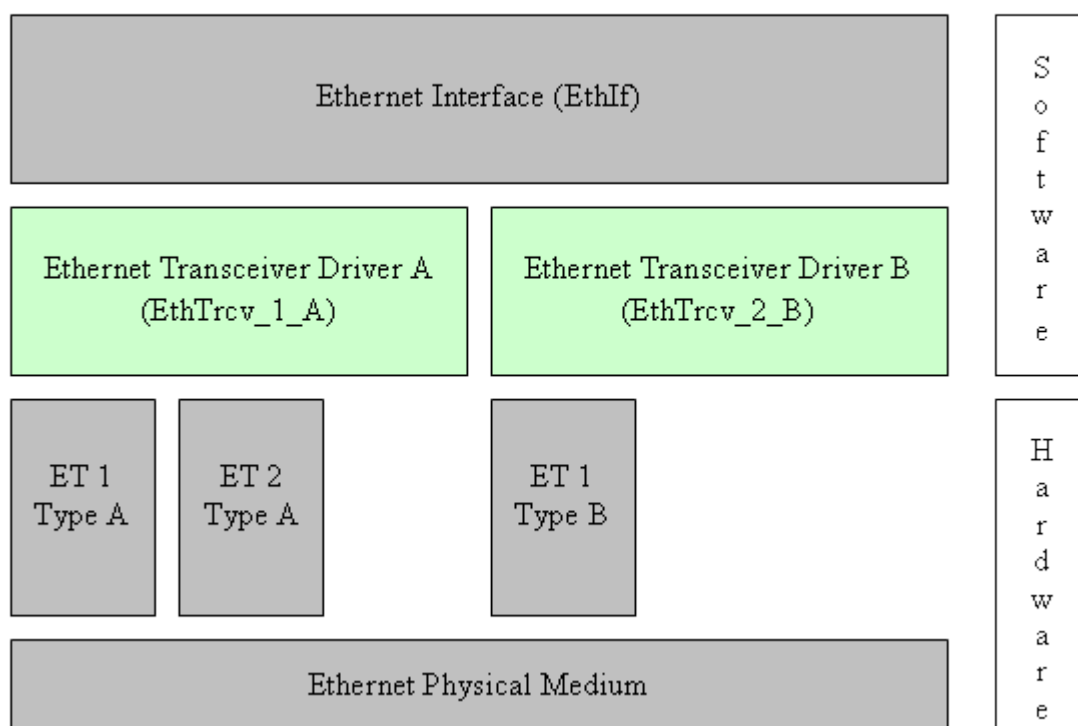


Figure 1.1: Ethernet stack module overview

Note: The Ethernet Transceiver Driver is specified in a way that allows for object code delivery of the code module, following the "one-fits-all" principle, i.e. the entire configuration of the Ethernet Interface can be carried out without modifying any source code. Thus, the configuration of the Ethernet Transceiver Driver can be carried out largely without detailed knowledge of the Ethernet Transceiver Driver software.

2 Acronyms and abbreviations

Abbreviation / Acronym:	Description:
EC	Ethernet controller
ET	Ethernet transceiver
Eth	Ethernet Controller Driver (AUTOSAR BSW module)
EthIf	Ethernet Interface (AUTOSAR BSW module)
EthTrcv	Ethernet Transceiver Driver (AUTOSAR BSW module)
MCG	Module Configuration Generator
MII	Media Independent Interface (standardized Interface provided by Ethernet controllers to access Ethernet transceivers, see [21])

3 Related documentation

3.1 Input documents

- [1] List of Basic Software Modules
AUTOSAR_TR_BSWModuleList.pdf

- [2] Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf

- [3] AUTOSAR General Requirements on Basic Software Modules
AUTOSAR_SRS_BSWGeneral.pdf

- [4] Specification of Communication
AUTOSAR_SWS_COM.pdf

- [5] Requirements on Ethernet Support in AUTOSAR
AUTOSAR_SRS_Ethernet.pdf

- [6] Specification of Ethernet Interface
AUTOSAR_SWS_EthernetInterface.pdf

- [7] Specification of Ethernet State Manager
AUTOSAR_SWS_EthernetStateManager.pdf

- [8] Specification of Ethernet Interface
AUTOSAR_SWS_EthernetInterface.pdf

- [9] Specification of Socket Adapter
AUTOSAR_SWS_SocketAdapter.pdf

- [10] Specification of UDP Network Management
AUTOSAR_SWS_UDPNetworkManagement.pdf

- [11] Specification of PDU Router
AUTOSAR_SWS_PDURouter.pdf

- [12] BSW Scheduler Specification
AUTOSAR_SWS_Scheduler.pdf

- [13] Specification of ECU Configuration
AUTOSAR_TPS_ECUConfiguration.pdf

- [14] Specification of Memory Mapping
AUTOSAR_SWS_MemoryMapping.pdf

- [15] Specification of Standard Types
AUTOSAR_SWS_StandardTypes.pdf

[16] Specification of Development Error Tracer
AUTOSAR_SWS_DevelopmentErrorTracer.pdf

[17] Specification of Diagnostics Event Manager
AUTOSAR_SWS_DiagnosticEventManager

[18] Specification of C Implementation Rules
AUTOSAR_TR_CImplementationRules.pdf

[19] Specification of ECU State Manager
AUTOSAR_SWS_ECUSTateManager.pdf

[20] General Specification of Basic Software Modules
AUTOSAR_SWS_BSWGeneral.pdf

3.2 Related standards and norms

[20] IEC 7498-1 The Basic Model, IEC Norm, 1994

[21] IEEE 802.3-2006

3.3 Related specification

AUTOSAR provides a General Specification on Basic Software modules [20] (SWS BSW General), which is also valid for Ethernet Transceiver Driver.

Thus, the specification SWS BSW General shall be considered as additional and required specification for Ethernet Transceiver Driver.

4 Constraints and assumptions

4.1 Limitations

The Ethernet Transceiver Driver module is only able to handle a single thread of execution. The execution must not be pre-empted by itself.

The implementation is limited to 10MBit and 100MBit Ethernet and transceivers connected via Media Independent Interface (MII).

Depending on the Ethernet hardware, it may become necessary that implementations deviate from API specifications in respect to the asynchronous/synchronous behaviour.

4.2 Applicability to car domains

The Ethernet BSW stack is intended to be used wherever high data rates are required but no hard real-time is required. Of course, it can also be used for less-demanding use cases, i.e. for low data rates.

5 Dependencies to other modules

This chapter lists the modules interacting with the Ethernet Transceiver Driver module.

Modules that use Ethernet Transceiver Driver module:

- Ethernet Interface (EthIf)

Modules used by the Ethernet Transceiver Driver module:

- Ethernet Controller Driver (Eth) for transceiver access via Media Independent Interface (MII).

Dependencies to other Modules:

- On certain systems the transceiver might share resources with other components (e.g. the MCU, Port), and may depend on their configuration. If those resources are within scope of the other modules (e.g. PLL configuration, memory mapping, etc.) the Ethernet Transceiver Driver module does not take care of configuring those components but requires their preceding initialization.

5.1 File structure

5.1.1 Header file structure

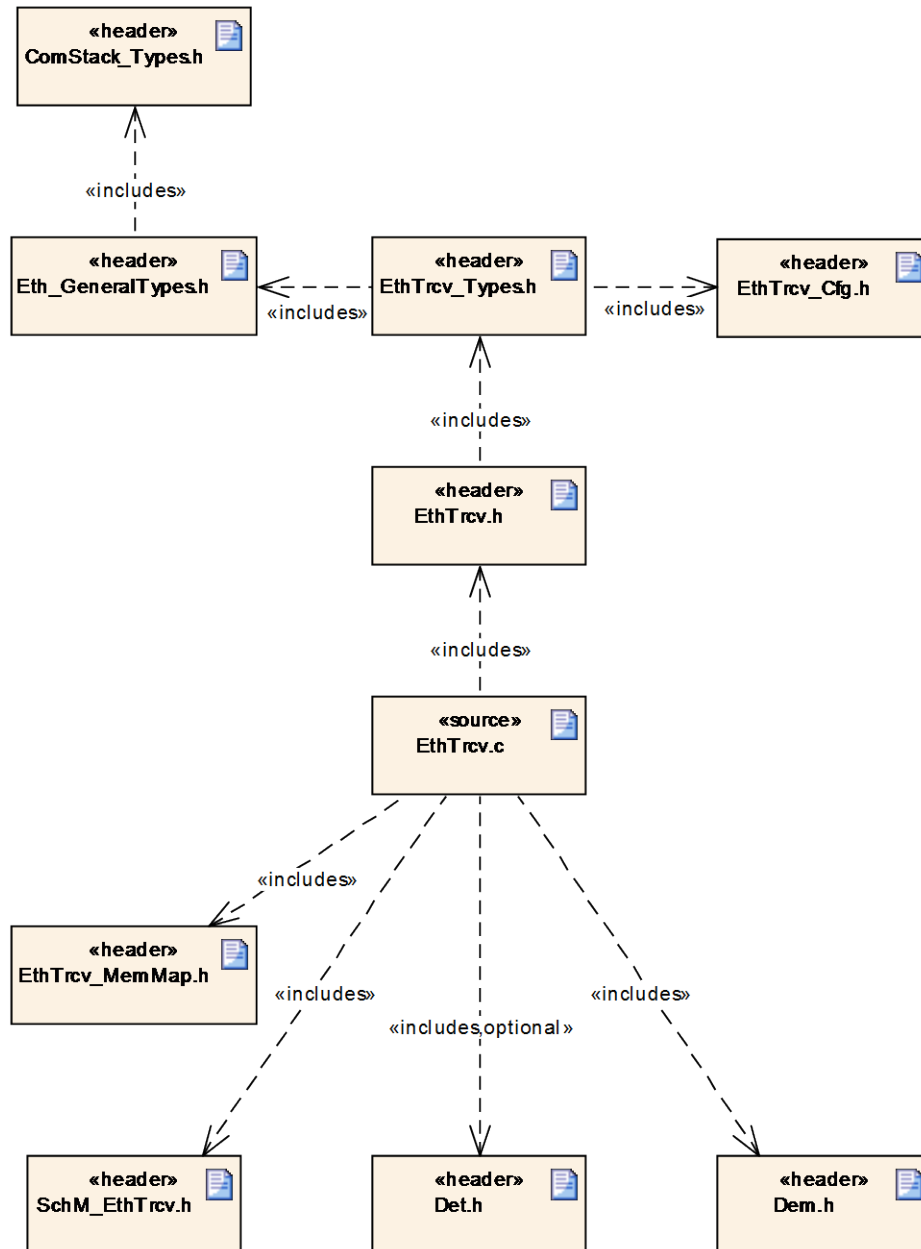


Figure 5.1: Ethernet Transceiver Driver file structure

6 Requirements traceability

Requirement	Description	Satisfied by
-	-	SWS_EthTrcv_00003
-	-	SWS_EthTrcv_00004
-	-	SWS_EthTrcv_00005
-	-	SWS_EthTrcv_00006
-	-	SWS_EthTrcv_00007
-	-	SWS_EthTrcv_00008
-	-	SWS_EthTrcv_00009
-	-	SWS_EthTrcv_00010
-	-	SWS_EthTrcv_00011
-	-	SWS_EthTrcv_00012
-	-	SWS_EthTrcv_00013
-	-	SWS_EthTrcv_00014
-	-	SWS_EthTrcv_00015
-	-	SWS_EthTrcv_00027
-	-	SWS_EthTrcv_00028
-	-	SWS_EthTrcv_00029
-	-	SWS_EthTrcv_00030
-	-	SWS_EthTrcv_00031
-	-	SWS_EthTrcv_00032
-	-	SWS_EthTrcv_00033
-	-	SWS_EthTrcv_00034
-	-	SWS_EthTrcv_00035
-	-	SWS_EthTrcv_00036
-	-	SWS_EthTrcv_00037
-	-	SWS_EthTrcv_00038
-	-	SWS_EthTrcv_00039
-	-	SWS_EthTrcv_00040
-	-	SWS_EthTrcv_00041
-	-	SWS_EthTrcv_00042
-	-	SWS_EthTrcv_00043
-	-	SWS_EthTrcv_00044
-	-	SWS_EthTrcv_00045
-	-	SWS_EthTrcv_00046
-	-	SWS_EthTrcv_00047
-	-	SWS_EthTrcv_00048

-	-	SWS_EthTrcv_00049
-	-	SWS_EthTrcv_00050
-	-	SWS_EthTrcv_00051
-	-	SWS_EthTrcv_00052
-	-	SWS_EthTrcv_00053
-	-	SWS_EthTrcv_00054
-	-	SWS_EthTrcv_00055
-	-	SWS_EthTrcv_00056
-	-	SWS_EthTrcv_00057
-	-	SWS_EthTrcv_00058
-	-	SWS_EthTrcv_00059
-	-	SWS_EthTrcv_00060
-	-	SWS_EthTrcv_00061
-	-	SWS_EthTrcv_00062
-	-	SWS_EthTrcv_00063
-	-	SWS_EthTrcv_00064
-	-	SWS_EthTrcv_00065
-	-	SWS_EthTrcv_00066
-	-	SWS_EthTrcv_00067
-	-	SWS_EthTrcv_00068
-	-	SWS_EthTrcv_00069
-	-	SWS_EthTrcv_00070
-	-	SWS_EthTrcv_00071
-	-	SWS_EthTrcv_00072
-	-	SWS_EthTrcv_00073
-	-	SWS_EthTrcv_00074
-	-	SWS_EthTrcv_00075
-	-	SWS_EthTrcv_00076
-	-	SWS_EthTrcv_00077
-	-	SWS_EthTrcv_00078
-	-	SWS_EthTrcv_00079
-	-	SWS_EthTrcv_00080
-	-	SWS_EthTrcv_00081
-	-	SWS_EthTrcv_00082
-	-	SWS_EthTrcv_00085
-	-	SWS_EthTrcv_00086
-	-	SWS_EthTrcv_00088
-	-	SWS_EthTrcv_00089
-	-	SWS_EthTrcv_00090

-	-	SWS_EthTrcv_00093
-	-	SWS_EthTrcv_00094
-	-	SWS_EthTrcv_00097
-	-	SWS_EthTrcv_00098
-	-	SWS_EthTrcv_00099
-	-	SWS_EthTrcv_00100
-	-	SWS_EthTrcv_00101
-	-	SWS_EthTrcv_00102
-	-	SWS_EthTrcv_00103
BSW00170	-	SWS_EthTrcv_00999

7 Functional specification

7.1 Ethernet BSW stack

As part of the AUTOSAR Layered Software Architecture according to Figure 7.1, the Ethernet BSW modules also form a layered software stack. Figure 7.1 depicts the basic structure of this Ethernet BSW stack. The EthIf module accesses several transceivers using the Ethernet Transceiver Driver layer, which can be made up of several Ethernet Transceiver Drivers modules.

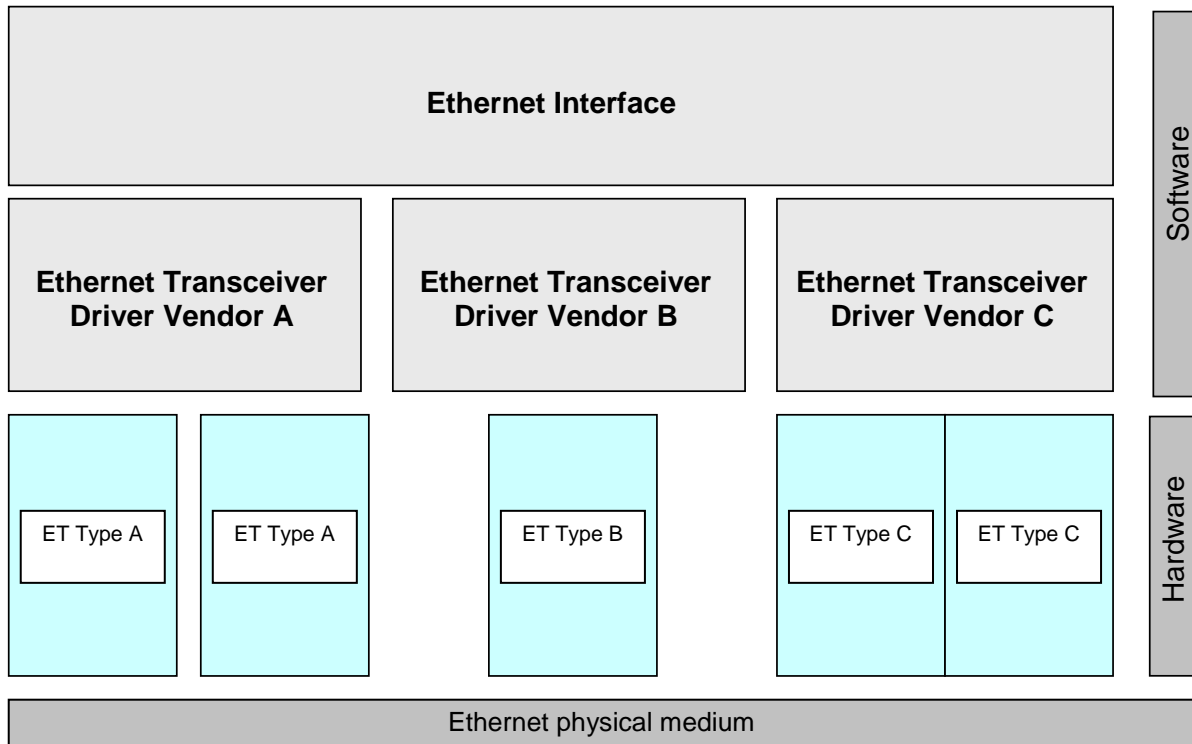


Figure 7.1: Basic Structure of the Ethernet BSW stack

7.1.1 Indexing scheme

Users of the Ethernet Transceiver Driver identify transceiver resources using an indexing scheme as depicted in Figure 7.2.

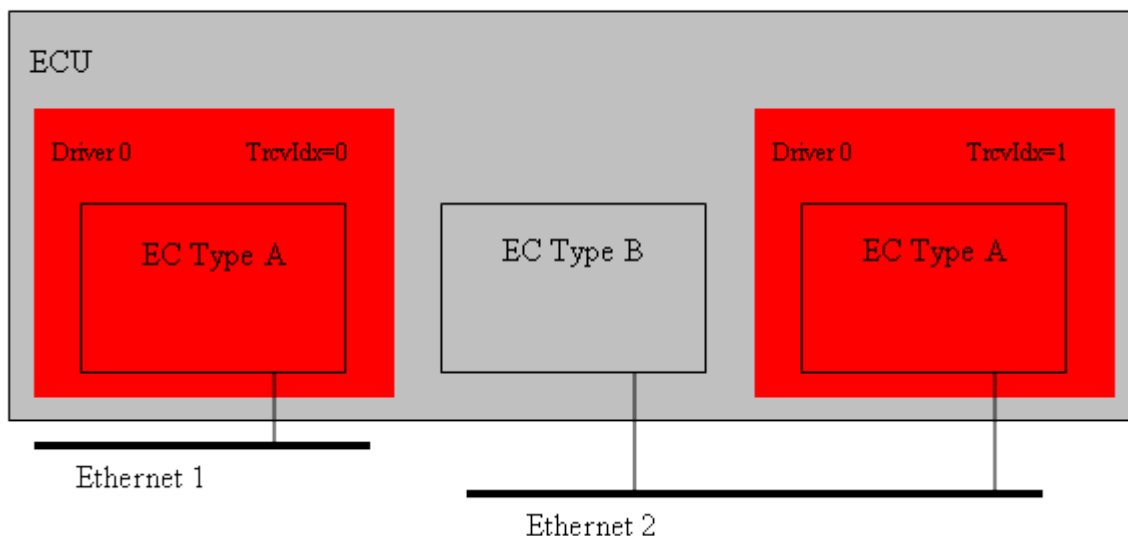


Figure 7.2: Ethernet Transceiver Driver indexing scheme

[SWS_EthTrcv_00003] ⌈

The Ethernet Transceiver Driver is using a zero-based index to abstract the access for upper software layers. The parameter `EthTrcv_CtrlIdx` within configuration corresponds to parameter `TrcvIdx` used in the API. ⌋()

7.1.2 Requirements

This chapter lists requirements that shall be fulfilled by Ethernet Transceiver Driver module implementations.

The Ethernet Interface module environment comprises all modules which are calling interfaces of the Ethernet Interface module.

[SWS_EthTrcv_00004] ⌈

The Ethernet Transceiver Driver module shall support pre-compile time, link time and post-build time configuration. ⌋()

[SWS_EthTrcv_00005] ⌈

The header file `EthTrcv.h` shall include a software and specification version number. ⌋()

[SWS_EthTrcv_00006] ⌈

The Ethernet Transceiver Driver module shall perform a consistency check between code files and header files based on pre-process-checking the version numbers of related code files and header files. ⌋()

[SWS_EthTrcv_00007] ⌈

In case development error detection is enabled for the Ethernet Transceiver Driver module: The Ethernet Transceiver Driver module shall check API parameters for validity and report detected errors to the DET. ⌋()

DET API functions are specified in [16].

[SWS_EthTrcv_00008] ⌈

The Ethernet Transceiver Driver module implementation shall conform to the HIS subset of the MISRA C Standard (see document [18]). ⌋()

[SWS_EthTrcv_00009] ⌈

The Ethernet Transceiver Driver module shall implement the API functions specified by the Ethernet Transceiver Driver SWS as real C-code functions and shall not implement the API as macros for object code deliveries. ⌋()

[SWS_EthTrcv_00010] ⌈

None of the Ethernet Transceiver Driver module header files shall define global variables. ⌋()

7.1.3 Configuration description

[SWS_EthTrcv_00011] ⌈

The Ethernet Transceiver Driver module shall provide an XML file that contains the data, which is required for the SW identification (it shall contain the vendor identification, module ID and software version information), configuration and integration process. This file should describe vendor specific configuration parameters as well as it should contain recommended configuration parameter values. ⌋()

[SWS_EthTrcv_00012] ⌈

The MCG shall read the ECU configuration description of the Ethernet Driver module(s). Ethernet Driver related configuration data is contained in the Ethernet Driver module configuration description. ⌋()

[SWS_EthTrcv_00013] ⌈

The MCG shall ensure the consistency of the generated configuration data. ⌋()

[SWS_EthTrcv_00014] ⌈

The configuration of the Ethernet Transceiver Driver module shall be calculated at ECU configuration time. None of the communication parameters shall be calculated at runtime. »()

[SWS_EthTrcv_00015] ⌈

The start address of post-build time configuration data shall be passed during module initialization (see chapter 8.3.1). »()

An assignment of those configuration classes to configuration parameters can be found in chapter 10.

A detailed description of all Ethernet Transceiver Driver related configuration parameters can be found in chapter 10 of this document.

7.2 Development Errors

[SWS_EthTrcv_00017] ⌈

<i>Type or error</i>	<i>Relevance</i>	<i>Related error code</i>	<i>Value [hex]</i>
Invalid transceiver index	Development	ETHTRCV_E_INV_TRCV_IDX	0x01
EthTrcv module was not initialized	Development	ETHTRCV_E_NOT_INITIALIZED	0x02
Invalid pointer in parameter list	Development	ETHTRCV_E_INV_POINTER	0x03
Invalid configuration	Development	ETHTRCV_E_INV_CONFIG	0x04

7.3 Production Errors

None

7.4 Extended Production Errors

[SWS_EthTrcv_00097] ⌈

<i>Type or error</i>	<i>Related error code</i>	<i>Value [hex]</i>
Ethernet Transceiver Access Failure	ETHTRCV_E_ACCESS	Assigned by DEM

»()

8 API specification

8.1 Imported types

This chapter lists all types included from the following files:

[SWS_EthTrcv_00027] ⌈

Module	Imported Type
ComStack_Types	BufReq_ReturnType
Dem	Dem_EventIdType
	Dem_EventStatusType
Eth_GeneralTypes	EthTrcv_BaudRateType
	EthTrcv_DuplexModeType
	EthTrcv_LinkStateType
	EthTrcv_ModeType
	Eth_FrameType
	Eth_ModeType
	Eth_ReturnType
	Eth_RxStatusType
	EthTrcv_ConfigType
	Eth_ConfigType
Std_Types	Std_ReturnType
	Std_VersionInfoType

⌋()

8.2 Type definitions

[ETHTRCV095]:

EthTrcv.h shall include Eth_GeneralTypes.h for include of general EthTrcv type declarations.

[ETHTRCV096]:

The types specified in SWS_EthernetTransceiverDriver shall be declared in Eth_GeneralTypes.h

8.2.1 EthTrcv_ConfigType

[SWS_EthTrcv_00098] ⌈

Name:	EthTrcv_ConfigType
Type:	Structure
Range:	Implementation specific.
Description:	Implementation specific structure of the post build configuration

⌋()

8.2.2 EthTrcv_ModeType

[SWS_EthTrcv_00099]┌

Name:	EthTrcv_ModeType	
Type:	Enumeration	
Range:	ETHTRCV_MODE_DOWN	0x00: Transceiver disabled
	ETHTRCV_MODE_ACTIVE	0x01: Transceiver enabled
Description:	This type defines the transceiver modes	

└()

8.2.3 EthTrcv_LinkStateType

[SWS_EthTrcv_00100]┌

Name:	EthTrcv_LinkStateType	
Type:	Enumeration	
Range:	ETHTRCV_LINK_STATE_DOWN	0x00: No physical Ethernet connection established
	ETHTRCV_LINK_STATE_ACTIVE	0x01: Physical Ethernet connection established
Description:	This type defines the Ethernet link state. The link state changes after an Ethernet cable gets plugged in and the transceivers on both ends negotiated the transmission parameters (i.e. baud rate and duplex mode)	

└()

8.2.4 EthTrcv_StateType

[SWS_EthTrcv_00101]┌

Name:	EthTrcv_StateType	
Type:	Enumeration	
Range:	ETHTRCV_STATE_UNINIT	0x00: Driver is not yet configured
	ETHTRCV_STATE_INIT	0x01: Driver is configured
	ETHTRCV_STATE_ACTIVE	0x02: Driver is active
Description:	Status supervision used for Development Error Detection. The state shall be available for debugging.	

└()

8.2.5 EthTrcv_BaudRateType

[SWS_EthTrcv_00102]┌

Name:	EthTrcv_BaudRateType	
Type:	Enumeration	
Range:	ETHTRCV_BAUD_RATE_10MBIT	0x00: 10MBIT Ethernet connection
	ETHTRCV_BAUD_RATE_100MBIT	0x01: 100MBIT Ethernet connection
	ETHTRCV_BAUD_RATE_1000MBIT	0x02: 1000MBIT Ethernet connection
Description:	This type defines the Ethernet baud rate. The baud rate gets either negotiated between the connected transceivers or has to be configured.	

└()

8.2.6 EthTrcv_DuplexModeType

[SWS_EthTrcv_00103]┐

Name:	EthTrcv_DuplexModeType	
Type:	Enumeration	
Range:	ETHTRCV_DUPLEX_MODE_HALF	0x00: Half duplex Ethernet connection
	ETHTRCV_DUPLEX_MODE_FULL	0x01: Full duplex Ethernet connection
Description:	This type defines the Ethernet duplex mode. The duplex mode gets either negotiated between the connected transceivers or has to be configured.	

┘()

8.3 Function definitions

This is a list of functions provided for upper layer modules.

8.3.1 EthTrcv_Init

[SWS_EthTrcv_00028]┐

Service name:	EthTrcv_Init		
Syntax:	void		EthTrcv_Init(const EthTrcv_ConfigType* CfgPtr)
Service ID[hex]:	0x01		
Sync/Async:	Synchronous		
Reentrancy:	Non Reentrant		
Parameters (in):	CfgPtr	Points to the implementation specific structure	
Parameters (inout):	None		
Parameters (out):	None		
Return value:	None		
Description:	Initializes the Ethernet Transceiver Driver		

┘()

[SWS_EthTrcv_00029]┐

The function shall store the access to the configuration structure for subsequent API calls.┘()

[SWS_EthTrcv_00030]┐

The function shall change the state of the component from ETHTRCV_STATE_UNINIT to ETHTRCV_STATE_INIT.┘()

[SWS_EthTrcv_00031]┐

If development error detection is enabled: the function shall check the parameter CfgPtr for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_POINTER.┘()

[SWS_EthTrcv_00032] ⌈

Caveat: The API has to be called during initialization. ⌋()

[SWS_EthTrcv_00033] ⌈

Configuration: The user shall pass the post-build configuration or a NULL_PTR as parameter depending on the configuration variant. ⌋()

8.3.2 EthTrcv_TransceiverInit

[SWS_EthTrcv_00034] ⌈

Service name:	EthTrcv_TransceiverInit	
Syntax:	Std_ReturnType EthTrcv_TransceiverInit(uint8 TrcvIdx, uint8 CfgIdx)	
Service ID[hex]:	0x02	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
	CfgIdx	Index of the configuration
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: success E_NOT_OK: transceiver could not be initialized
	Description: Initializes the indexed transceiver	

⌋()

[SWS_EthTrcv_00035] ⌈

The function shall:

- Configure all transceiver configuration parameters (e.g. baud rate, duplex mode, automatic negotiation, ...) ⌋()

[SWS_EthTrcv_00036] ⌈

The function shall change the state of the component from ETHTRCV_STATE_INIT to ETHTRCV_STATE_ACTIVE. ⌋()

[SWS_EthTrcv_00037] ⌈

If development error detection is enabled: the function shall check that the service EthTrcv_Init was previously called. If the check fails, the function shall raise the development error ETHTRCV_E_NOT_INITIALIZED and return E_NOT_OK. ⌋()

[SWS_EthTrcv_00038] ⌈

If development error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_TRCV_IDX and return E_NOT_OK.>()

[SWS_EthTrcv_00039] ⌈

If development error detection is enabled: the function shall check the parameter CfgIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_CONFIG and return E_NOT_OK.>()

[SWS_EthTrcv_00040] ⌈

The function shall check the access to the Ethernet controller. If the check fails, the function shall raise the production error ETHTRCV_E_ACCESS and return E_NOT_OK.>()

[SWS_EthTrcv_00041] ⌈

Caveat: The function requires previous initialization (EthTrcv_Init.>()

8.3.3 EthTrcv_SetTransceiverMode

[SWS_EthTrcv_00042] ⌈

Service name:	EthTrcv_SetTransceiverMode	
Syntax:	Std_ReturnType EthTrcv_SetTransceiverMode(uint8 TrcvIdx, EthTrcv_ModeType CtrlMode)	
Service ID[hex]:	0x03	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
	CtrlMode	ETHTRCV_MODE_DOWN: disable the transceiver ETHTRCV_MODE_ACTIVE: enable the transceiver
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: success E_NOT_OK: transceiver could not be initialized
Description:	Enables / disables the indexed transceiver	

()

[SWS_EthTrcv_00043] ⌈

The function shall put the index transceiver in the specified mode.>()

[SWS_EthTrcv_00044] ⌈

If development error detection is enabled: the function shall check that the service EthTrcv_TransceiverInit was previously called. If the check fails, the function shall

raise the development error ETHTRCV_E_NOT_INITIALIZED and return E_NOT_OK.」()

[SWS_EthTrcv_00045] 「

If development error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_TRCV_IDX and return E_NOT_OK.」()

[SWS_EthTrcv_00046] 「

The function shall be pre compile time configurable On/Off by the configuration parameter: EthTrcvSetTransceiverModeApi.」()

[SWS_EthTrcv_00094] 「

If the transceiver is already in the requested mode E_OK shall be returned and no development error shall be raised.」()

[SWS_EthTrcv_00047] 「

Caveat: The function requires previous transceiver initialization (EthTrcv_TransceiverInit).」()

8.3.4 EthTrcv_GetTransceiverMode

[SWS_EthTrcv_00048] 「

Service name:	EthTrcv_GetTransceiverMode	
Syntax:	Std_ReturnType EthTrcv_GetTransceiverMode(uint8 TrcvIdx, EthTrcv_ModeType* TrcvModePtr)	
Service ID[hex]:	0x04	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
Parameters (inout):	None	
Parameters (out):	TrcvModePtr	ETHTRCV_MODE_DOWN: the transceiver is disabled ETHTRCV_MODE_ACTIVE: the transceiver is enable
Return value:	Std_ReturnType	E_OK: success E_NOT_OK: transceiver could not be initialized
Description:	Obtains the state of the indexed transceiver	

」()

[SWS_EthTrcv_00049] 「

The function shall read the current transceiver mode.」()

[SWS_EthTrcv_00050] ⌈

If development error detection is enabled: the function shall check that the service EthTrcv_TransceiverInit was previously called. If the check fails, the function shall raise the development error ETHTRCV_E_NOT_INITIALIZED and return E_NOT_OK.⌋()

[SWS_EthTrcv_00051] ⌈

If development error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_TRCV_IDX and return E_NOT_OK.⌋()

[SWS_EthTrcv_00052] ⌈

If development error detection is enabled: the function shall check the parameter TrcvModePtr for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_POINTER and return E_NOT_OK.⌋()

[SWS_EthTrcv_00053] ⌈

The function shall be pre compile time configurable On/Off by the configuration parameter: EthTrcvGetTransceiverModeApi.⌋()

[SWS_EthTrcv_00054] ⌈

Caveat: The function requires previous transceiver initialization (EthTrcv_TransceiverInit).⌋()

8.3.5 EthTrcv_StartAutoNegotiation

[SWS_EthTrcv_00055] ⌈

Service name:	EthTrcv_StartAutoNegotiation	
Syntax:	Std_ReturnType EthTrcv_StartAutoNegotiation(uint8 TrcvIdx)	
Service ID[hex]:	0x05	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: success E_NOT_OK: transceiver could not be initialized
Description:	Restarts the negotiation of the transmission parameters used by the indexed transceiver	

⌋()

[SWS_EthTrcv_00056] ⌈

The function shall restart the automatic negotiation of the transmission parameters used by the indexed transceiver.)()

[SWS_EthTrcv_00057] ⌈

If development error detection is enabled: the function shall check that the service EthTrcv_TransceiverInit was previously called. If the check fails, the function shall raise the development error ETHTRCV_E_NOT_INITIALIZED and return E_NOT_OK.)()

[SWS_EthTrcv_00058] ⌈

If development error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_TRCV_IDX and return E_NOT_OK.)()

[SWS_EthTrcv_00059] ⌈

The function shall be pre compile time configurable On/Off by the configuration parameter: EthTrcvStartAutoNegotiationApi.)()

[SWS_EthTrcv_00060] ⌈

Caveat: The function requires previous transceiver initialization (EthTrcv_TransceiverInit).)()

[SWS_EthTrcv_00088] ⌈

Caveat: The function is not required or called by an upper layer BSW software component.)()

8.3.6 EthTrcv_GetLinkState

[SWS_EthTrcv_00061] ⌈

Service name:	EthTrcv_GetLinkState		
Syntax:	Std_ReturnType	EthTrcv_GetLinkState(
		uint8	TrcvIdx,
	EthTrcv_LinkStateType*		LinkStatePtr
)		
Service ID[hex]:	0x06		
Sync/Async:	Synchronous		
Reentrancy:	Non Reentrant		
Parameters (in):	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver	
Parameters (inout):	None		
Parameters (out):	LinkStatePtr	ETHTRCV_LINK_STATE_DOWN: transceiver is connected ETHTRCV_LINK_STATE_ACTIVE: transceiver is disconnected	
Return value:	Std_ReturnType	E_OK:	success
		E_NOT_OK:	transceiver could not be initialized

Description:	Obtains the link state of the indexed transceiver
---------------------	---

」()

[SWS_EthTrcv_00062] 「

The function shall read the current transceiver link state.」()

[SWS_EthTrcv_00063] 「

If development error detection is enabled: the function shall check that the service EthTrcv_TransceiverInit was previously called. If the check fails, the function shall raise the development error ETHTRCV_E_NOT_INITIALIZED and return

E_NOT_OK.」()

[SWS_EthTrcv_00064] 「

If development error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_TRCV_IDX and return E_NOT_OK.」()

[SWS_EthTrcv_00065] 「

If development error detection is enabled: the function shall check the parameter LinkStatePtr for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_POINTER and return E_NOT_OK.」()

[SWS_EthTrcv_00066] 「

The function shall be pre compile time configurable On/Off by the configuration parameter: EthTrcvGetLinkStateApi.」()

[SWS_EthTrcv_00067] 「

Caveat: The function requires previous transceiver initialization (EthTrcv_TransceiverInit).」()

8.3.7 EthTrcv_GetBaudRate

[SWS_EthTrcv_00068] 「

Service name:	EthTrcv_GetBaudRate	
Syntax:	Std_ReturnType	EthTrcv_GetBaudRate(uint8 TrcvIdx, EthTrcv_BaudRateType* BaudRatePtr)
Service ID[hex]:	0x07	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
Parameters (inout):	None	

Parameters (out):	BaudRatePtr	ETHTRCV_BAUD_RATE_10MBIT: 10MBit connection ETHTRCV_BAUD_RATE_100MBIT: 100MBit connection ETHTRCV_BAUD_RATE_1000MBIT: 1000MBit connection
Return value:	Std_ReturnType	E_OK: success E_NOT_OK: transceiver could not be initialized
Description:	Obtains the baud rate of the indexed transceiver	

」()

[SWS_EthTrcv_00069] 「

The function shall read the current transceiver baud rate. 」()

[SWS_EthTrcv_00070] 「

If development error detection is enabled: the function shall check that the service EthTrcv_TransceiverInit was previously called. If the check fails, the function shall raise the development error ETHTRCV_E_NOT_INITIALIZED and return E_NOT_OK. 」()

[SWS_EthTrcv_00071] 「

If development error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_TRCV_IDX and return E_NOT_OK. 」()

[SWS_EthTrcv_00072] 「

If development error detection is enabled: the function shall check the parameter BaudRatePtr for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_POINTER and return E_NOT_OK. 」()

[SWS_EthTrcv_00073] 「

The function shall be pre compile time configurable On/Off by the configuration parameter: EthTrcvGetBaudRateApi. 」()

[SWS_EthTrcv_00074] 「

Caveat: The function requires previous transceiver initialization (EthTrcv_TransceiverInit). 」()

[SWS_EthTrcv_00089] 「

Caveat: The function is not required or called by an upper layer BSW software component. 」()

8.3.8 EthTrcv_GetDuplexMode

[SWS_EthTrcv_00075] 「

Service name:	EthTrcv_GetDuplexMode	
Syntax:	Std_ReturnType EthTrcv_GetDuplexMode(uint8 TrcvIdx, EthTrcv_DuplexModeType* DuplexModePtr)	
Service ID[hex]:	0x08	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
Parameters (inout):	None	
Parameters (out):	DuplexModePtr	ETHTRCV_DUPLEX_MODE_HALF: half duplex connections ETHTRCV_DUPLEX_MODE_FULL: full duplex connection
Return value:	Std_ReturnType	E_OK: success E_NOT_OK: transceiver could not be initialized
Description:	Obtains the duplex mode of the indexed transceiver	

」()

[SWS_EthTrcv_00076] 「

The function shall read the current transceiver duplex mode.」()

[SWS_EthTrcv_00077] 「

If development error detection is enabled: the function shall check that the service EthTrcv_TransceiverInit was previously called. If the check fails, the function shall raise the development error ETHTRCV_E_NOT_INITIALIZED and return E_NOT_OK.」()

[SWS_EthTrcv_00078] 「

If development error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_TRCV_IDX and return E_NOT_OK.」()

[SWS_EthTrcv_00079] 「

If development error detection is enabled: the function shall check the parameter DuplexModePtr for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_POINTER and return E_NOT_OK.」()

[SWS_EthTrcv_00080] 「

The function shall be pre compile time configurable On/Off by the configuration parameter: EthTrcvGetDuplexModeApi.」()

[SWS_EthTrcv_00081] 「

Caveat: The function requires previous transceiver initialization (EthTrcv_TransceiverInit).」()

[SWS_EthTrcv_00090] ⌈

Caveat: The function is not required or called by an upper layer BSW software component. ⌋()

8.3.9 EthTrcv_GetVersionInfo

[SWS_EthTrcv_00082] ⌈

Service name:	EthTrcv_GetVersionInfo	
Syntax:	void	EthTrcv_GetVersionInfo(Std_VersionInfoType* VersionInfoPtr)
Service ID[hex]:	0x0b	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	None	
Parameters (inout):	None	
Parameters (out):	VersionInfoPtr	Version information of this module
Return value:	None	
Description:	Returns the version information of this module	

⌋()

[SWS_EthTrcv_00093] ⌈

If development error detection is enabled: the function shall check the parameter VersionInfoPtr for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_POINTER. ⌋()

8.4 Callback notifications

The Ethernet Transceiver Driver does not provide any callback functions.

8.5 Interrupt service routines

The Ethernet Transceiver Driver does not provide any interrupt service routines.

8.6 Scheduled functions

The Ethernet Transceiver Driver runs in the context of the Ethernet Interface and has thus no scheduled functions.

8.7 Expected Interfaces

This chapter lists all interfaces required from other modules.

8.7.1 Mandatory Interfaces

This chapter defines all interfaces required to fulfill the core functionality of the module.

[SWS_EthTrcv_00085] ⌈

<i>API function</i>	<i>Description</i>
Dem_ReportErrorStatus	Queues the reported events from the BSW modules (API is only used by BSW modules). The interface has an asynchronous behavior, because the processing of the event is done within the Dem main function. OBD Events Suppression shall be ignored for this computation.
Eth_ControllerInit	Initializes the indexed controller
Eth_GetControllerMode	Obtains the state of the indexed controller
Eth_GetCounterState	Reads the value of a counter specified with its memory offset
Eth_GetPhysAddr	Obtains the physical source address used by the indexed controller
Eth_GetVersionInfo	Returns the version information of this module
Eth_Init	Initializes the Ethernet Driver
Eth_ProvideTxBuffer	Provides access to a transmit buffer of the specified controller
Eth_ReadMii	Reads a transceiver register
Eth_Receive	Triggers frame reception
Eth_SetControllerMode	Enables / disables the indexed controller
Eth_Transmit	Triggers transmission of a previously filled transmit buffer
Eth_TxConfirmation	Triggers frame transmission confirmation
Eth_WriteMii	Configures a transceiver register or triggers a function offered by the receiver
SchM_Enter_EthTrcv	Invokes the SchM_Enter function to enter a module local exclusive area.
SchM_Exit_EthTrcv	Invokes the SchM_Exit function to exit an exclusive area.

⌋()

8.7.2 Optional Interfaces

This chapter defines all interfaces required to fulfill an optional functionality of the module.

[SWS_EthTrcv_00086] ⌈

<i>API function</i>	<i>Description</i>
Det_ReportError	Service to report development errors.

⌋()

8.7.3 Configurable interfaces

The Ethernet Transceiver Driver does not use configurable interfaces.

Terms and definitions:

Reentrant: interface is expected to be reentrant

Don't care: reentrancy of interface not relevant for this module (in general it is in this case not reentrant).

9 Sequence diagrams

The usage of the Ethernet Transceiver Driver is depicted in the sequence diagrams of the Ethernet Interface.

10 Configuration specification

In general, this chapter defines configuration parameters and their clustering into containers. In order to support the specification Chapter 10.1 describes fundamentals. It also specifies a template (table) you shall use for the parameter specification. We intend to leave Chapter 10.1 in the specification to guarantee comprehension.

Chapter 10.2 specifies the structure (containers) and the parameters of the module Ethernet Transceiver Driver.

Chapter 10.3 specifies published information of the module Ethernet Transceiver Driver.

10.1 Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters describe Chapters 7 and Chapter 10.

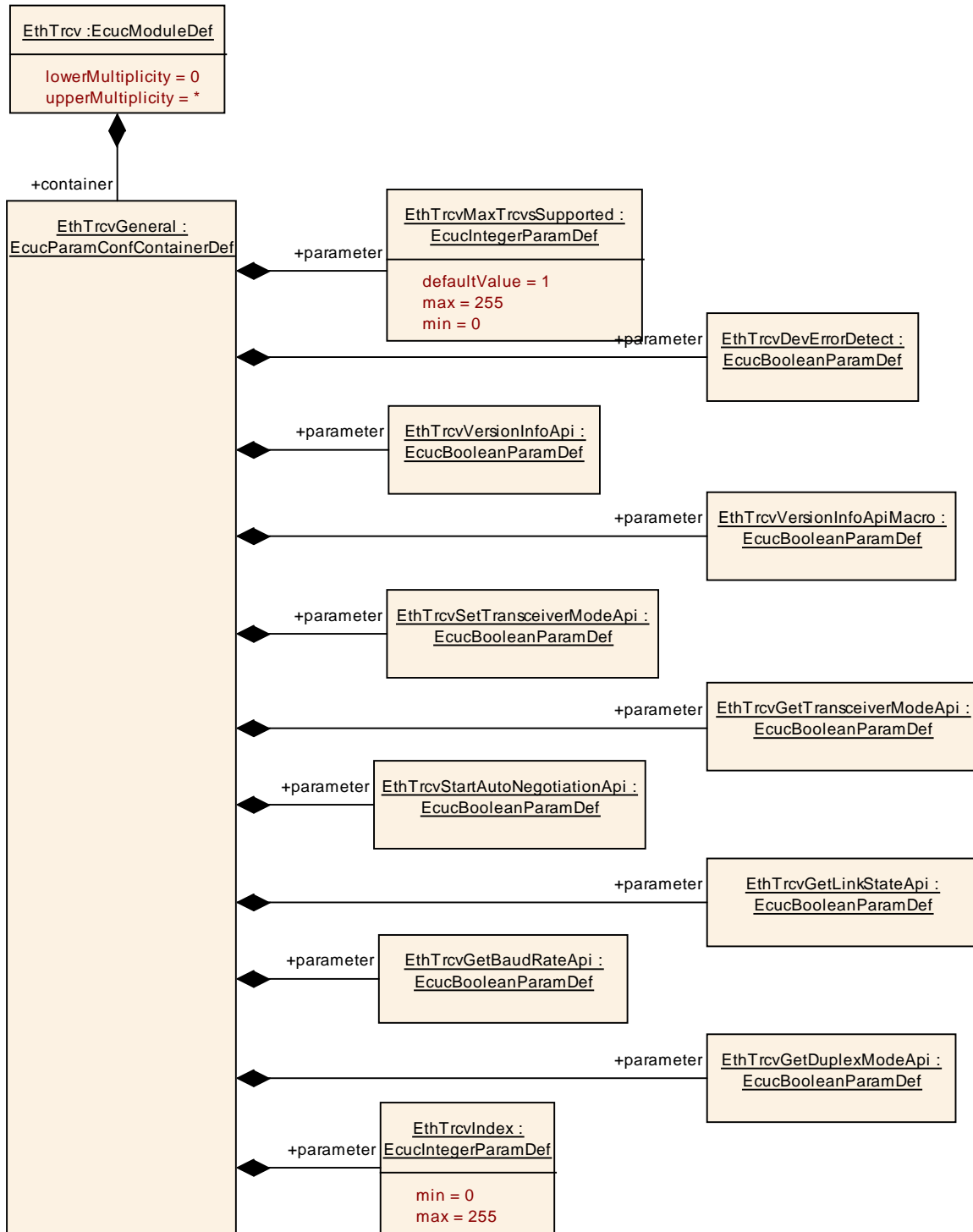


Figure 10.1: Ethernet Transceiver Driver configuration structure

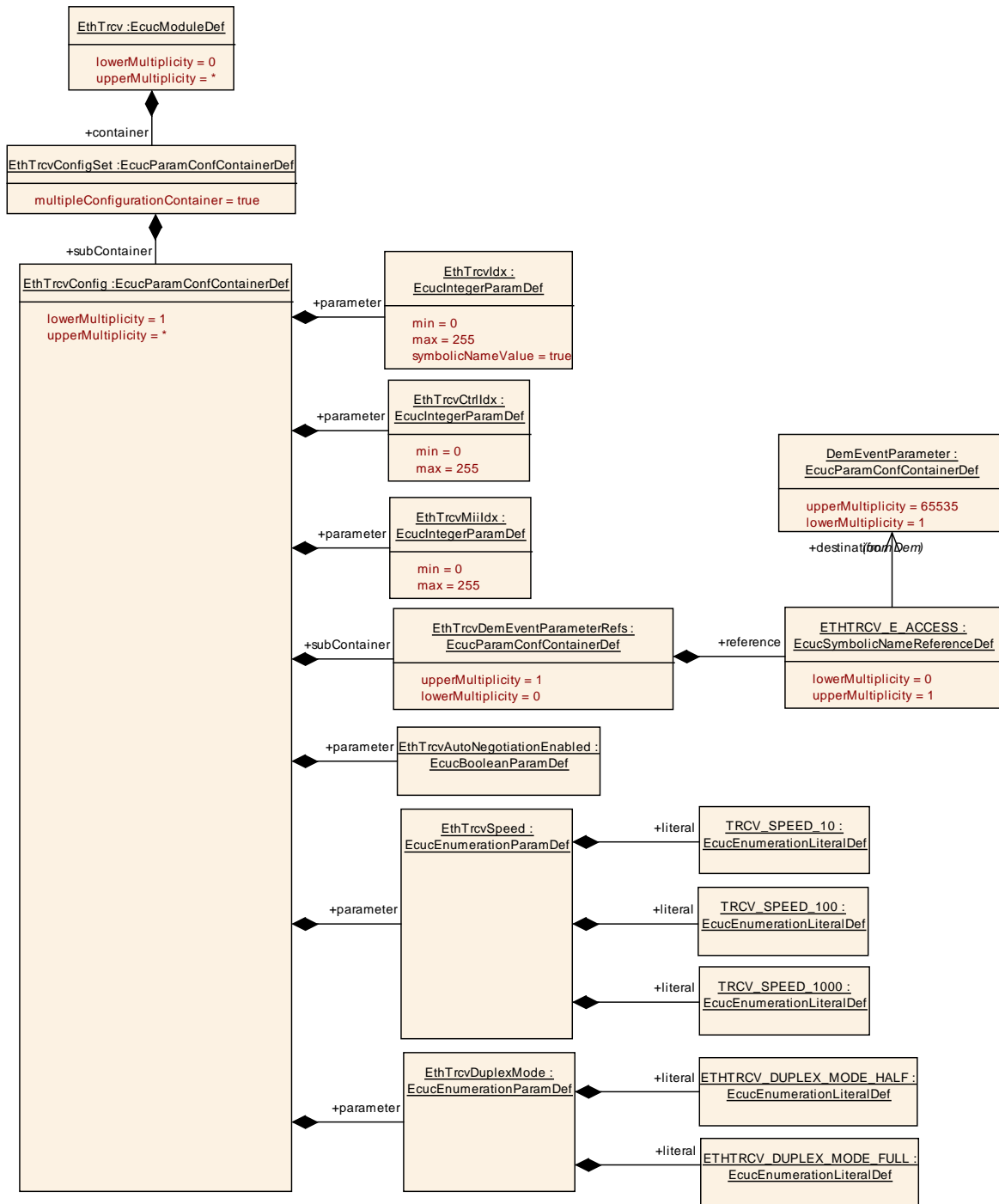


Figure 10.2: Ethernet Transceiver Driver Transceiver configuration structure

10.1.1 Variants

VARIANT-POST-BUILD: All configuration parameters in container 'EthTrcvGeneral' shall be configurable at pre-compile time.

Use case: Object code delivery, selectable configuration

VARIANT-LINK-TIME: All configuration parameters in container 'EthTrcvGeneral' shall be configurable at pre-compile time.

Use case: Object code delivery, single configuration

VARIANT-PRE-COMPILE: All configuration parameters shall be configurable at pre-compile time.

Use case: Execution time optimizations, fix configuration

10.1.2 EthTrcv

Module Name	EthTrcv
Module Description	Configuration of Ethernet Transceiver Driver module

Included Containers		
Container Name	Multiplicity	Scope / Dependency
EthTrcvConfigSet	1	All underlying parameters may be part of a multiple configuration set.
EthTrcvGeneral	1	General configuration of Ethernet Transceiver Driver module

10.1.3 EthTrcvConfigSet

SWS Item	ECUC_EthTrcv_00016 :
Container Name	EthTrcvConfigSet [Multi Config Container]
Description	All underlying parameters may be part of a multiple configuration set.
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
EthTrcvConfig	1..*	Configuration of the individual transceiver

10.1.4 EthTrcvConfig

SWS Item	ECUC_EthTrcv_00012 :
Container Name	EthTrcvConfig
Description	Configuration of the individual transceiver
Configuration Parameters	

SWS Item	ECUC_EthTrcv_00021 :		
Name	EthTrcvAutoNegotiationEnabled		
Description	Specifies if Auto-Negotiation is enabled (TRUE) or disabled (FALSE) for determination of the Ethernet transceiver speed.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	ECUC_EthTrcv_00014 :
Name	EthTrcvCtrlIdx
Description	Specifies the controller used for MII access to the transceiver

Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	ECUC_EthTrcv_00023 :		
Name	EthTrcvDuplexMode		
Description	Specifies the duplex mode of the Ethernet transceiver link if Auto-Negotiation is disabled. This parameter is ignored if Auto-Negotiation is enabled.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	ETHTRCV_DUPLEX_MODE_FULL		Full duplex.
	ETHTRCV_DUPLEX_MODE_HALF		Half duplex.
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope:		local
	dependency: EthTrcvAutoNegotiationEnabled		

SWS Item	ECUC_EthTrcv_00013 :		
Name	EthTrcvIdx		
Description	Specifies the instance ID of the configured transceiver.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 255		
Default value	--		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

SWS Item	ECUC_EthTrcv_00015 :		
Name	EthTrcvMiidx		
Description	Specifies the transceiver index used for MII access to the transceiver		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	ECUC_EthTrcv_00022 :		
Name	EthTrcvSpeed		
Description	Specifies the speed of the Ethernet transceiver link in [MBit/s]. If AutoNegotiation is enabled this is the maximum speed advertised for Auto-Negotiation.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	TRCV_SPEED_10		10 MBit/s
	TRCV_SPEED_100		100 MBit/s

	TRCV_SPEED_1000	1000 MBit/s
ConfigurationClass	Pre-compile time	X VARIANT-PRE-COMPILE
	Link time	X VARIANT-LINK-TIME
	Post-build time	X VARIANT-POST-BUILD
Scope / Dependency	scope: dependency: EthTrcvAutoNegotiationEnabled	local

Included Containers		
Container Name	Multiplicity	Scope / Dependency
EthTrcvDemEventParameterRefs	0..1	Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus API in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.

10.1.5 EthTrcvDemEventParameterRefs

SWS Item	ECUC_EthTrcv_00017 :
Container Name	EthTrcvDemEventParameterRefs
Description	Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus API in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.
Configuration Parameters	

SWS Item	ECUC_EthTrcv_00018 :	
Name	ETHTRCV_E_ACCESS	
Description	Reference to the DemEventParameter which shall be issued when the error "Transceiver access failed" has occurred.	
Multiplicity	0..1	
Type	Symbolic name reference to [DemEventParameter]	
ConfigurationClass	Pre-compile time	X VARIANT-PRE-COMPILE
	Link time	X VARIANT-LINK-TIME
	Post-build time	X VARIANT-POST-BUILD
Scope / Dependency	scope: local	

No Included Containers

10.1.6 EthTrcvGeneral

SWS Item	ECUC_EthTrcv_00001 :
Container Name	EthTrcvGeneral
Description	General configuration of Ethernet Transceiver Driver module
Configuration Parameters	

SWS Item	ECUC_EthTrcv_00003 :
Name	EthTrcvDevErrorDetect
Description	Enables / Disables development error detection
Multiplicity	1

Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	ECUC_EthTrcv_00010 :		
Name	EthTrcvGetBaudRateApi		
Description	Enables / Disables EthTrcv_GetBaudRate API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	ECUC_EthTrcv_00011 :		
Name	EthTrcvGetDuplexModeApi		
Description	Enables / Disables EthTrcv_GetDuplexMode API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	ECUC_EthTrcv_00009 :		
Name	EthTrcvGetLinkStateApi		
Description	Enables / Disables EthTrcv_GetLinkState API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	ECUC_EthTrcv_00007 :		
Name	EthTrcvGetTransceiverModeApi		
Description	Enables / Disables EthTrcv_GetTransceiverMode API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	ECUC_EthTrcv_00020 :		
Name	EthTrcvIndex		
Description	Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.		

Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	ECUC_EthTrcv_00002 :		
Name	EthTrcvMaxTrcvSupported		
Description	--		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	1		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	ECUC_EthTrcv_00006 :		
Name	EthTrcvSetTransceiverModeApi		
Description	Enables / Disables EthTrcv_SetTransceiverMode API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	ECUC_EthTrcv_00008 :		
Name	EthTrcvStartAutoNegotiationApi		
Description	Enables / Disables EthTrcv_StartAutoNegotiation API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	ECUC_EthTrcv_00004 :		
Name	EthTrcvVersionInfoApi		
Description	Enables / Disables version info API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	ECUC_EthTrcv_00005 :		
-----------------	-----------------------------	--	--

Name	EthTrcvVersionInfoApiMacro		
Description	Enables / Disables version info API macro implementation		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

11 Not applicable requirements

[SWS_EthTrcv_00999] 「 These requirements are not applicable to this specification.
」(BSW00170)