Document Title	Specification of Ethernet Transceiver Driver
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	431

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R24-11

Document Change History

Date	Release	Changed by	Description
2024-11-27	R24-11	AUTOSAR Release Management	• Support of multi-gigabite(2.5Gbit/s, 5Gbit/s, 10Gbit/s) • Multicore Support • Support for clause 45 MII register access • Change the container of EthTrcvEnablePLCA from EthTrcvGeneral to EthTrcvConfig • Editorial changes
2023-11-23	R23-11	AUTOSAR Release Management	• Missing run time error added • withAuto=TRUE for EthTrcvIdx • Editorial changes
2022-11-24	R22-11	AUTOSAR Release Management	• MACSec support • Editorial changes
2021-11-25	R21-11	AUTOSAR Release Management	• New runtime error and return code handling modified • Detailing added in buffer handling • Editorial changes

▽

△

2020-11-30	R20-11	AUTOSAR Release Management	• EthTrcvGetTransceiverModeApi and EthTrcvSetTransceiverModeApi obsolete • Eth_GeneralTypes removed from imported module list • Introduction WakeOnDataLine • Introduction 10BASE-T1S
2019-11-29	R19-11	AUTOSAR Release Management	• 2500Mbit Ethernet Support • Cable Diagnostic • Uniformatisation Eth_Mode Type • Changed Document Status from final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• Explicite transceiver link control • Support of host controllers with multiple cores
2017-12-08	4.3.1	AUTOSAR Release Management	• Minor corrections and adaptions • Improved Switch integration
2016-11-30	4.3.0	AUTOSAR Release Management	• Configuration via Switch or Mii • 100BASE-T1 test mode support
2015-07-31	4.2.2	AUTOSAR Release Management	• EthTrcv_TransceiverInit functionality merged into EthTrcv_Init API • Development Error Tracer renamed to Default Error Tracer
2014-10-31	4.2.1	AUTOSAR Release Management	• Change from Synchronous to Asynchronous API • Ethernet Wakeup Support
2013-10-31	4.1.2	AUTOSAR Release Management	• Introduction of Eth_GeneralTypes.h • Support of API deviation for asynchronous implementation • Editorial changes • Removed chapter(s) on change documentation
2013-03-15	4.1.1	AUTOSAR Administration	• 1000 kbit Ethernet Support
2011-12-22	4.0.3	AUTOSAR Administration	• EthTrcv_GetVersionInfo revised

▽

△

2010-09-30	3.1.5	AUTOSAR Administration	• 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
2010-02-02	3.1.4	AUTOSAR Administration	• Initial Release

Disclaimer

This work (specification and/or software implementation) 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 work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work 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 work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Contents

Known Limitations

Currently, chapter 5 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, see *SWS Ethernet Interface* [1]) 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.

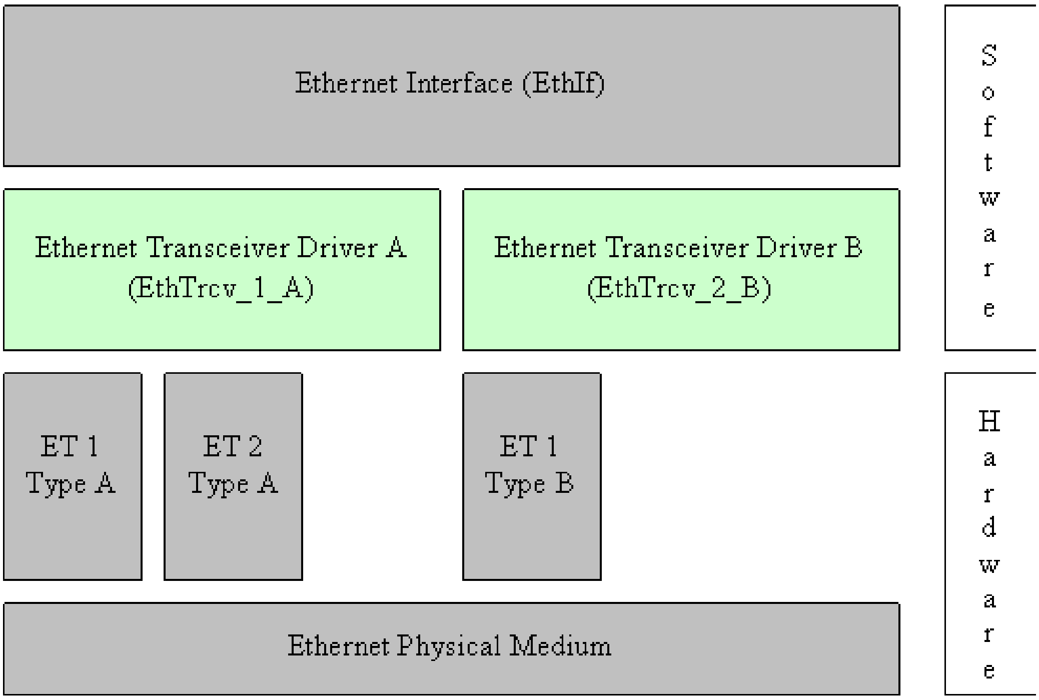
Ethernet Interface (EthIf)
Ethernet Transceiver Driver A (EthTrcv_1_A)
Ethernet Transceiver Driver B (EthTrcv_2_B)
ET 1 Type A
ET 2 Type A
ET 1 Type B
Ethernet Physical Medium
Software
Hardware

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

The glossary below includes acronyms and abbreviations relevant to the Ethernet Transceiver Driver module that are not included in the *AUTOSAR glossary* [2].

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 IEEE 802.3 -2006)
PLCA	Physical Layer Collision Avoidance
P2P	Point-to-Point
TO	Transmit Opportunity
OA TC10	Open Alliance TC10 (see [3])

3 Related documentation

3.1 Input documents

[1] Specification of Ethernet Interface
AUTOSAR_CP_SWS_EthernetInterface

[2] Glossary
AUTOSAR_FO_TR_Glossary

[3] OPEN Sleep/Wake-up Specification for Automotive Ethernet
http://www.opensig.org/Automotive-Ethernet-Specifications/

[4] General Specification of Basic Software Modules
AUTOSAR_CP_SWS_BSWGeneral

[5] Requirements on Ethernet Support in AUTOSAR
AUTOSAR_CP_RS_Ethernet

[6] Specification of Default Error Tracer
AUTOSAR_CP_SWS_DefaultErrorTracer

[7] IEEE 802.3cg-2019
https://www.ieee802.org/3/

3.2 Related standards and norms

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

[9] IEEE 802.3-2006

3.3 Related specification

AUTOSAR provides a *General Specification on Basic Software modules* [4], 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.

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)
- Ethernet Switch Driver (EthSwt) for configuring the PHY related to a port and access the PHY registers.

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.

6 Requirements Tracing

The following tables reference the requirements specified in *SRS_Ethernet* [5] and links to the fulfillment of these. Please note that if column "Satisfied by" is empty for a specific requirement this means that this requirement is not fulfilled by this document.

Requirement	Description	Satisfied by
[SRS_Eth_00039]	The Ethernet Transceiver Driver shall provide hardware configuration and initialization.	[SWS_EthTrcv_00029] [SWS_EthTrcv_00030] [SWS_EthTrcv_00035] [SWS_EthTrcv_00201] [SWS_EthTrcv_00203] [SWS_EthTrcv_00204] [SWS_EthTrcv_00205]
[SRS_Eth_00040]	The Ethernet Transceiver Driver shall provide access to the link state.	[SWS_EthTrcv_00062]
[SRS_Eth_00107]	The Ethernet Transceiver Driver shall support access to the wake up reason.	[SWS_EthTrcv_00135] [SWS_EthTrcv_00146] [SWS_EthTrcv_00185] [SWS_EthTrcv_00186] [SWS_EthTrcv_00202] [SWS_EthTrcv_91012]
[SRS_Eth_00108]	The Ethernet Transceiver Driver shall be able to wake-up an Ethernet network.	[SWS_EthTrcv_00110] [SWS_EthTrcv_00114] [SWS_EthTrcv_00115] [SWS_EthTrcv_00118] [SWS_EthTrcv_00183] [SWS_EthTrcv_00184] [SWS_EthTrcv_00185] [SWS_EthTrcv_00191] [SWS_EthTrcv_00207]
[SRS_Eth_00117]	The Ethernet Transceiver Driver shall provide access to standardized hardware features	[SWS_EthTrcv_00147] [SWS_EthTrcv_00149] [SWS_EthTrcv_91001] [SWS_EthTrcv_91002] [SWS_EthTrcv_91003] [SWS_EthTrcv_91004] [SWS_EthTrcv_91005] [SWS_EthTrcv_91006] [SWS_EthTrcv_91007] [SWS_EthTrcv_91008] [SWS_EthTrcv_91009] [SWS_EthTrcv_91010]
[SRS_Eth_00149]	The Ethernet Transceiver Driver shall provide 10BASE-T1S support	[SWS_EthTrcv_00059] [SWS_EthTrcv_00174]
[SRS_Eth_00151]	The Ethernet Transceiver Driver shall support a controlled link shutdown (sleep request)	[SWS_EthTrcv_00180] [SWS_EthTrcv_00181] [SWS_EthTrcv_00182] [SWS_EthTrcv_00193]
[SRS_Eth_00152]	The Ethernet Transceiver Driver shall be able to react on sleep request received from the network.	[SWS_EthTrcv_00192]
[SRS_Eth_00154]	The Ethernet Transceiver Driver shall provide a possibility to re-trigger a wake-up request.	[SWS_EthTrcv_00042] [SWS_EthTrcv_00184] [SWS_EthTrcv_00196]
[SRS_Eth_00155]	The Ethernet Transceiver Driver shall provide a possibility to re-trigger a sleep request.	[SWS_EthTrcv_00194] [SWS_EthTrcv_00195]
[SRS_ModeMgm_-09267]	ComM shall support communication channels which act as communication slaves without wake-up capability	[SWS_EthTrcv_00192]

Table 6.1: Requirements Tracing

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.

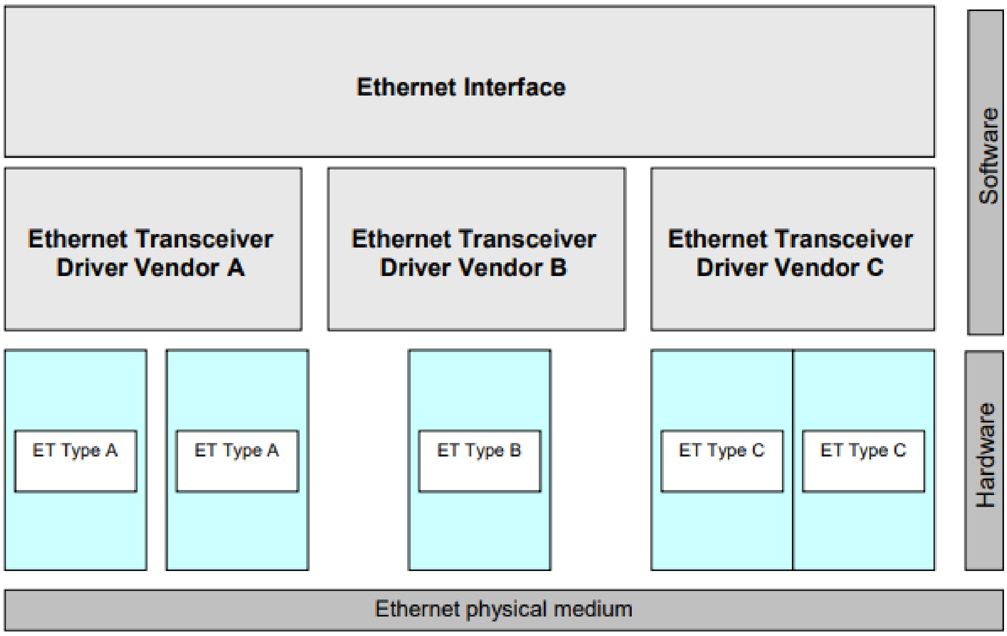
Ethernet Interface
Ethernet Transceiver Driver Vendor A
Ethernet Transceiver Driver Vendor B
Ethernet Transceiver Driver Vendor C
Software
ET Type A
ET Type A
ET Type B
ET Type C
ET Type C
Hardware
Ethernet physical medium

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.

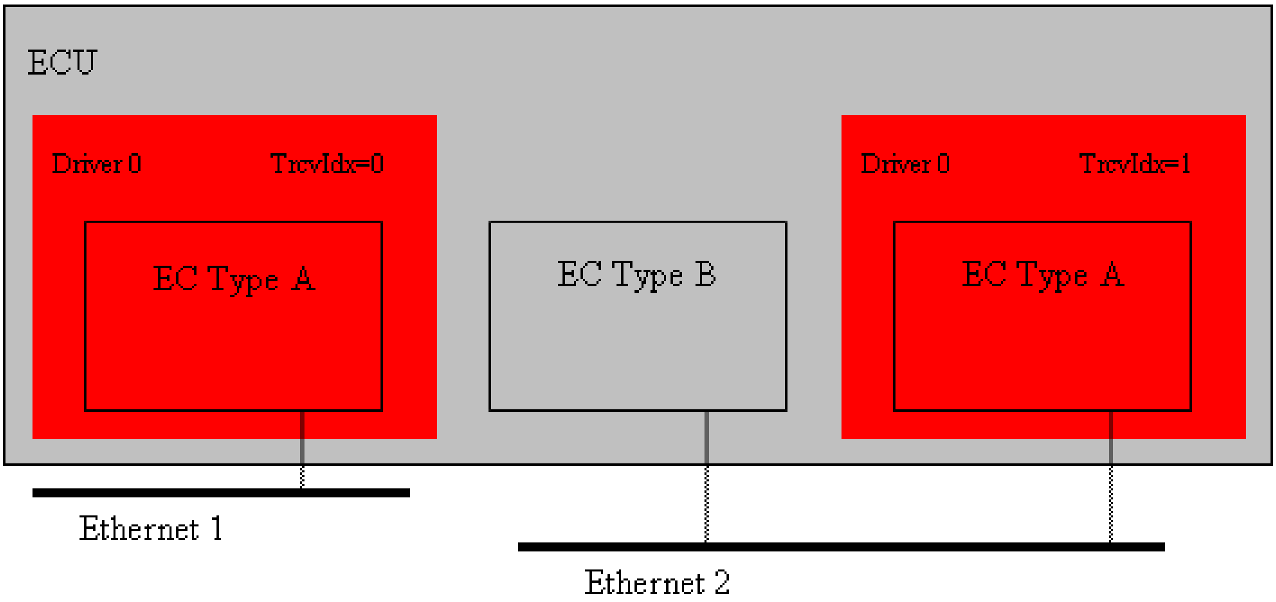

Figure 7.2: Ethernet Transceiver Driver indexing scheme

[SWS_EthTrcv_00003] ⌈The Ethernet Transceiver Driver is using a zero-based index to abstract the access for upper software layers. The parameter EthTrcvCtrlIdx [ECUC_EthTrcv_00014] within configuration corresponds to parameter TrcvIdx used in the APIs.⌋

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 *SWS Default Error Tracer* [6].

[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.⌋

Note: For more details regarding the initialization please refer to section 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.

7.1.4 Wake-up support

[SWS_EthTrcv_00110]

Upstream requirements: SRS_Eth_00108

⌈The Ethernet Transceiver driver shall support wake up detection depending on the configuration parameter EthTrcvWakeUpSupport either not at all (ETHTRCV_WAKEUP_NOT_SUPPORTED) or by interrupt (ETHTRCV_WAKEUP_BY_INTERRUPT) or by asynchronous check (ETHTRCV_WAKEUP_BY_ASYNCHRONOUS_CHECK) or by polling (ETHTRCV_WAKEUP_BY_POLLING).⌋

Note: If the Ethernet Transceiver driver detects a wakeup it will map the wake-up reason provided by the Ethernet hardware (PHY) to wake-up events defined by EcuM. The Ethernet Transceiver driver will support the following scenarios:

- Sleeping ECU and sleeping network -> wake up detection via EthTrcv_Init (called during Power On)
- Awake ECU and sleeping network -> wake up detection via EthTrcv_MainFunction or wake up interrupt handler (checked by EcuM within CheckWakeup)

[SWS_EthTrcv_00111] ⌈If the EthTrcv is requested with ETH_MODE_DOWN (e.g. low power mode), the Ethernet Transceiver driver shall enable the corresponding ICU channel (see EthTrcvIcuChannelRef) by calling Icu_EnableNotification if configured.⌋

[SWS_EthTrcv_00112] ⌈If the EthTrcv is requested with ETH_MODE_ACTIVE (e.g. normal mode), the Ethernet Transceiver driver shall disable the corresponding ICU channel (see EthTrcvIcuChannelRef) by calling Icu_DisableNotification if configured.⌋

[SWS_EthTrcv_00146]

Upstream requirements: SRS_Eth_00107

⌈The wake up interrupt handler (if present) shall clear the interrupt and identify the wake up reason and store it.⌋

7.1.4.1 Wake-up and Sleep on dataline

AUTOSAR supports OA TC10 compliant Ethernet hardware (PHY) (see [3]). Such PHYs have the capability to transmit wake-up and sleep signals on the dataline. Wake-up on dataline will bring the connected communication partners from a low-power sleep mode to a normal mode. Sleep on dataline will bring the connected communication partners from a normal mode to a low-power mode in a synchronized manner.

The Ethernet Transceiver driver is requested to trigger a wake-up on the network if the upper layer requests ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST.

The Ethernet transceiver driver is requested to trigger a sleep on the network if the upper layer requests ETH_MODE_DOWN.

A wake-up on the network could be propagated to neighboring PHYs according to the configuration.

Therefore, the configuration contains particular parameter to configure the behaviour regarding the forwarding of a received wake-up:

- Forward a locally received wake-up from a neighboring PHY to the network (EthTrcvWakeupForwardLocalEnabled);
- Forward a received wake-up from the network to the neighboring PHY. (EthTrcvWakeupForwardRemoteEnabled).

Addtionally timing parameters are available to define the timing behaviour of used Ethernet hardware (PHY):

- Define the time when a local wake-up on the used wake-up detection connection (e.g. I/O pins) is evaluated and detected as valid (EthTrcvWakeupLocalDetectionTime). Please note, local wake-up is triggered by a connected neighboring PHY;
- Define the time duration how long a wake-up is present on the used wake-up detection connection (e.g. I/O pins) for the connected neighboring PHYs (EthTrcvWakeupLocalDurationTime).

Note: If using OA TC10 compliant PHYs, then the ECU needs its own control of the power supply and should not rely on the PHY, since the INH will be switched off per specification (see [3], *Figure 1: PHY power mode sequence*)

7.1.5 PLCA support

Physical Layer Collision Avoidance (PLCA) media access control specified by IEEE (IEEE 802.3cg - 2019 [7]) allows several nodes to share the same medium.

This media access control mode is meant to be used in half-duplex and requires to be supported by the HW. This feature supported by specific HW (i.e. 10BASE-T1S) can be deactivated and in that case CSMA/CD is used for multidrop topologies. In this case, autonegociation is configured the same way as the others Ethernet PHY types.

7.1.6 Handling of cable diagnostic

Cable diagnostic measurement is triggered by calling EthTrcv_RunCableDiagnostic. The current state of the cable diagnostic measurement is polled by calling EthTrcv_GetCableDiagnosticsResult. If EthTrcv_GetCableDiagnosticsResult return with other value then ETHTRCV_CABLEDIAG_PENDING, then the cable diagnostic has finished.

It is up to the caller to re-trigger cable diagnostic again, if the measurement failed by returning ETHTRCV_CABLEDIAG_ERROR.

[SWS_EthTrcv_00159] ⌈If EthTrcv_RunPortCableDiagnostic is called, EthTrcv has to ensure that the Ethernet hardware (PHY) is in a state to run the cable diagnostic by considering at least the following points:

- the corresponding Ethernet transceiver is in state ETH_MODE_ACTIVE
- the corresponding Ethernet transceiver is in state ETHTRCV_LINK_STATE_DOWN

If all pre conditions are fullfilled to run the cable diagnostic measurement, EthTrcv shall trigger the cable diagnostic measurement and set the state internally to ETHTRCV_CABLEDIAG_PENDING of the affected Ethernet transceiver.⌋

[SWS_EthTrcv_00160] ⌈If EthTrcv_GetCableDiagnosticsResult is called, the current state of the cable diagnostic measurement of the affected Ethernet transceiver shall be returned and stored per Ethernet transceiver as internal cable diagnostic state.⌋

[SWS_EthTrcv_00161] ⌈As long as the cable diagnostic measurement is running (internal cable diagnostic state is ETHTRCV_CABLEDIAG_PENDING), a mode request (indicated by EthTrcv_SetTransceiverMode) and link request (indicated by EthTrcv_TransceiverLinkStateRequest), respectively, shall be stored and not executed.⌋

[SWS_EthTrcv_00162] ⌈As soon as the cable diagnostic meassurement has finished (internal cable diagnostic state is different from ETHTRCV_CABLEDIAG_PENDING), EthTrcv shall execute the last mode request and link request, respectively, of the affected Ethernet transceiver.⌋

Note: Cable diagnostic measurement is triggered by a CDD that maintain the cable diagnostic result. The CDD should use the dedicated APIs of EthIf to execute the cable diagnostic measurement:

- EthIf_RunCableDiagnostic: For a single Ethernet transceiver which is not referenced by an Ethernet switch port;
- EthIf_RunPortCableDiagnostic: For an Ethernet transceiver which is referenced by an Ethernet switch port.

Thus, the upper layer of the EthTrcv is either EthIf or an Ethernet switch.

7.2 Error Classification

Section 7.2 "Error Handling" of the document *General Specification of Basic Software Modules* [4] describes the error handling of the Basic Software in detail. Above all, it constitutes a classification scheme consisting of five error types which may occur in BSW modules.

Based on this foundation, the following section specifies particular errors arranged in the respective subsections below.

7.2.1 Development Errors

[SWS_EthTrcv_00017] Definiton of development errors in module EthTrcv ⌈

Type of error	*Related error code*	*Error value*
Invalid transceiver index	ETHTRCV_E_INV_TRCV_IDX	0x01
EthTrcv module was not initialized	ETHTRCV_E_UNINIT	0x02
Invalid pointer in parameter list	ETHTRCV_E_PARAM_POINTER	0x03

⌋

7.2.2 Runtime Errors

[SWS_EthTrcv_91015] Definiton of runtime errors in module EthTrcv ⌈

Type of error	*Related error code*	*Error value*
Failure or incorrect communication mode	ETHTRCV_E_MDI_COMMUNICATION	0x04

⌋

7.2.3 Production Errors

There are no production errors.

7.2.4 Extended Production Errors

Extended production errors are handled as events of the Diagnostic Event Manager. The event IDs are defined in the following tables, while the actual values are assigned externally by the configuration of the Diagnostic Event Manager, and are included in the module via Dem.h.

[SWS_EthTrcv_00105] ⌈

Error Name:	ETHTRCV_E_ACCESS	
Short Description:	Ethernet Transceiver Access Failure.	
Long Description:	Monitors the access to the Ethernet Transceiver.	
Detection Criteria:	Fail	When access to the Ethernet Transceiver fails the module shall report the extended production error with event status DEM_EVENT_STATUS_PREFAILED to DEM.
	Pass	When access to the Ethernet Transceiver succeds the module shall report the extended production error with event status DEM_EVENT_STATUS_PREPASSED to DEM.
Secondary Parameters:	None.	
Time Required:	None.	
Monitor Frequency	None.	

⌋

8 API specification

8.1 Imported types

This chapter lists all types included from the following modules:

[SWS_EthTrcv_00027] Definition of imported datatypes of module EthTrcv ⌈

Module	*Header File*	*Imported Type*
Dem	Rte_Dem_Type.h	Dem_EventIdType
	Rte_Dem_Type.h	Dem_EventStatusType
EcuM	EcuM.h	EcuM_WakeupSourceType
Eth	Eth_GeneralTypes.h	Eth_ModeType
Icu	Icu.h	Icu_ChannelType
Mka	Mka.h	Mka_ConfidentialityOffsetType (draft)
	Mka.h	Mka_MacSecConfigType (draft)
	Mka.h	Mka_SakKeyPtrType (draft)
	Mka.h	Mka_Stats_Rx_ScType (draft)
	Mka.h	Mka_Stats_Rx_SecYType (draft)
	Mka.h	Mka_Stats_SecYType (draft)
	Mka.h	Mka_Stats_Tx_ScType (draft)
	Mka.h	Mka_Stats_Tx_SecYType (draft)
	Mka.h	Mka_ValidateFramesType (draft)
Std	Std_Types.h	Std_ReturnType
	Std_Types.h	Std_VersionInfoType

⌋

8.2 Type definitions

8.2.1 EthTrcv_ConfigType

[SWS_EthTrcv_00098] Definition of datatype EthTrcv_ConfigType ⌈

Name	EthTrcv_ConfigType
Kind	Structure
Description	Implementation specific structure of the post build configuration
Available via	EthTrcv.h

⌋

8.2.2 EthTrcv_LinkStateType

[SWS_EthTrcv_00100] Definition of datatype EthTrcv_LinkStateType ⌈

Name	EthTrcv_LinkStateType		
Kind	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)		
Available via	Eth_GeneralTypes.h		

⌋

8.2.3 EthTrcv_StateType

[SWS_EthTrcv_00101] Definition of datatype EthTrcv_StateType ⌈

Name	EthTrcv_StateType		
Kind	Enumeration		
Range	ETHTRCV_STATE_UNINIT	0x00	Driver is not yet configured
	ETHTRCV_STATE_INIT	0x01	Driver is configured
Description	Status supervision used for Development Error Detection. The state shall be available for debugging.		
Available via	Eth_GeneralTypes.h		

⌋

8.2.4 EthTrcv_BaudRateType

[SWS_EthTrcv_00102] Definition of datatype EthTrcv_BaudRateType ⌈

Name	EthTrcv_BaudRateType		
Kind	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
	ETHTRCV_BAUD_RATE_2500MBIT	0x03	2500MBIT 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.		
Available via	Eth_GeneralTypes.h		

⌋

8.2.5 EthTrcv_DuplexModeType

[SWS_EthTrcv_00103] Definition of datatype EthTrcv_DuplexModeType ⌈

Name	EthTrcv_DuplexModeType		
Kind	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.		
Available via	Eth_GeneralTypes.h		

⌋

8.2.6 EthTrcv_ WakeupModeType

[SWS_EthTrcv_00114] Definition of datatype EthTrcv_WakeupReasonType

Upstream requirements: SRS_Eth_00108

⌈

Name	EthTrcv_WakeupReasonType		
Kind	Enumeration		
Range	ETHTRCV_WUR_NONE	0x00	No wake up reason detected.
	ETHTRCV_WUR_GENERAL	0x01	General wake up detected, no distinct reason supported by hardware.
	ETHTRCV_WUR_INTERNAL	0x03	Internal wake up detected. Available if supported by hardware.
	ETHTRCV_WUR_RESET	0x04	Reset wake up detected. Available if supported by hardware.

▽

△

	ETHTRCV_WUR_POWER_ON	0x05	Power on wake up detected. Available if supported by hardware.
	ETHTRCV_WUR_PIN	0x06	Pin wake up detected. Available if supported by hardware.
	ETHTRCV_WUR_SYSERR	0x07	System error wake up detected. Available if supported by hardware.
	ETHTRCV_WUR_WODL_WUP	0x08	Wake-up on data line (WUP = wake up pulse) detected while link is down of the corresponding Ethernet hardware (e.g. PHY). Only available if EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE.
	ETHTRCV_WUR_WODL_WUR	0x09	Wake-up on data line (WUR = wake up request) detected while link is active of the used Ethernet hardware (e.g. PHY). Only available if EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE.
	ETHTRCV_WUR_TRANSFER	0xA	Device internal transfer of wake-up on data line from a neighboring PHY. Only possible if Ethernet hardware is compliant to OA TC10. Only available if EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE.
Description	This type defines the transceiver wake up reasons.		
Available via	Eth_GeneralTypes.h		

⌋

8.2.7 EthTrcv_ PhyTestModeType

[SWS_EthTrcv_91002] Definition of datatype EthTrcv_PhyTestModeType

Upstream requirements: SRS_Eth_00117

⌈

Name	EthTrcv_PhyTestModeType		
Kind	Enumeration		
Range	ETHTRCV_PHYTESTMODE_NONE	0x00	normal operation
	ETHTRCV_PHYTESTMODE_1	0x01	test transmitter droop
	ETHTRCV_PHYTESTMODE_2	0x02	test master timing jitter
	ETHTRCV_PHYTESTMODE_3	0x03	test slave timing jitter
	ETHTRCV_PHYTESTMODE_4	0x04	test transmitter distortion
	ETHTRCV_PHYTESTMODE_5	0x05	test power spectral density (PSD) mask
Description	Describes the possible PHY test modes		
Available via	Eth_GeneralTypes.h		

⌋

8.2.8 EthTrcv_ PhyLoopbackModeType

[SWS_EthTrcv_91004] Definition of datatype EthTrcv_PhyLoopbackModeType

Upstream requirements: SRS_Eth_00117

⌈

Name	EthTrcv_PhyLoopbackModeType		
Kind	Enumeration		
Range	ETHTRCV_ PHYLOOPBACK_NONE	0x00	normal operation
	ETHTRCV_ PHYLOOPBACK_ INTERNAL	0x01	internal loopback
	ETHTRCV_ PHYLOOPBACK_ EXTERNAL	0x02	external loopback
	ETHTRCV_ PHYLOOPBACK_REMOTE	0x03	remote loopback
Description	Describes the possible PHY loopback modes		
Available via	Eth_GeneralTypes.h		

⌋

8.2.9 EthTrcv_ PhyTxModeType

[SWS_EthTrcv_91006] Definition of datatype EthTrcv_PhyTxModeType

Upstream requirements: SRS_Eth_00117

⌈

Name	EthTrcv_PhyTxModeType		
Kind	Enumeration		
Range	ETHTRCV_PHYTXMODE_ NORMAL	0x00	normal operation
	ETHTRCV_PHYTXMODE_ TX_OFF	0x01	transmitter disabled
	ETHTRCV_PHYTXMODE_ SCRAMBLER_OFF	0x02	scrambler disabled
Description	Describes the possible PHY transmit modes		
Available via	Eth_GeneralTypes.h		

⌋

8.2.10 EthTrcv_ CableDiagResultType

[SWS_EthTrcv_91008] Definition of datatype EthTrcv_CableDiagResultType

Upstream requirements: SRS_Eth_00117

⌈

Name	EthTrcv_CableDiagResultType		
Kind	Enumeration		
Range	ETHTRCV_CABLEDIAG_OK	0x00	Cable diagnostic ok
	ETHTRCV_CABLEDIAG_ERROR	0x01	Cable diagnostic failed
	ETHTRCV_CABLEDIAG_SHORT	0x02	Short circuit detected
	ETHTRCV_CABLEDIAG_OPEN	0x03	Open circuit detected
	ETHTRCV_CABLEDIAG_PENDING	0x04	cable diagnostic is still running
	ETHTRCV_CABLEDIAG_WRONG_POLARITY	0x05	cable diagnostics has detected wrong polarity of the "Ethernet physical+" or "Ethernet physical-" lines
Description	Describes the results of the cable diagnostics.		
Available via	Eth_GeneralTypes.h		

⌋

8.2.11 EthTrcv_MacMethodType

[SWS_EthTrcv_91013] Definition of datatype EthTrcv_MacMethodType

Status: DRAFT

⌈

Name	EthTrcv_MacMethodType (draft)		
Kind	Enumeration		
Range	ETHTRCV_MAC_TYPE_CSMA_CD	0x00	Carrier sence multiple access with collision detection
	ETHTRCV_MAC_TYPE_PLCA	0x01	Physical layer collision avoidance
Description	This type defines the media access control type in half duplex. If ETHTRCV_MAC_TYPE_PLCA is configured, PLCA is supported and the parameters listed in ECUC_EthTrcv_00055 have to be configured. **Tags:** atp.Status=draft		
Available via	Eth_GeneralTypes.h		

⌋

8.3 Function definitions

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

8.3.1 EthTrcv_Init

[SWS_EthTrcv_00028] Definition of API function EthTrcv_Init ⌈

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	
Available via	EthTrcv.h	

⌋

[SWS_EthTrcv_00029]

Status: DRAFT
Upstream requirements: SRS_Eth_00039

⌈The function shall store the access to the configuration structure for subsequent API calls.⌋

[SWS_EthTrcv_00203]

Upstream requirements: SRS_Eth_00039

⌈The function shall consider at least the following initialization sequence for the Ethernet hardware (PHY):

- check for wake-up reasons according to [SWS_EthTrcv_00115]
- initialize the PHY according to [SWS_EthTrcv_00035]
- ensure that the PHY resides in a defined hardware state according to [SWS_EthTrcv_00204] and [SWS_EthTrcv_00205].

⌋

[SWS_EthTrcv_00115]

Upstream requirements: SRS_Eth_00108

⌈The function shall check for wake-up reasons and propagate the corresponding wake-up source (see EthTrcvWakeupMap configuration [ECUC_EthTrcv_00027]) to the EcuM by calling EcuM_SetWakeupEvent.⌋

[SWS_EthTrcv_00204]

Upstream requirements: SRS_Eth_00039

⌈If EthTrcvWakeupSleepOnDatalineEnabled is not configured, then the Ethernet Transceiver driver shall ensure within the function call of EthTrcv_Init that the Ethernet hardware (PHY) resides in a sleep mode.⌋

[SWS_EthTrcv_00035]

Upstream requirements: SRS_Eth_00039

⌈The function shall initialize the Ethernet hardware (PHY) with all available configuration parameters:

- EthTrcvConnNeg (see [ECUC_EthTrcv_00025])
- EthTrcvDuplexMode (see [ECUC_EthTrcv_00023])
- EthTrcvMacLayerSpeed (see [ECUC_EthTrcv_00044])
- EthTrcvMacLayerSubType (see [ECUC_EthTrcv_00043])
- EthTrcvMacLayerType (see [ECUC_EthTrcv_00035])
- EthTrcvPhysLayerType (see [ECUC_EthTrcv_00024])
- EthTrcvSpeed (see [ECUC_EthTrcv_00022]).

⌋

Note: Only available configuration parameter of Ethernet hardware (PHY) are considered by the Ethernet Transceiver driver initialization. The remaining configuration parameter may be hardware-configured (e.g. pin strapping).

[SWS_EthTrcv_00030]

Upstream requirements: SRS_Eth_00039

⌈The function shall change the state of the component from ETHTRCV_STATE_UNINIT to ETHTRCV_STATE_INIT.⌋

[SWS_EthTrcv_00040] ⌈The function shall check the access to the Ethernet transceiver. If the check fails, the function shall raise the production error ETHTRCV_E_ACCESS otherwise pass the production error ETHTRCV_E_ACCESS.⌋

[SWS_EthTrcv_00032] ⌈*EthTrcv_Init()* shall be called during initialization.⌋

8.3.1.1 Initialization of OA TC10 PHYs

[SWS_EthTrcv_00205]

Upstream requirements: SRS_Eth_00039

⌈If EthTrcvWakeupSleepOnDatalineEnabled is configured and no wake-up reason was detected, then the Ethernet Transceiver driver shall ensure within the call of EthTrcv_Init that the Ethernet hardware (PHY) resides in the OA TC10 [3] specified sleep mode⌋

[SWS_EthTrcv_00201]

Upstream requirements: SRS_Eth_00039

⌈If EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE and a wake-up reason was detected which relates to OA TC10 [3] compatible Ethernet hardware (PHY), then the Ethernet hardware state (e.g. normal mode) shall be kept, since a wake-up arrived and the corresponding delay timer with EthTrcvSleepModeExecutionDelay shall be started.⌋

Note:

- OA TC10 related wake-up reasons are ETHTRCV_WUR_WODL_WUP, ETHTRCV_WUR_WODL_WUR and ETHTRCV_WUR_TRANSFER.
- If a wake-up was detected, the propagation of the wake-up to the upperlayers may need some processing time. A timer to delay a sleep execution is used to prevent the Ethernet hardware (PHY) to immediately execute a sleep request while the forwarding of the wake-up to all affected upperlayers (EcuM, EthSM, EthIf) has not finished yet.

8.3.2 EthTrcv_SetTransceiverMode

[SWS_EthTrcv_00042] Definition of API function EthTrcv_SetTransceiverMode

Upstream requirements: SRS_Eth_00154

⌈

Service Name	EthTrcv_SetTransceiverMode	

Syntax	`Std_ReturnType EthTrcv_SetTransceiverMode (`
	`  uint8 TrcvIdx,`
	`  Eth_ModeType TrcvMode`
	`)`	

Service ID [hex]	0x03	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	TrcvIdx	Index of the transceiver within the context of the Transceiver Driver
	TrcvMode	ETH_MODE_DOWN: disable the transceiver ETH_MODE_ACTIVE: enable the transceiver ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST: enable the transceiver and request to trigger a wake-up on the network, if the used PHY support such a feature. E.g. used for PHYs compliant to OA TC10
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Service accepted `E_NOT_OK`: Service denied
Description	Enables / disables the indexed transceiver	
Available via	EthTrcv.h	

⌋

[SWS_EthTrcv_00163] ⌈If EthTrcv_SetTransceiverMode() is called and the internal cable diagnostic state is equal to ETHTRCV_CABLEDIAG_PENDING, the Ethernet Transceiver driver shall store the mode request per EthTrcv (TrcvIdx).⌋

Note: Furhter processing of the mode will be done when cable diagnostic has finished, see SWS_EthTrcv_00162.

[SWS_EthTrcv_00043] ⌈The function shall put the indexed transceiver in the specified mode according to the following specified requirements. Each time a new mode is entered which was previously requested by EthTrcv_SetTransceiverMode, EthTrcv shall call EthIf_TrcvModeIndication latest during the next EthTrcv_MainFunction.⌋

[SWS_EthTrcv_00179] ⌈A new requested EthTrcv mode shall overwrite the last requested EthTrcv mode, except in case the last mode is ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST and the new mode is ETH_MODE_ACTIVE.⌋

Note: The Ethernet Transceiver driver store the requested mode per EthTrcv. The stored mode is used to re-trigger the requested mode in the context of the main function:

- A stored wake-up is repeated according to the available configuration (see EthTrcvWakeupRequestNumberOfRepetitions and EthTrcvWakeUpRequestRepetitionPeriod).
- A stored sleep request is repeated according the available configuration (EthTrcvSleepRequestNumberOfRepetitions and EthTrcvSleepRequestRepetitionPeriod).

8.3.2.1 ETH_MODE_DOWN request

[SWS_EthTrcv_00117] ⌈If the function is called with ETH_MODE_DOWN and EthTrcvWakeupSleepOnDatalineEnabled is not configured, it shall set the corresponding Ethernet Hardware (PHY) into a mode (e.g. sleep mode) where wakeups can be detected.⌋

[SWS_EthTrcv_00206] ⌈If the function is called with ETH_MODE_DOWN and EthTrcvWakeupSleepOnDatalineEnabled is set to FALSE, the Ethernet Transceiver shall set the corresponding Ethernet Hardware (PHY) into a sleep mode without performing the sleep process according to OA TC10 [3].⌋

[SWS_EthTrcv_00180]

Upstream requirements: SRS_Eth_00151

⌈If the function is called with ETH_MODE_DOWN, EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE, the Ethernet Transceiver driver has detected a pending wake-up and a corresponding delay timer with EthTrcvSleepModeExecutionDelay is not running, the Ethernet Transceiver driver shall start a delay timer with EthTrcvSleepModeExecutionDelay and return with E_OK.⌋

Note: The further processing of the ETH_MODE_DOWN is done in the main function, see [SWS_EthTrcv_00193]

Rational for SWS_EthTrcv_00180: The delay of the execution for a sleep (ETH_MODE_DOWN) should avoid the race condition, if a ETH_MODE_DOWN was requested while a wake-up of a neighboring PHY was received via a local wake-up connection (e.g. I/O pin). The EthSM shall be able to finish the shutdown process (transition from ETHSM_STATE_WAIT_OFFLINE to ETHSM_STATE_OFFLINE), but the EthTrcv should delay the sleep processing, because it could be that EthSM request the network with COMM_FULL_COMMUNICATION again, due the pending wake-up, received via a local wake-up connection. An immediate execution of a sleep could bring the Ethernet hardware (PHY) to sleep, while the local wake-up request needs the Ethernet hardware (PHY) in normal state. A local received wake-up is handled as a passive communication request and will not bring the Ethernet hardware back to normal mode.

[SWS_EthTrcv_00181]

Upstream requirements: SRS_Eth_00151

⌈If the function is called with ETH_MODE_DOWN, EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE and the corresponding delay timer with EthTrcvSleepModeExecutionDelay is running, then the Ethernet Transceiver driver shall return with E_OK.⌋

Note: A running delay timer with EthTrcvSleepModeExecutionDelay inidicate that a sleep (ETH_MODE_DOWN) was already requested. Therefor no further handling for the sleep request is needed.

[SWS_EthTrcv_00182]

Upstream requirements: SRS_Eth_00151

⌈If the function is called with ETH_MODE_DOWN, EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE and the EthTrcv has not detected a pending wake-up (see SWS_EthTrcv_00180), the Ethernet hardware (PHY) is in another mode than sleep mode and has not already triggered a sleep request, the Ethernet Transceiver driver shall trigger a sleep request according to the specification of OA TC10 (see [3]).⌋

Note: The Ethernet Transceiver driver is checking if the Ethernet hardware (PHY) has already entered the sleep mode to avoid to trigger the affected Ethernet hardware (PHY) again with a sleep request. This is needed, because the connected Ethernet ECUs trigger a sleep request may be in a different point in time (e.g. if using Ethernet switch port switching the executing of sleep request is delayed by EthIfSwitchOffPortTimeDelay). The Ethernet ECU which shutdown a bit earlier trigger the sleep request and bring the own Ethernet hardware (PHY) and the Ethernet hardware of the connected Ethernet ECUs to sleep mode. The Ethernet Transceiver driver of the ECU which shutdown a bit later detect that the affected Ethernet hardware (PHY) has already reached sleep mode. In this case, the Ethernet hardware (PHY) shall be left as it is and return from the function call.

8.3.2.2 ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST or ETH_MODE_ACTIVE request

[SWS_EthTrcv_00118]

Upstream requirements: SRS_Eth_00108

⌈If EthTrcv_SetTransceiverMode() is called with parameter ETH_MODE_ACTIVE or ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST, EthTrcvWakeupSleepOnDatalineEnabled is not configured and the internal cable diagnostic state is different from ETHTRCV_CABLEDIAG_PENDING, the Ethernet Transceiver driver shall:

- Check for wake-up reasons when entering the transceiver's active mode;
- If no wake-up reason has been detected, the Ethernet transceiver shall send a wake-up symbol on the bus if configured;
- Invoke the call-out <*EthTrcvWakeUpCallout*> function if configured.

⌋

[SWS_EthTrcv_00207]

Upstream requirements: SRS_Eth_00108

⌈If EthTrcv_SetTransceiverMode() is called with parameter ETH_MODE_ACTIVE or ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST, EthTrcvWakeupSleepOnDatalineEnabled set to FALSE and the internal cable diagnostic state is different from ETHTRCV_CABLEDIAG_PENDING, the Ethernet Transceiver driver shall:

- Put the Ethernet hardware (PHY) to normal mode,
- Invoke the call-out <*EthTrcvWakeUpCallout*> function if configured.

⌋

[SWS_EthTrcv_00183]

Upstream requirements: SRS_Eth_00108

⌈If the function is called with ETH_MODE_ACTIVE,

EthTrcvWakeupSleepOnDatalineEnabled is TRUE and the internal cable diagnostic state is different from ETHTRCV_CABLEDIAG_PENDING, the Ethernet Transceiver driver shall:

- Put the Ethernet hardware (PHY) to normal mode, if not already switched automatically by the statemachine of the Ethernet hardware (PHY),
- Invoke the call-out <*EthTrcvWakeUpCallout*> function if configured.

⌋

Note: A requested mode with ETH_MODE_ACTIVE indicate a passive wake-up. Thus, the Ethernet hardware was remotely woken up by the connected communication partner. In this case no wake-up shall be transmitted on the network. The Ethernet hardware (PHY) should be put to normal mode, due to a received wake-up according to the OA TC10 (see [3]). But to increase the robustness for the communication, the Ethernet Transciever Driver shall check the current mode and bring the Ethernet hardware (PHY) to normal mode, if not done by the statemachine Ethernet hardware (PHY).

[SWS_EthTrcv_00184]

Upstream requirements: SRS_Eth_00108, SRS_Eth_00154

⌈If the function is called with ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST,

EthTrcvWakeupSleepOnDatalineEnabled is TRUE, the internal cable diagnostic state is different from ETHTRCV_CABLEDIAG_PENDING and EthTrcv does not currently perform wakeup repetition algorithm (see [SWS_EthTrcv_00194]), the Ethernet Transceiver driver shall:

- trigger the Ethernet hardware (PHY) to transmit a wake-up on the network according to the specification of OA TC10 (see [3]),
- invoke the call-out <*EthTrcvWakeUpCallout*> function if configured.

⌋

Note: A requested mode with ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST indicate an active wake-up. This would result in a trigger of a wake-up on the network and putting the Ethernet hardware (PHY) implicitely to normal mode.

Comment on [SWS_EthTrcv_00183] and [SWS_EthTrcv_00184]: In case of former requested mode was ETH_MODE_DOWN and new mode is ETH_MODE_ACTIVE or ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST all actions regarding sleep handling will no longer be processed (refer to [SWS_EthTrcv_00193], [SWS_EthTrcv_00194], [SWS_EthTrcv_00195]).

Independent of the former mode when ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST is requested also the number of wakeup repetions will start from beginning (refer to [SWS_EthTrcv_00196]).

8.3.2.3 Error handling

[SWS_EthTrcv_00044] ⌈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_UNINIT.⌋

[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.⌋

[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_00104] ⌈EthTrcv_SetTransceiverMode() shall check the access to the Ethernet transceiver. If the check fails, the function shall raise the production error ETHTRCV_E_ACCESS and return E_NOT_OK, otherwise pass the production error ETHTRCV_E_ACCESS and return E_OK.⌋

[SWS_EthTrcv_00047] ⌈*EthTrcv_Init()* shall be called before *EthTrcv_SetTransceiverMode()*.⌋

8.3.3 EthTrcv_GetTransceiverMode

[SWS_EthTrcv_00048] Definition of API function EthTrcv_GetTransceiverMode ⌈

Service Name	EthTrcv_GetTransceiverMode	

Syntax	`Std_ReturnType EthTrcv_GetTransceiverMode (`
	`  uint8 TrcvIdx,`
	`  Eth_ModeType* TrcvModePtr`
	`)`	

Service ID [hex]	0x04	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	TrcvIdx	Index of the transceiver within the context of the Transceiver Driver
Parameters (inout)	None	
Parameters (out)	TrcvModePtr	ETH_MODE_DOWN: the transceiver is disabled ETH_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	
Available via	EthTrcv.h	

⌋

[SWS_EthTrcv_00049] ⌈The function shall read the current transceiver mode of the Ethernet hardware (PHY), whereby ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST shall always be returned as ETH_MODE_ACTIVE.⌋

[SWS_EthTrcv_00050] ⌈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_UNINIT.⌋

[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.⌋

[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_PARAM_POINTER.⌋

[SWS_EthTrcv_00054] ⌈*EthTrcv_Init()* shall be called before *EthTrcv_GetTransceiverMode()*.⌋

8.3.4 EthTrcv_GetBusWuReason

[SWS_EthTrcv_91012] Definition of API function EthTrcv_GetBusWuReason

Upstream requirements: SRS_Eth_00107

⌈

Service Name	EthTrcv_GetBusWuReason	

Syntax	`Std_ReturnType EthTrcv_GetBusWuReason (`
	`  uint8 TrcvIdx,`
	`  EthTrcv_WakeupReasonType* WakeupReasonPtr`
	`)`	

Service ID [hex]	0x17	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
Parameters (inout)	None	
Parameters (out)	WakeupReasonPtr	Pointer to structure of least recent wakeup event, which was detected by the Ethernet PHY
Return value	Std_ReturnType	`E_OK`: PHY wake up reason request has been accepted. `E_NOT_OK`: PHY wake up reason request has not been accepted.
Description	This function returns the least recent wakeup reasons.	
Available via	EthTrcv.h	

⌋

[SWS_EthTrcv_00186]

Upstream requirements: SRS_Eth_00107

⌈The function EthTrcv_GetBusWuReason shall read the stored wake-up reason and provide the information in WakeupReasonPtr.⌋

[SWS_EthTrcv_00187] ⌈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_UNINIT.⌋

[SWS_EthTrcv_00188] ⌈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.⌋

[SWS_EthTrcv_00189] ⌈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_PARAM_POINTER.⌋

[SWS_EthTrcv_00190] ⌈The function shall be pre compile time configurable On/Off by the configuration parameter: EthTrcvGetBusWuReasonApi [ECUC_EthTrcv_00061].⌋

8.3.5 EthTrcv_CheckWakeup

[SWS_EthTrcv_00134] Definition of API function EthTrcv_CheckWakeup ⌈

Service Name	EthTrcv_CheckWakeup	

Syntax	`Std_ReturnType EthTrcv_CheckWakeup (`
	`  uint8 TrcvIdx`
	`)`	

Service ID [hex]	0x0f	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	TrcvIdx	Index of the transceiver within the context of the Transceiver Driver
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The function has been successfully executed `E_NOT_OK`: The function could not be successfully executed
Description	Service is called by EthIf in case a wake-up interrupt is detected.	
Available via	EthTrcv.h	

⌋

[SWS_EthTrcv_00135]

Upstream requirements: SRS_Eth_00107

⌈If EthTrcvWakeUpSupport is set to ETHTRCV_WAKEUP_BY_INTERRUPT or ETHTRCV_WAKEUP_BY_POLLING, the function EthTrcv_CheckWakeup() shall check if a wake up has been detected and if yes propagate the corresponding wake up source (see EthTrcvWakeupMap configuration [ECUC_EthTrcv_00027]) to the EcuM by calling EcuM_SetWakeupEvent.⌋

[SWS_EthTrcv_00202]

Upstream requirements: SRS_Eth_00107

⌈If the function EthTrcv_CheckWakeup() is called, EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE, a wake-up reason which relates to OA TC10 [3] compliant Ethernet hardware (PHY) was detected (ETHTRCV_WUR_WODL_WUP, ETHTRCV_WUR_WODL_WUR or ETHTRCV_WUR_TRANSFER) and EthTrcvWakeUpSupport is to ETHTRCV_WAKEUP_BY_ASYNCHRONOUS_CHECK, ETHTRCV_WAKEUP_BY_INTERRUPT or ETHTRCV_WAKEUP_BY_POLLING, then the Ethernet Transceiver driver shall start a delay timer with EthTrcvSleepModeExecutionDelay.⌋

Note:

- A wake-up has to be propagated to the affected upper layer (EcuM, EthSM, EthIf). As long as EthIf is not aware of the detected wake-up, EthIf attempts to ensure that the requested communication mode matches to the current Ethernet hardware (PHY, Ethernet port) state. In case a wake-up for OA TC10 compliant Ethernet hardware (PHY) arrives, the PHY transits to a normal mode (ETH_MODE_ACTIVE). Thus, EthIf may detect a mismatch between the current requested mode (ETH_MODE_DOWN) and the current PHY state (ETH_MODE_ACTIVE) and re-triggers the PHY to transit back to sleep mode. To avoid this, a timer to delay the sleep mode execution is started. As long as the timer is running, the Ethernet transceiver driver will not execute a requested sleep mode.
- If ETHTRCV_WAKEUP_BY_ASYNCHRONOUS_CHECK is configured, then the timer to delay a sleep execution may be started twice. First start in the context of this function and second start in the context of the EthTrcv_Mainfunction where the check for wake-up reasons is performed.

[SWS_EthTrcv_00185]

Upstream requirements: SRS_Eth_00108, SRS_Eth_00107

⌈If the function EthTrcv_CheckWakeup() is called and EthTrcvWakeUpSupport is set to ETHTRCV_WAKEUP_BY_ASYNCHRONOUS_CHECK, the Ethernet Transceiver Driver shall store the request to asynchronously check for a wake-up of used

Ethernet hardware (e.g. the OA TC10 compliant PHY) in the context of the EthTrcv_MainFunction.⌋

Note: Asychronous check for a wake-up could be used for Ethernet switches, due to the time consumption to read out all Ethernet switch ports of an Ethernet switch. E.g. the Ethernet switch is signaled that an interrupt of one or more of its Ethernet switch ports occurred. The corresponding host ECU (ECU that maintain an Ethernet switch) has to read out all registers of the cooresponding PHYs, to check which PHYs signaled the interrupt and the reason for the interrupt. This could be done in the context of the EthTrcv_MainFunction() asynchronously (and NOT synchronously in the context of EthTrcv_CheckWakeup()) to support timing constraints regarding runtime of the host ECU.

[SWS_EthTrcv_00137] ⌈If development error detection is enabled: The function EthTrcv_CheckWakeup() shall check that the service EthTrcv_Init was previously called. If the check fails, the function shall raise the development error ETHTRCV_E_UNINIT.⌋

[SWS_EthTrcv_00138] ⌈If development error detection is enabled: The function EthTrcv_CheckWakeup() shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_TRCV_IDX.⌋

[SWS_EthTrcv_00140] ⌈*EthTrcv_Init()* shall be called before *EthTrcv_CheckWakeup()*.⌋

8.3.6 EthTrcv_StartAutoNegotiation

[SWS_EthTrcv_00055] Definition of API function EthTrcv_StartAutoNegotiation ⌈

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
Available via	EthTrcv.h

⌋

[SWS_EthTrcv_00056] ⌈The function shall restart the automatic negotiation of the transmission parameters used by the indexed transceiver if the internal cable diagnostic state is different from ETHTRCV_CABLEDIAG_PENDING. Otherwise, the API shall return with E_NOT_OK.⌋

[SWS_EthTrcv_00057] ⌈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_UNINIT.⌋

[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.⌋

[SWS_EthTrcv_00059]

Upstream requirements: SRS_Eth_00149

⌈The function shall be pre compile time configurable On/Off by the configuration parameter: EthTrcvStartAutoNegotiationApi [ECUC_EthTrcv_00008] and EthTrcvEnablePLCA [ECUC_EthTrcv_00055].⌋

[SWS_EthTrcv_00060] ⌈*EthTrcv_Init()* shall be called before *EthTrcv_StartAutoNegotiation()*.⌋

[SWS_EthTrcv_00088] ⌈*EthTrcv_StartAutoNegotiation()* shall not be called or required by an upper layer BSW software component.⌋

8.3.7 EthTrcv_TransceiverLinkStateRequest

[SWS_EthTrcv_91025] Definition of API function EthTrcv_TransceiverLinkStateRequest ⌈

Service Name	EthTrcv_TransceiverLinkStateRequest	

Syntax	`Std_ReturnType EthTrcv_TransceiverLinkStateRequest (`
	`  uint8 TrcvIdx,`
	`  EthTrcv_LinkStateType LinkState`
	`)`	

Service ID [hex]	0x02	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.	
Parameters (in)	TrcvIdx	Index of the transceiver within the context of the Transceiver Driver
	LinkState	The link state of a physical connection.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Request the given link state for the given transceiver	
Available via	EthTrcv.h	

⌋

8.3.7.1 Point to point network topology

If EthTrcvPhysLayerType is not set to TRCV_PHYS_LAYER_TYPE_10BASE_T1S, the following requirements of this subsection have to be applied:

[SWS_EthTrcv_00151] ⌈The function shall start link training of the indexed transceiver if:

- the requested link state is ETHTRCV_LINK_STATE_ACTIVE and
- the internal cable diagnostic state is different from ETHTRCV_CABLEDIAG_PENDING and
- EthTrcvConnNeg is set to TRCV_CONN_NEG_MASTER or TRCV_CONN_NEG_AUTO.

If EthTrcvConnNeg is set to TRCV_CONN_NEG_SLAVE, the indexed transceiver shall be put in a state to wait for the link training of the link partner.⌋

[SWS_EthTrcv_00165] ⌈If EthTrcv_TransceiverLinkStateRequest() is called and the internal cable diagnostic state is ETHTRCV_CABLEDIAG_PENDING, the Ethernet

Transceiver driver shall store the link state request per Ethernet transceiver and proceed as specified in [SWS_EthTrcv_00162].⌋

[SWS_EthTrcv_00152] ⌈The function shall stop link training of the indexed transceiver, if the requested link state is ETHTRCV_LINK_STATE_DOWN and EthTrcvConnNeg is set to TRCV_CONN_NEG_MASTER or TRCV_CONN_NEG_AUTO.⌋

[SWS_EthTrcv_00153] ⌈The function shall put the link down of the indexed transceiver, if the requested link state is ETHTRCV_LINK_STATE_DOWN.⌋

[SWS_EthTrcv_00154] ⌈If the Ethernet transceiver is already in the requested link state, E_OK shall be returned and no development error shall be raised.⌋

8.3.7.2 Bus topology

If EthTrcvPhysLayerType is set to TRCV_PHYS_LAYER_TYPE_10BASE_T1S, the following requirement of this subsection has to be applied:

[SWS_EthTrcv_00197]

Status: DRAFT

⌈If EthTrcvPhysLayerType is set to TRCV_PHYS_LAYER_TYPE_10BASE_T1S, the function shall internally store the requested link state and return E_OK.⌋

Note: this state is needed in context of *EthTrcv_GetLinkState()*.

8.3.8 EthTrcv_GetLinkState

[SWS_EthTrcv_00061] Definition of API function EthTrcv_GetLinkState ⌈

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 Transceiver Driver
Parameters (inout)	None	

▽

△

Parameters (out)	LinkStatePtr	ETHTRCV_LINK_STATE_DOWN: transceiver is disconnected ETHTRCV_LINK_STATE_ACTIVE: transceiver is connected
Return value	Std_ReturnType	`E_OK`: success `E_NOT_OK`: transceiver could not be initialized
Description	Obtains the link state of the indexed transceiver	
Available via	EthTrcv.h	

⌋

[SWS_EthTrcv_00063] ⌈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_UNINIT.⌋

[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.⌋

[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_PARAM_POINTER.⌋

[SWS_EthTrcv_00066] ⌈The function shall be pre compile time configurable On/Off by the configuration parameter: EthTrcvGetLinkStateApi [ECUC_EthTrcv_00009].⌋

[SWS_EthTrcv_00067] ⌈*EthTrcv_Init()* shall be called before *EthTrcv_GetLinkState()*.⌋

8.3.8.1 Point to point network topology

If EthTrcvPhysLayerType is not set to TRCV_PHYS_LAYER_TYPE_10BASE_T1S, the following requirement of this subsection has to be applied:

[SWS_EthTrcv_00062]

Upstream requirements: SRS_Eth_00040

⌈The function shall read the current transceiver link state, expect for the following condition:

If EthTrcvWakeupSleepOnDatalineEnabled set to TRUE and the Ethernet Transceiver Driver detect that a sleep process is performed for the given TrcvIdx, then ETHTRCV_LINK_STATE_DOWN shall be returned.⌋

Note: OA TC10 compliant Ethernet hardware is not able to transfer data on the date line, if a sleep process is performed. In that case a link down is reported to the upper layer, to indicate that communication is not possible on data line.

8.3.8.2 Bus topology

If EthTrcvPhysLayerType is set to TRCV_PHYS_LAYER_TYPE_10BASE_T1S, the following requirements of this subsection have to be applied:

[SWS_EthTrcv_00198]

Status: DRAFT

⌈The function shall check the access to the Ethernet PHY to read the link register. If the check fails, the function shall raise the production error ETHTRCV_E_ACCESS return E_NOT_OK and state ETHTRCV_LINK_STATE_DOWN.⌋

[SWS_EthTrcv_00199]

Status: DRAFT

⌈The function shall return the stored requested state given by a previous call of *EthTrcv_TransceiverLinkStateRequest()* (see [SWS_EthTrcv_00197]) and return E_OK. If *EthTrcv_TransceiverLinkStateRequest()* was not called before, ETHTRCV_LINK_STATE_DOWN shall be returned with E_OK.⌋

[SWS_EthTrcv_00200]

Status: DRAFT

⌈If the PHY register indicates a link up, the function shall additionally read the PLCA Status register of the PHY. If the PLCA mode is not active, the Ethernet transceiver driver shall report a runtime error ETHTRCV_E_MDI_COMMUNICATION and return E_NOT_OK.⌋

Note: In case of head-node failure, the client node will switch the media access mode from PLCA to CSMA/CD. Thus, the client node is still able to communicate.

8.3.9 EthTrcv_GetBaudRate

[SWS_EthTrcv_00068] Definition of API function EthTrcv_GetBaudRate ⌈

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 ETHTRCV_BAUD_RATE_2500MBIT: 2500MBit 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	
Available via	EthTrcv.h	

⌋

[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_Init was previously called. If the check fails, the function shall raise the development error ETHTRCV_E_UNINIT.⌋

[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.⌋

[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_PARAM_POINTER.⌋

[SWS_EthTrcv_00073] ⌈The function shall be pre compile time configurable On/Off by the configuration parameter: EthTrcvGetBaudRateApi [ECUC_EthTrcv_00010].⌋

[SWS_EthTrcv_00074] ⌈*EthTrcv_Init()* shall be called before *EthTrcv_GetBaudRate()*.⌋

[SWS_EthTrcv_00089] ⌈*EthTrcv_GetBaudRate()* shall not be called or required by an upper layer BSW software component.⌋

8.3.10 EthTrcv_GetDuplexMode

[SWS_EthTrcv_00075] Definition of API function EthTrcv_GetDuplexMode ⌈

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	
Available via	EthTrcv.h	

⌋

[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_Init was previously called. If the check fails, the function shall raise the development error ETHTRCV_E_UNINIT.⌋

[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.⌋

[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_PARAM_POINTER.⌋

[SWS_EthTrcv_00080] ⌈The function shall be pre compile time configurable On/Off by the configuration parameter: EthTrcvGetDuplexModeApi [ECUC_EthTrcv_00011].⌋

[SWS_EthTrcv_00081] ⌈*EthTrcv_Init()* shall be called before *EthTrcv_GetDuplexMode()*.⌋

[SWS_EthTrcv_00090] ⌈*EthTrcv_GetDuplexMode()* shall not be called or required by an upper layer BSW software component.⌋

8.3.11 EthTrcv_SetPhyTestMode

[SWS_EthTrcv_91003] Definition of API function EthTrcv_SetPhyTestMode

Upstream requirements: SRS_Eth_00117

⌈

Service Name	EthTrcv_SetPhyTestMode	

Syntax	`Std_ReturnType EthTrcv_SetPhyTestMode (`
	`  uint8 TrcvIdx,`
	`  EthTrcv_PhyTestModeType Mode`
	`)`	

Service ID [hex]	0x11	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.	
Parameters (in)	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
	Mode	Test mode to be activated
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted.
Description	Activates a given test mode.	
Available via	EthTrcv.h	

⌋

[SWS_EthTrcv_00166] ⌈If the internal cable diagnostic state is ETHTRCV_CABLEDIAG_PENDING, EthTrcv_SetPhyTestMode() shall return E_NOT_OK.⌋

[SWS_EthTrcv_00147]

Upstream requirements: SRS_Eth_00117

⌈If development error detection is enabled: the function EthTrcv_SetPhyTestMode() shall check the parameter Mode for being supported by the hardware. If the check fails, the function shall raise the development error ETHTRCV_E_NOT_SUPPORTED.⌋

[SWS_EthTrcv_00169] ⌈The function shall be pre compile time configurable On/Off by the configuration parameter EthTrcvSetPhyTestModeApi [ECUC_EthTrcv_00047].⌋

8.3.12 EthTrcv_SetPhyLoopbackMode

[SWS_EthTrcv_91005] Definition of API function EthTrcv_SetPhyLoopbackMode

Upstream requirements: SRS_Eth_00117

⌈

Service Name	EthTrcv_SetPhyLoopbackMode	

Syntax	`Std_ReturnType EthTrcv_SetPhyLoopbackMode (`
	`uint8 TrcvIdx,`
	`EthTrcv_PhyLoopbackModeType Mode`
	`)`	

Service ID [hex]	0x12	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.	
Parameters (in)	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
	Mode	Loopback mode to be activated
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted.
Description	Activates a given loopback mode.	
Available via	EthTrcv.h	

⌋

[SWS_EthTrcv_00167] ⌈If the internal cable diagnostic state is ETHTRCV_CABLEDIAG_PENDING, the EthTrcv_SetPhyLoopbackMode() shall return E_NOT_OK.⌋

[SWS_EthTrcv_00149]

Upstream requirements: SRS_Eth_00117

⌈If development error detection is enabled: the function EthTrcv_SetPhyLoopbackMode shall check the parameter Mode for being supported by the hardware. If the check fails, the function shall raise the development error ETHTRCV_E_NOT_SUPPORTED.⌋

8.3.13 EthTrcv_GetPhySignalQuality

[SWS_EthTrcv_91001] Definition of API function EthTrcv_GetPhySignalQuality

Upstream requirements: SRS_Eth_00117

⌈

Service Name	EthTrcv_GetPhySignalQuality	

Syntax	`Std_ReturnType EthTrcv_GetPhySignalQuality (`
	`  uint8 TrcvIdx,`
	`  uint32* SignalQualityPtr`
	`)`	

Service ID [hex]	0x10	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.	
Parameters (in)	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
Parameters (inout)	None	
Parameters (out)	SignalQualityPtr	Pointer to the memory where the signal quality shall be stored.
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted.
Description	Obtains the current signal quality of the link of the indexed transceiver	
Available via	EthTrcv.h	

⌋

8.3.14 EthTrcv_SetPhyTxMode

[SWS_EthTrcv_91007] Definition of API function EthTrcv_SetPhyTxMode

Upstream requirements: SRS_Eth_00117

⌈

Service Name	EthTrcv_SetPhyTxMode	

Syntax	`Std_ReturnType EthTrcv_SetPhyTxMode (`
	`  uint8 TrcvIdx,`
	`  EthTrcv_PhyTxModeType Mode`
	`)`	

Service ID [hex]	0x13	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.	
Parameters (in)	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
	Mode	Transmission mode to be activated
Parameters (inout)	None	
Parameters (out)	None	

▽

△

Return value	Std_ReturnType	E_OK: The request has been accepted E_NOT_OK: The request has not been accepted
Description	Activates a given transmission mode.	
Available via	EthTrcv.h	

⌋

[SWS_EthTrcv_00168] ⌈If the internal cable diagnostic state is ETHTRCV_CABLEDIAG_PENDING, the EthTrcv_SetPhyTxMode shall return E_NOT_OK.⌋

[SWS_EthTrcv_00148] ⌈If development error detection is enabled: the function EthTrcv_SetPhyTxMode shall check the parameter Mode for being supported by the hardware. If the check fails, the function shall raise the development error ETHTRCV_E_NOT_SUPPORTED.⌋

8.3.15 EthTrcv_ RunCableDiagnostic

[SWS_EthTrcv_91011] Definition of API function EthTrcv_RunCableDiagnostic ⌈

Service Name	EthTrcv_RunCableDiagnostic	

Syntax	`Std_ReturnType EthTrcv_RunCableDiagnostic (`
	`  uint8 TrcvIdx`
	`)`	

Service ID [hex]	0x16	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: The trigger has been accepted. E_NOT_OK: The trigger has not been accepted.
Description	Trigger the cable diagnostics for the given Ethernet transceiver.	
Available via	EthTrcv.h	

⌋

[SWS_EthTrcv_00170] ⌈The function shall be pre compile time configurable On/Off by the configuration parameter EthTrcvEnableCableDiagnosticApi [ECUC_EthTrcv_00054].⌋

8.3.16 EthTrcv_GetCableDiagnosticsResult

[SWS_EthTrcv_91009] Definition of API function EthTrcv_GetCableDiagnosticsResult

Upstream requirements: SRS_Eth_00117

⌈

Service Name	EthTrcv_GetCableDiagnosticsResult	

Syntax	`Std_ReturnType EthTrcv_GetCableDiagnosticsResult (`
	`  uint8 TrcvIdx,`
	`  EthTrcv_CableDiagResultType* ResultPtr`
	`)`	

Service ID [hex]	0x14	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.	
Parameters (in)	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
Parameters (inout)	None	
Parameters (out)	ResultPtr	Pointer to the location where the cable diagnostics result shall be stored
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Retrieves the cable diagnostics result of a given transceiver.	
Available via	EthTrcv.h	

⌋

[SWS_EthTrcv_00171] ⌈The function shall be pre compile time configurable On/Off by the configuration parameter EthTrcvEnableCableDiagnosticApi [ECUC_EthTrcv_00054].⌋

8.3.17 EthTrcv_GetPhyIdentifier

[SWS_EthTrcv_91010] Definition of API function EthTrcv_GetPhyIdentifier

Upstream requirements: SRS_Eth_00117

⌈

Service Name	EthTrcv_GetPhyIdentifier

Syntax	`Std_ReturnType EthTrcv_GetPhyIdentifier (`
	`  uint8 TrcvIdx,`
	`  uint32* OrgUniqueIdPtr,`
	`  uint8* ModelNrPtr,`
	`  uint8* RevisionNrPtr`
	`)`

▽

△

Service ID [hex]	0x15	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.	
Parameters (in)	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
Parameters (inout)	None	
Parameters (out)	OrgUniqueIdPtr	Pointer to the memory where the Organizationally Unique Identifier shall be stored.
	ModelNrPtr	Pointer to the memory where the Manufacturer's Model Number shall be stored.
	RevisionNrPtr	Pointer to the memory where the Revision Number shall be stored.
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Obtains the PHY identifier of the Ethernet Transceiver according to IEEE 802.3-2015 chapter 22.2.4.3.1 PHY Identifer.	
Available via	EthTrcv.h	

⌋

[SWS_EthTrcv_00172] ⌈The function shall be pre compile time configurable On/Off by the configuration parameter EthTrcvGetPhyIdentifierApi [ECUC_EthTrcv_00046].⌋

8.3.18 EthTrcv_GetMacMethod

[SWS_EthTrcv_91014] Definition of API function EthTrcv_GetMacMethod ⌈

Service Name	EthTrcv_GetMacMethod	

Syntax	`Std_ReturnType EthTrcv_GetMacMethod (`
	`  uint8* TrcvIdx,`
	`  EthTrcv_MacMethodType* MacModePtr`
	`)`	

Service ID [hex]	0x18	
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)	MacModePtr	ETHTRCV_MAC_TYPE_CSMA_CD: Carrier-sense multiple access with collicion detection ETHTRCV_MAC_TYPE_PLCA: Physical layer collision avoidance
Return value	Std_ReturnType	`E_OK`: success `E_NOT_OK`: MacType could not be returned
Description	Obtains the media access mode of the transceiver when EthTrcvDuplexMode is configured as ETHTRCV_DUPLEX_MODE_HALF	
Available via	EthTrcv.h	

⌋

[SWS_EthTrcv_00174]

Upstream requirements: SRS_Eth_00149

⌈If the parameter EthTrcvDuplexMode is set to ETHTRCV_DUPLEX_MODE_HALF the function shall return the current transceiver media access mode. Otherwise, it shall return E_NOT_OK.⌋

[SWS_EthTrcv_00175] ⌈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_UNINIT.⌋

[SWS_EthTrcv_00176] ⌈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.⌋

[SWS_EthTrcv_00177] ⌈If development error detection is enabled: the function shall check the parameter MacModePtr for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_PARAM_POINTER.⌋

[SWS_EthTrcv_00178] ⌈*EthTrcv_Init()* shall be called before *EthTrcv_GetMacMethod()*.⌋

8.3.19 EthTrcv_GetVersionInfo

[SWS_EthTrcv_00082] Definition of API function EthTrcv_GetVersionInfo ⌈

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	
Available via	EthTrcv.h	

⌋

[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_PARAM_POINTER.⌋

[SWS_EthTrcv_00173] ⌈The function shall be pre compile time configurable On/Off by the configuration parameter EthTrcvVersionInfoApi [ECUC_EthTrcv_00004].⌋

8.3.20 EthTrcv_MacSecUpdateSecY

[SWS_EthTrcv_91026] Definition of API function EthTrcv_MacSecUpdateSecY

Status: DRAFT

⌈

Service Name	EthTrcv_MacSecUpdateSecY (draft)	

Syntax	`Std_ReturnType EthTrcv_MacSecUpdateSecY (`
	`  uint8 TrcvIdx,`
	`  const Mka_MacSecConfigType* MACsecCfgPtr,`
	`  uint64 TxSci`
	`)`	

Service ID [hex]	0x19	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different TrcvIdx, Non reentrant for the same TrcvIdx	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver
	MACsecCfgPtr	Pointer to the structure to configure a MACsec Entity (SecY)
	TxSci	Secure Channel Identifier for the MACsec's Transmission Secure channel
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Requests the Ethernet Transceiver to update the SecY/PAC of the PHY with the provided parameters. A Transmission Secure Channel with the provided SCI shall be configured during the first call. A pointer to a MACsec Basic Parameters Configuration file shall be provided to create the Secure Channel. **Tags:** atp.Status=draft	
Available via	EthTrcv.h	

⌋

8.3.21 EthTrcv_MacSecInitRxSc

[SWS_EthTrcv_91027] Definition of API function EthTrcv_MacSecInitRxSc

Status: DRAFT

⌈

Service Name	EthTrcv_MacSecInitRxSc (draft)	

Syntax	`Std_ReturnType EthTrcv_MacSecInitRxSc (`
	`uint8 TrcvIdx,`
	`uint64 Sci`
	`)`	

Service ID [hex]	0x1a	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different TrcvIdx, Non reentrant for the same TrcvIdx	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver
	Sci	Secure Channel Identifier for the MACsec's Reception Secure channel
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Requests the Ethernet Transceiver Driver to configure a Reception Secure Channel for the given Secure Channel Identifier. **Tags:** atp.Status=draft	
Available via	EthTrcv.h	

⌋

8.3.22 EthTrcv_MacSecResetRxSc

[SWS_EthTrcv_91028] Definition of API function EthTrcv_MacSecResetRxSc

Status: DRAFT

⌈

Service Name	EthTrcv_MacSecResetRxSc (draft)	

Syntax	`Std_ReturnType EthTrcv_MacSecResetRxSc (`
	`uint8 TrcvIdx,`
	`uint64 Sci`
	`)`	

Service ID [hex]	0x1b	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different TrcvIdx, Non reentrant for the same TrcvIdx	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver

▽

△

	Sci	Secure Channel Identifier for the MACsec's Reception Secure channel
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Requests the Ethernet Transceiver Driver to reset to default the MACsec values of the Reception Secure Channel for the given Secure Channel Identifier. **Tags:** atp.Status=draft	
Available via	EthTrcv.h	

⌋

8.3.23 EthTrcv_MacSecAddTxSa

[SWS_EthTrcv_91030] Definition of API function EthTrcv_MacSecAddTxSa

Status: DRAFT

⌈

Service Name	EthTrcv_MacSecAddTxSa (draft)	

Syntax	`Std_ReturnType EthTrcv_MacSecAddTxSa (`
	`  uint8 TrcvIdx,`
	`  uint8 An,`
	`  uint64 NextPn,`
	`  uint32 Ssci,`
	`  const Mka_SakKeyPtrType* KeysPtr,`
	`  boolean Active`
	`)`	

Service ID [hex]	0x1c	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different TrcvIdx, Non reentrant for the same TrcvIdx	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver
	An	Association Number to use in the MACsec's transmission secure association
	NextPn	Next accepted Packet Number in the MACsec's transmission secure association
	Ssci	Short Secure Channel Identifiert used in the MACsec's transmission secure association
	KeysPtr	Pointer to the SAKs Key (and needed Key information) to use in the MACsec's transmission secure association
	Active	Boolean to enable/disable the MACsec's transmission secure association
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted

▽

△

Description	Requests the Ethernet Transceiver Driver to create a Transmission Secure Association in the Transceiver. The Short Secure Channel Identifier is included to support XPN configurations. **Tags:** atp.Status=draft
Available via	EthTrcv.h

⌋

8.3.24 EthTrcv_MacSecUpdateTxSa

[SWS_EthTrcv_91029] Definition of API function EthTrcv_MacSecUpdateTxSa

Status: DRAFT

⌈

Service Name	EthTrcv_MacSecUpdateTxSa (draft)	

Syntax	`Std_ReturnType EthTrcv_MacSecUpdateTxSa (`
	`  uint8 TrcvIdx,`
	`  uint8 An,`
	`  uint64 NextPn,`
	`  boolean Active`
	`)`	

Service ID [hex]	0x1d	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different TrcvIdx, Non reentrant for the same TrcvIdx	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver
	An	Association Number to use in the MACsec's transmission secure association
	NextPn	Next accepted Packet Number in the MACsec's transmission secure association
	Active	Boolean to enable/disable the MACsec's transmission secure association
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Requests the Ethernet Transceiver Driver to update the Transmission Secure Association with the given Packet Number. The Active parameter is included to change the specified AN status. **Tags:** atp.Status=draft	
Available via	EthTrcv.h	

⌋

8.3.25 EthTrcv_MacSecDeleteTxSa

[SWS_EthTrcv_91031] Definition of API function EthTrcv_MacSecDeleteTxSa

Status: DRAFT

⌈

Service Name	EthTrcv_MacSecDeleteTxSa (draft)	

Syntax	`Std_ReturnType EthTrcv_MacSecDeleteTxSa (`
	`  uint8 TrcvIdx,`
	`  uint8 An`
	`)`	

Service ID [hex]	0x1e	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different TrcvIdx, Non reentrant for the same TrcvIdx	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver
	An	Association Number to use in the MACsec's transmission secure association
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Request the Ethernet Transceiver Driver to remove the Transmission Secure Association identified by the provided Association Number. **Tags:** atp.Status=draft	
Available via	EthTrcv.h	

⌋

8.3.26 EthTrcv_MacSecAddRxSa

[SWS_EthTrcv_91032] Definition of API function EthTrcv_MacSecAddRxSa

Status: DRAFT

⌈

Service Name	EthTrcv_MacSecAddRxSa (draft)

Syntax	`Std_ReturnType EthTrcv_MacSecAddRxSa (`
	`  uint8 TrcvIdx,`
	`  uint8 An,`
	`  uint64 LowestPn,`
	`  uint32 Ssci,`
	`  const Mka_SakKeyPtrType* KeysPtr,`
	`  boolean Active`
	`)`

Service ID [hex]	0x1f
Sync/Async	Asynchronous

▽

△

Reentrancy	Reentrant for different TrcvIdx, Non reentrant for the same TrcvIdx	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver
	An	Association Number to use in the MACsec's reception secure association
	LowestPn	Lowest accepted Packet Number in the MACsec's reception secure association
	Ssci	Short Secure Channel Identifiert used in the MACsec's reception secure association
	KeysPtr	Pointer to the SAKs Key (and needed Key information) to use in the MACsec's reception secure association
	Active	Boolean to enable/disable the MACsec's reception secure association
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Request the Ethernet Transceiver Driver to create a Reception Secure Association in the Transceiver. The Short Secure Channel Identifier is included to support XPN configurations. **Tags:** atp.Status=draft	
Available via	EthTrcv.h	

⌋

8.3.27 EthTrcv_MacSecUpdateRxSa

[SWS_EthTrcv_91033] Definition of API function EthTrcv_MacSecUpdateRxSa

Status: DRAFT

⌈

Service Name	EthTrcv_MacSecUpdateRxSa (draft)	

Syntax	`Std_ReturnType EthTrcv_MacSecUpdateRxSa (`
	`  uint8 TrcvIdx,`
	`  uint8 An,`
	`  uint64 LowestPn,`
	`  boolean Active`
	`)`	

Service ID [hex]	0x20	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different TrcvIdx, Non reentrant for the same TrcvIdx	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver
	An	Association Number to use in the MACsec's reception secure association
	LowestPn	Lowest accepted Packet Number in the MACsec's reception secure association
	Active	Boolean to enable/disable the MACsec's reception secure association

▽

△

Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: The request has been accepted E_NOT_OK: The request has not been accepted
Description	Request the Ethernet Transceiver Driver to update the Reception Secure Association with the given Packet Number. The Active parameter is included to change the specified AN status. **Tags:** atp.Status=draft	
Available via	EthTrcv.h	

⌋

8.3.28 EthTrcv_MacSecDeleteRxSa

[SWS_EthTrcv_91034] Definition of API function EthTrcv_MacSecDeleteRxSa

Status: DRAFT

⌈

Service Name	EthTrcv_MacSecDeleteRxSa (draft)	

Syntax	`Std_ReturnType EthTrcv_MacSecDeleteRxSa (`
	`  uint8 TrcvIdx,`
	`  uint8 An`
	`)`	

Service ID [hex]	0x21	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different TrcvIdx, Non reentrant for the same TrcvIdx	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver
	An	Association Number to use in the MACsec's reception secure association
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: The request has been accepted E_NOT_OK: The request has not been accepted
Description	Request the Ethernet Transceiver Driver to remove the Reception Secure Association identified by the provided Association Number. **Tags:** atp.Status=draft	
Available via	EthTrcv.h	

⌋

8.3.29 EthTrcv_MacSecGetTxSaNextPn

[SWS_EthTrcv_91035] Definition of API function EthTrcv_MacSecGetTxSaNextPn

Status: DRAFT

⌈

Service Name	EthTrcv_MacSecGetTxSaNextPn (draft)	

Syntax	`Std_ReturnType EthTrcv_MacSecGetTxSaNextPn (`
	`  uint8 TrcvIdx,`
	`  uint8 An,`
	`  uint64* NextPnPtr`
	`)`	

Service ID [hex]	0x22	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different TrcvIdx, Non reentrant for the same TrcvIdx	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver
	An	Association Number to use in the MACsec's reception secure association
Parameters (inout)	None	
Parameters (out)	NextPnPtr	Pointer to the Next Packet Number read out from the MACsec Entity (SecY)
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Request the Ethernet Transceiver Driver to return the Packet Number that is used for the next packet in the given Transmission Secure Association. **Tags:** atp.Status=draft	
Available via	EthTrcv.h	

⌋

8.3.30 EthTrcv_MacSecGetMacSecStatistics

[SWS_EthTrcv_91036] Definition of API function EthTrcv_MacSecGetMacSecStatistics

Status: DRAFT

⌈

Service Name	EthTrcv_MacSecGetMacSecStatistics (draft)

Syntax	`Std_ReturnType EthTrcv_MacSecGetMacSecStatistics (`
	`  uint8 TrcvIdx,`
	`  Mka_Stats_SecYType* MacSecStatsPtr`
	`)`

Service ID [hex]	0x23

▽

△

Sync/Async	Asynchronous	
Reentrancy	Reentrant for different TrcvIdx, Non reentrant for the same TrcvIdx	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver
Parameters (inout)	None	
Parameters (out)	MacSecStatsPtr	Pointer to a structure including the MACsec statistics of an MKA participant
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Request the Ethernet Transceiver Driver to provide MACsec statistics. **Tags:** atp.Status=draft	
Available via	EthTrcv.h	

⌋

8.3.31 EthTrcv_MacSecSetControlledPortEnabled

[SWS_EthTrcv_91037] Definition of API function EthTrcv_MacSecSetControlledPortEnabled

Status: DRAFT

⌈

Service Name	EthTrcv_MacSecSetControlledPortEnabled (draft)	

Syntax	`Std_ReturnType EthTrcv_MacSecSetControlledPortEnabled (`
	`  uint8 TrcvIdx,`
	`  boolean ControlledPortEnabled`
	`)`	

Service ID [hex]	0x24	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different TrcvIdx, Non reentrant for the same TrcvIdx	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver
	ControlledPortEnabled	Boolean to activate the Controlled Port of the PAE
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Requests to set the Controlled Port enabled parameter of a PAE. **Tags:** atp.Status=draft	
Available via	EthTrcv.h	

⌋

8.3.32 EthTrcv_EthSwitchMacSecUpdateSecY

[SWS_EthTrcv_91038] Definition of API function EthTrcv_EthSwitchMacSecUpdateSecY

Status: DRAFT

⌈

Service Name	EthTrcv_EthSwitchMacSecUpdateSecY (draft)	

Syntax	`Std_ReturnType EthTrcv_EthSwitchMacSecUpdateSecY (`
	`  uint8 TrcvIdx,`
	`  const Mka_MacSecConfigType* MACsecCfgPtr,`
	`  uint64 TxSci`
	`)`	

Service ID [hex]	0x25	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different TrcvIdx, Non reentrant for the same TrcvIdx	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver
	MACsecCfgPtr	Pointer to the structure to configure a MACsec Entity (SecY)
	TxSci	Secure Channel Identifier for the MACsec's Transmission Secure channel
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Requests the Ethernet Transceiver to update the SecY/PAC of the PHY with the provided parameters. A Transmission Secure Channel with the provided SCI shall be configured during the first call. A pointer to a MACsec Basic Parameters Configuration file shall be provided to create the Secure Channel. **Tags:** atp.Status=draft	
Available via	EthTrcv.h	

⌋

8.3.33 EthTrcv_EthSwitchMacSecAddTxSa

[SWS_EthTrcv_91039] Definition of API function EthTrcv_EthSwitchMacSecAddTxSa

Status: DRAFT

⌈

Service Name	EthTrcv_EthSwitchMacSecAddTxSa (draft)	

Syntax	`Std_ReturnType EthTrcv_EthSwitchMacSecAddTxSa (`
	`  uint8 TrcvIdx,`
	`  uint8 An,`
	`  uint64 NextPn,`
	`  uint32 Ssci,`
	`  const Mka_SakKeyPtrType* KeysPtr,`
	`  boolean Active`
	`)`	

Service ID [hex]	0x26	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different TrcvIdx, Non reentrant for the same TrcvIdx	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver
	An	Association Number to use in the MACsec's transmission secure association
	NextPn	Next accepted Packet Number in the MACsec's transmission secure association
	Ssci	Short Secure Channel Identifiert used in the MACsec's transmission secure association
	KeysPtr	Pointer to the SAKs Key (and needed Key information) to use in the MACsec's transmission secure association
	Active	Boolean to enable/disable the MACsec's transmission secure association
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Requests the Ethernet Transceiver Driver to create a Transmission Secure Association in the Transceiver. The Short Secure Channel Identifier is included to support XPN configurations. **Tags:** atp.Status=draft	
Available via	EthTrcv.h	

⌋

8.3.34 EthTrcv_SwitchMacSecAddRxSa

[SWS_EthTrcv_91040] Definition of API function EthTrcv_SwitchMacSecAddRxSa

Status: DRAFT

⌈

Service Name	EthTrcv_SwitchMacSecAddRxSa (draft)	

Syntax	`Std_ReturnType EthTrcv_SwitchMacSecAddRxSa (`
	`  uint8 TrcvIdx,`
	`  uint8 An,`
	`  uint64 LowestPn,`
	`  uint32 Ssci,`
	`  const Mka_SakKeyPtrType* KeysPtr,`
	`  boolean Active`
	`)`	

Service ID [hex]	0x27	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different TrcvIdx, Non reentrant for the same TrcvIdx	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver
	An	Association Number to use in the MACsec's transmission secure association
	LowestPn	Lowest accepted Packet Number in the MACsec's reception secure association
	Ssci	Short Secure Channel Identifiert used in the MACsec's reception secure association
	KeysPtr	Pointer to the SAKs Key (and needed Key information) to use in the MACsec's reception secure association
	Active	Boolean to enable/disable the MACsec's reception secure association
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Request the Ethernet Transceiver Driver to create a Reception Secure Association in the Transceiver. The Short Secure Channel Identifier is included to support XPN configurations. **Tags:** atp.Status=draft	
Available via	EthTrcv.h	

⌋

8.3.35 EthTrcv_SwitchMacSecGetMacSecStatistics

[SWS_EthTrcv_91041] Definition of API function EthTrcv_SwitchMacSecGetMacSecStatistics

Status: DRAFT

⌈

Service Name	EthTrcv_SwitchMacSecGetMacSecStatistics (draft)	

Syntax	`Std_ReturnType EthTrcv_SwitchMacSecGetMacSecStatistics (`
	`  uint8 TrcvIdx,`
	`  Mka_Stats_SecYType* MacSecStatsPtr`
	`)`	

Service ID [hex]	0x28	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different TrcvIdx, Non reentrant for the same TrcvIdx	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver
Parameters (inout)	None	
Parameters (out)	MacSecStatsPtr	Pointer to a structure including the MACsec statistics of an MKA participant
Return value	Std_ReturnType	`E_OK`: The request has been accepted `E_NOT_OK`: The request has not been accepted
Description	Request the Ethernet Transceiver Driver to provide MACsec statistics. **Tags:** atp.Status=draft	
Available via	EthTrcv.h	

⌋

This chapter lists all functions provided by the Ethernet Transceiver module to lower layer modules.

8.4 Scheduled functions

8.4.1 EthTrcv_MainFunction

[SWS_EthTrcv_00106] Definition of scheduled function EthTrcv_MainFunction ⌈

Service Name	EthTrcv_MainFunction

Syntax	`void EthTrcv_MainFunction (`
	`  void`
	`)`

Service ID [hex]	0x0c
Description	Used for polling state changes and wakeup reasons. Calls EthIf_TrcvModeIndication when the transceiver mode changed. Stores wakeup events if EthTrcvWakeUpSupport is set to ETHTRCV_WAKEUP_BY_POLLING.

▽

△

Available via	SchM_EthTrcv.h

⌋

[SWS_EthTrcv_00107] ⌈Used for polling state changes. Calls EthIf_TrcvModeIndication when the transceiver mode changed.⌋

[SWS_EthTrcv_00141] ⌈If EthTrcvWakeUpSupport is set to ETHTRCV_WAKEUP_POLLING, the function EthTrcv_MainFunction() shall check for wake up reasons. If wake-up reasons were detected, the wake-up events shall be stored and EcuM_CheckWakeup shall be called with the corresponding wake-up source (see EthTrcvWakeupMap configuration [ECUC_EthTrcv_00027]).⌋

[SWS_EthTrcv_00191]

Upstream requirements: SRS_Eth_00108

⌈If EthTrcvWakeUpSupport is set to ETHTRCV_WAKEUP_BY_ASYNCHRONOUS_CHECK, the EthTrcv_MainFunction shall check the maintained Ethernet hardware (PHY) for a signaled wake-up (e.g. Wakeup.Indication if using OA TC10 [3] compliant PHYs) of those EthTrcvs where the check for wake-up was requested by EthTrcv_CheckWakeup. If a wake-up is detected, the Ethernet Transceiver Driver shall inform the EcuM by calling EcuM_SetWakeupEvent with the corresponding WakeupSource.⌋

[SWS_EthTrcv_00192]

Status: DRAFT
Upstream requirements: SRS_ModeMgm_09267, SRS_Eth_00152

⌈If EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE and EthTrcvActAsSlavePassiveEnabled is set to TRUE, the EthTrcv_MainFunction shall evaluate the maintained Ethernet hardware (PHY) for a signaled Sleep.Indication (see [3]). If a Sleep.Indication is detected, the Ethernet Transceiver Driver shall inform the EthIf by calling EthIf_SleepIndication.⌋

[SWS_EthTrcv_00193]

Upstream requirements: SRS_Eth_00151

⌈If EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE, the EthTrcv_MainFunction shall process running delay timers. If a sleep delay timer exceeds and the requested transceiver mode is still ETH_MODE_DOWN, the Ethernet Transceiver driver shall trigger a sleep request according to the specification of OA TC10 (see [3]).⌋

The EthTrcv_MainFunction checks all EthTrcvs with EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE where the sleep request was triggered (requested mode ETH_MODE_DOWN): If no SleepFail.Indication was signaled within the configured time period specified by EthTrcvSleepRequestRepetitionPeriod, then the sleep request was confirmed by the connected Ethernet hardware (PHY). Thus, the Ethernet hardware transit to sleep.

The following requirements specify the procedure in case a sleep request was rejected by the counter part of the Ethernet connection.

[SWS_EthTrcv_00194]

Upstream requirements: SRS_Eth_00155

⌈If EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE, the requested transceiver mode is still ETH_MODE_DOWN and SleepFail.Indication was reported less than EthTrcvSleepRequestNumberOfRepetitions, the Ethernet Transceiver driver shall trigger a sleep request according to the specification of OA TC10 (see [3]) after EthTrcvSleepRequestRepetitionPeriod has been elapsed.⌋

[SWS_EthTrcv_00195]

Upstream requirements: SRS_Eth_00155

⌈If EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE, the requested transceiver mode is still ETH_MODE_DOWN and SleepFail.Indication was reported EthTrcvSleepRequestNumberOfRepetitions times, the Ethernet Transceiver driver shall call EthIf_TrcvModeIndication(ETH_MODE_DOWN). If EthTrcvForceSleepEnabled is set to true the Ethernet Transceiver driver shall additionally force the Ehternet hardware (PHY) to go to sleep.⌋

[SWS_EthTrcv_00196]

Upstream requirements: SRS_Eth_00154

⌈After each call of EthTrcv_SetTransceiverMode with ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST for a particular EthTrcv the EthTrcv_MainFunction shall repeat a wake-up request EthTrcvWakeupRequestNumberOfRepetitions times with a delay of EthTrcvWakeUpRequestRepetitionPeriod in between as long as no EthTrcv_SetTransceiverMode with ETH_MODE_DOWN occurs for the same EthTrcv.⌋

8.5 Expected interfaces

This chapter lists all interfaces required from other modules.

8.5.1 Mandatory Interfaces

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

[SWS_EthTrcv_00085] Definition of mandatory interfaces required by module EthTrcv ⌈

API Function	***Header File***	***Description***
Dem_SetEventStatus	Dem.h	Called by SW-Cs or BSW modules to report monitor status information to the Dem. BSW modules calling Dem_SetEventStatus can safely ignore the return value. This API will be available only if ({Dem/DemConfigSet/DemEventParameter/DemEventReportingType} == STANDARD_REPORTING)
EthIf_TrcvModeIndication	EthIf.h	Called asynchronously when a mode change has been read out. If the function is triggered by previous call of EthTrcv_SetTransceiverMode it can directly be called within the trigger function.
SchM_Enter_EthTrcv	SchM_<Mip>.h	Invokes the SchM_Enter function to enter a module local exclusive area.
SchM_Exit_EthTrcv	SchM_<Mip>.h	Invokes the SchM_Exit function to exit an exclusive area.

⌋

8.5.2 Optional Interfaces

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

[SWS_EthTrcv_00086] Definition of optional interfaces requested by module EthTrcv ⌈

API Function	***Header File***	***Description***
Det_ReportError	Det.h	Service to report development errors.
EcuM_SetWakeupEvent	EcuM.h	Sets the wakeup event.
Eth_ReadMii	Eth.h	Reads a transceiver register
Eth_ReadMmd	Eth.h	Reads a transceiver register using Clause45 access if supported by hardware or implements a Clause45 access using Clause 22 operations
Eth_WriteMii	Eth.h	Configures a transceiver register or triggers a function offered by the receiver
Eth_WriteMmd	Eth.h	Writes a transceiver register using Clause 45 access or implements a Clause45 access using Clause 22 operations
EthIf_SleepIndication (draft)	EthIf.h	This API is called by the corresponding EthTrcv, if a sleep indication was detected on the network. This could be used e.g. for Ethernet hardware which is compliant to the OA TC10. In this case the Ethernet hardware (PHY) detect an Sleep.Indication which was triggered by a Sleep.Request of the connected link partner. **Tags:** atp.Status=draft

▽

△

API Function	***Header File***	***Description***
EthSwt_ReadTrcvRegister	EthSwt.h	Generic API for reading the content of a transceiver register
EthSwt_WriteTrcvRegister	EthSwt.h	Generic API for writing the content of a transceiver register
Icu_DisableNotification	Icu.h	This function disables the notification of a channel.
Icu_EnableNotification	Icu.h	This function enables the notification on the given channel.

⌋

8.5.3 Configurable interfaces

This chapter lists all interfaces with configurable target functions. The target function is usually a callback function. The function names are configurable.

[SWS_EthTrcv_00144] Definition of configurable interface <EthTrcvWakeUpCallout> ⌈

Service Name	<EthTrcvWakeUpCallout>	

Syntax	`void <EthTrcvWakeUpCallout> (`
	`  uint8 TrcvIdx,`
	`  Eth_ModeType TrcvMode`
	`)`	

Service ID [hex]	0x29	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant Dont care	
Parameters (in)	TrcvIdx	Index of the Ethernet Transceiver
	TrcvMode	ETH_MODE_ACTIVE: enable the transceiver ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST: enable the transceiver and request to trigger a wake-up on the network, if the used Ethernet PHY support such a feature. E.g. used for PHYs which are compliant to OA TC10.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Indicates an wake-up request for the specified Ethernet Transceiver. The given TrcvMode indicates if the request is based on a remote request from the network (ETH_MODE_ACTIVE) or if the request is based on active user request (ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST). The callback function Ccan be used to trigger integrator code that initiates a remote wake-up .(e.g. trigger a wake-up line for an active user request).	
Available via	EthTrcv_Externals.h	

⌋

[SWS_EthTrcv_00145] ⌈The callback function shall be configurable by the configuration parameter: EthTrcvWakeUpCallout.⌋

8.6 InterruptServiceRoutines

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

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 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 EthTrcv.

Chapter 10.3 specifies published information of the module EthTrcv.

10.1 How to read this chapter

For details refer to *SWS BSWGeneral, chapter 10.1 "Introduction to configuration specification"* [4].

10.2 Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters describe Chapters 1 and Chapter 7.

[SWS_EthTrcv_00155] ⌈The Ethernet Transceiver Driver module shall reject configurations with partition mappings which are not supported by the implementation.⌋

10.2.1 EthTrcv

[ECUC_EthTrcv_00034] Definition of EcucModuleDef EthTrcv ⌈

Module Name	EthTrcv
Description	Configuration of Ethernet Transceiver Driver module
Post-Build Variant Support	true
Supported Config Variants	VARIANT-LINK-TIME, VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthTrcvConfigSet	1	This container contains the configuration parameters and sub containers of the AUTOSAR EthTrcv module.
EthTrcvGeneral	1	General configuration of Ethernet Transceiver Driver module

⌋

10.2.2 EthTrcvConfigSet

[ECUC_EthTrcv_00016] Definition of EcucParamConfContainerDef EthTrcvConfigSet ⌈

Container Name	EthTrcvConfigSet
Parent Container	EthTrcv
Description	This container contains the configuration parameters and sub containers of the AUTOSAR EthTrcv module.
Configuration Parameters	

No Included Parameters

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

⌋

10.2.3 EthTrcvConfig

[ECUC_EthTrcv_00012] Definition of EcucParamConfContainerDef EthTrcvConfig ⌈

Container Name	EthTrcvConfig
Parent Container	EthTrcvConfigSet
Description	Configuration of the individual transceiver
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTrcvActAsSlavePassiveEnabled	0..1	[ECUC_EthTrcv_00071]
EthTrcvConnNeg	1	[ECUC_EthTrcv_00025]
EthTrcvDuplexMode	1	[ECUC_EthTrcv_00023]
EthTrcvEnablePLCA	1	[ECUC_EthTrcv_00055]
EthTrcvForceSleepEnabled	1	[ECUC_EthTrcv_00074]
EthTrcvIdx	1	[ECUC_EthTrcv_00013]
EthTrcvMacLayerSpeed	0..1	[ECUC_EthTrcv_00044]
EthTrcvMacLayerSubType	0..1	[ECUC_EthTrcv_00043]
EthTrcvMacLayerType	0..1	[ECUC_EthTrcv_00035]
EthTrcvPhysLayerPlcaLocalNodeId	0..1	[ECUC_EthTrcv_00056]
EthTrcvPhysLayerPlcaMaxBurstCount	0..1	[ECUC_EthTrcv_00059]
EthTrcvPhysLayerPlcaMaxBurstTimer	0..1	[ECUC_EthTrcv_00060]

▽

△

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTrcvPhysLayerPlcaNodeCount	0..1	[ECUC_EthTrcv_00058]
EthTrcvPhysLayerPlcaTransmitOpportunityTimer	0..1	[ECUC_EthTrcv_00057]
EthTrcvPhysLayerType	0..1	[ECUC_EthTrcv_00024]
EthTrcvSleepModeExecutionDelay	0..1	[ECUC_EthTrcv_00063]
EthTrcvSleepRequestNumberOfRepetitions	1	[ECUC_EthTrcv_00072]
EthTrcvSleepRequestRepetitionPeriod	1	[ECUC_EthTrcv_00073]
EthTrcvSpeed	1	[ECUC_EthTrcv_00022]
EthTrcvWakeUpCallout	0..1	[ECUC_EthTrcv_00028]
EthTrcvWakeupForwardLocalEnabled	1	[ECUC_EthTrcv_00067]
EthTrcvWakeupForwardRemoteEnabled	1	[ECUC_EthTrcv_00068]
EthTrcvWakeupLocalDetectionTime	0..1	[ECUC_EthTrcv_00069]
EthTrcvWakeupLocalDurationTime	0..1	[ECUC_EthTrcv_00070]
EthTrcvWakeupLocalEnabled	1	[ECUC_EthTrcv_00065]
EthTrcvWakeupRemoteEnabled	1	[ECUC_EthTrcv_00066]
EthTrcvWakeupRequestNumberOfRepetitions	1	[ECUC_EthTrcv_00075]
EthTrcvWakeUpRequestRepetitionPeriod	0..1	[ECUC_EthTrcv_00062]
EthTrcvWakeupSleepOnDatalineEnabled	0..1	[ECUC_EthTrcv_00064]
EthTrcvConfigEcucPartitionRef	0..1	[ECUC_EthTrcv_00051]
EthTrcvIcuChannelRef	0..1	[ECUC_EthTrcv_00026]

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_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.
EthTrcvMgmtInterface	0..1	The choice container allow to configure either the EthTrcv is accessed by a MII interface or Switch interface.
EthTrcvWakeupMap	0..7	Container for the mapping of wake up reasons to wake up sources. At least one container is needed if EthTrcvWakeUp Support is not ETHTRCV_WAKEUP_NOT_SUPPORTED.

⌋

[ECUC_EthTrcv_00071] Definition of EcucBooleanParamDef EthTrcvActAsSlavePassiveEnabled

Status: DRAFT

⌈

Parameter Name	EthTrcvActAsSlavePassiveEnabled		
Parent Container	EthTrcvConfig		
Description	Specifies if the ECU is acting as a passive communication slave on the corresponding ComM channel (corresponding ComM channel has ComMNMVariant set to SLAVE_PASSIVE). If the parameter is set to TRUE, the Ethernet transceiver driver shall poll the maintained Ethernet hardware for a signaled Sleep.Indication (according to OA TC10) in the context of the EthTrcv_MainFunction. **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: This parameter is only valid, if EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE and the corresponding ComM channel has set ComMNMVariant to SLAVE_PASSIVE.		

⌋

[ECUC_EthTrcv_00025] Definition of EcucEnumerationParamDef EthTrcvConnNeg ⌈

Parameter Name	EthTrcvConnNeg		
Parent Container	EthTrcvConfig		
Description	Specifies the connection negotiation of the Ethernet transceiver link.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	TRCV_CONN_NEG_AUTO	Automatic Negotiation	
	TRCV_CONN_NEG_MASTER	Master	
	TRCV_CONN_NEG_NONE	PLCA	
	TRCV_CONN_NEG_SLAVE	Slave	
Post-Build Variant Value	true		
Value Configuration Class	**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: If EthTrcvEnablePLCA is TRUE this parameter shall be set to TRCV_CONN_NEG_NONE. If EthTrcvEnablePLCA is FALSE and EthTrcvPhysLayerType is set to TRCV_PHYS_LAYER_TYPE_10BASE_T1S this parameter shall be configured (master or slave). Note: 10BASE-T1S can be used with PLCA or CSMA/CD media access.

⌋

[ECUC_EthTrcv_00023] Definition of EcucEnumerationParamDef EthTrcvDuplexMode ⌈

Parameter Name	EthTrcvDuplexMode		
Parent Container	EthTrcvConfig		
Description	Specifies the duplex mode of the Ethernet transceiver link if Auto-Negotiation is disabled. This parameter is ignored if Auto-Negotiation is enabled (EthTrcvConnNeg=TRCV_CONN_NEG_AUTO).		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	ETHTRCV_DUPLEX_MODE_FULL	Full duplex.	
	ETHTRCV_DUPLEX_MODE_HALF	Half duplex.	
Post-Build Variant Value	true		
Value Configuration Class	**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: EthTrcvConnNeg=TRCV_CONN_NEG_AUTO		

⌋

[ECUC_EthTrcv_00055] Definition of EcucBooleanParamDef EthTrcvEnablePLCA ⌈

Parameter Name	EthTrcvEnablePLCA		
Parent Container	EthTrcvConfig		
Description	Enables the transmission with PLCA (Physical Layer Collision Avoidance) TRUE: PLCA enabled FALSE: PLCA disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	

▽

△

Scope / Dependency	scope: local dependency: This parameter is relevant in case that EthTrcvPhysLayerType = TRCV_PHYS_LAYER_TYPE_10BASE_T1S. If PLCA mode is enabled via the parameter EthTrcvEnablePLCA the following parameter must be configured: * EthTrcvPhysLayerPlcaLocalNodeID, * EthTrcvPhysLayerPlcaTransmitOpportunityTime, * EthTrcvPhysLayerPlcaNodeCount, * EthTrcvPhysLayerPlcaMaxBurstTime, * EthTrcvPhysLayerPlcaMaxBurstCount.

⌋

[ECUC_EthTrcv_00074] Definition of EcucBooleanParamDef EthTrcvForceSleepEnabled ⌈

Parameter Name	EthTrcvForceSleepEnabled		
Parent Container	EthTrcvConfig		
Description	Specifies if the Ethernet hardware (PHY) support to go to sleep without the confirmation of the Ethernet hardware (PHY) of the linked Ethernet communication partner according to the OA TC10. If the parameter is set to TRUE, the Ethernet hardware (PHY) support to force its hardware state to the sleep state. This is used, if the Ethernet communication partner do not accept the Sleep.Request after the specified repetitions on Sleep.Request (see EthTrcvSleepRequestNumberOfRepetitions). If EthTrcvSleepRequestNumberOfRepetitions is not available, the EthTransceiver driver shall force the Ethernet hardware (PHY) to sleep state after the first unsuccessful Sleep.Request.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: This parameter is only valid, if EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE.		

⌋

[ECUC_EthTrcv_00013] Definition of EcucIntegerParamDef EthTrcvIdx ⌈

Parameter Name	EthTrcvIdx		
Parent Container	EthTrcvConfig		
Description	Specifies the instance ID of the configured transceiver.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 255		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants

▽

△

	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU withAuto = true		

⌋

[ECUC_EthTrcv_00044] Definition of EcucEnumerationParamDef EthTrcvMacLayerSpeed ⌈

Parameter Name	EthTrcvMacLayerSpeed		
Parent Container	EthTrcvConfig		
Description	Defines the baud rate of the MAC layer.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	ETH_MAC_LAYER_SPEED_100M	–	
	ETH_MAC_LAYER_SPEED_10G	–	
	ETH_MAC_LAYER_SPEED_10M	–	
	ETH_MAC_LAYER_SPEED_1G	–	
	ETH_MAC_LAYER_SPEED_2500M	–	
	ETH_MAC_LAYER_SPEED_5G	–	
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: ECU		

⌋

[ECUC_EthTrcv_00043] Definition of EcucEnumerationParamDef EthTrcvMacLayerSubType ⌈

Parameter Name	EthTrcvMacLayerSubType	
Parent Container	EthTrcvConfig	
Description	Defines the MAC layer subtype of a switch port	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	LIGHT	–

▽

△

	REDUCED	–	
	REVERSED	–	
	SERIAL	–	
	STANDARD	–	
	UNIVERSAL_SERIAL	–	
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: ECU		

⌋

[ECUC_EthTrcv_00035] Definition of EcucEnumerationParamDef EthTrcvMacLayerType ⌈

Parameter Name	EthTrcvMacLayerType		
Parent Container	EthTrcvConfig		
Description	Defines the MAC layer type of the ethernet transceiver.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	TRCV_MAC_LAYER_TYPE_XGMII	MAC layer interface (data) bandwith class 1Gbit/s (e.g. GMII, RGMII, SGMII, RvGMII, USGMII)	
	TRCV_MAC_LAYER_TYPE_XMII	MAC layer interface (data) bandwith class 10-100Mbit/s (e.g. RMII, RvMII, SMII, MII)	
	TRCV_MAC_LAYER_TYPE_XXGMII	MAC layer interface (data) bandwith class 10Gbit/s	
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

⌋

[ECUC_EthTrcv_00056] Definition of EcucIntegerParamDef EthTrcvPhysLayerPlcaLocalNodeId ⌈

Parameter Name	EthTrcvPhysLayerPlcaLocalNodeId		
Parent Container	EthTrcvConfig		
Description	Configuration parameter for the transceiver node ID when the PLCA mode for 10BASE-T1S is used.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	255		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: This parameter is only relevant in case that EthTrcvEnablePLCA is set to true. If the value is set to 0, the transceiver is configured as head-node. Note: Within a PLCA cluster the head-node ID shall be unique. A head-node schedules the transmission cycles and the node ID at the client level defines the transmission order.		

⌋

[ECUC_EthTrcv_00059] Definition of EcucIntegerParamDef EthTrcvPhysLayerPlcaMaxBurstCount ⌈

Parameter Name	EthTrcvPhysLayerPlcaMaxBurstCount		
Parent Container	EthTrcvConfig		
Description	Defines maximum packets allowed to be transmitted within a TO. The maximum burst count could differ per ECU within a PLCA mixed segment.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: This parameter is only relevant in case that EthTrcvEnablePLCA is set to true. If EthTrcvPhysLayerPlcaMaxBurstCount is set to 0 (default value), only one ethernet frame per TO shall be transmitted.		

⌋

[ECUC_EthTrcv_00060] Definition of EcucIntegerParamDef EthTrcvPhysLayerPlcaMaxBurstTimer ⌈

Parameter Name	EthTrcvPhysLayerPlcaMaxBurstTimer		
Parent Container	EthTrcvConfig		
Description	Limits the burst frames in bit time. The maximum burst time could differ per ECU within a PLCA mixed segment.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	128		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: This parameter is only relevant in case that EthTrcvEnablePLCA is set to true. Note: For PLCA burst mode to work properly this time should be set greater than one IPG.		

⌋

[ECUC_EthTrcv_00058] Definition of EcucIntegerParamDef EthTrcvPhysLayerPlcaNodeCount ⌈

Parameter Name	EthTrcvPhysLayerPlcaNodeCount		
Parent Container	EthTrcvConfig		
Description	Defines the number of communication participants on the mixed segment. This value is relevant for the head-node transceiver (EthTrcvPhysLayerNodeID = 0) in order to know when the BEACON has to be transmitted.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	8		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: This parameter is only relevant in case that EthTrcvEnablePLCA is set to true.		

⌋

[ECUC_EthTrcv_00057] Definition of EcucIntegerParamDef EthTrcvPhysLayerPlcaTransmitOpportunityTimer ⌈

Parameter Name	EthTrcvPhysLayerPlcaTransmitOpportunityTimer		
Parent Container	EthTrcvConfig		
Description	Timer for the transmission in bit time to evaluate if a Transmission Opportunity is yield or not.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 255		
Default value	32		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: This parameter is only relevant in case that EthTrcvEnablePLCA is set to true. This parameter shall be set identical within a PLCA cluster.		

⌋

[ECUC_EthTrcv_00024] Definition of EcucEnumerationParamDef EthTrcvPhysLayerType ⌈

Parameter Name	EthTrcvPhysLayerType	
Parent Container	EthTrcvConfig	
Description	Specifies the physical layer type of the Ethernet transceiver link.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	TRCV_PHYS_LAYER_TYPE_10000BASE_T1	physical layer interface 10GBASE-T1 (10Gbit/s, 1pair). Used for automotive.
	TRCV_PHYS_LAYER_TYPE_1000BASE_T	physical layer interface 1000BASE-T (1Gbit/s, 4 pairs). Used for consumer electronic.
	TRCV_PHYS_LAYER_TYPE_1000BASE_T1	physical layer interface 1000BASE-T1 (1Gbit/s, 1 pair). Used for automotive.
	TRCV_PHYS_LAYER_TYPE_100BASE_T1	physical layer interface 100BASE-T1 (100Mbit/s, 1 pair). Used for automotive.
	TRCV_PHYS_LAYER_TYPE_100BASE_TX	physical layer interface 100BASE-TX (100Mbit/s, 2 pairs). Used for consumer electronic.
	TRCV_PHYS_LAYER_TYPE_10BASE_T1S	Physical layer interface 10BASE-T1S (10Mbit/s, 1 pair). Used for automotive.
	TRCV_PHYS_LAYER_TYPE_2500BASE_T1	physical layer interface 2.5GBASE-T1 (2.5Gbit/s, 1pair). Used for automotive.
	TRCV_PHYS_LAYER_TYPE_5000BASE_T1	physical layer interface 5GBASE-T1 (5Gbit/s, 1pair). Used for automotive.
Post-Build Variant Multiplicity	true	

▽

△

Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00063] Definition of EcucFloatParamDef EthTrcvSleepModeExecutionDelay ⌈

Parameter Name	EthTrcvSleepModeExecutionDelay		
Parent Container	EthTrcvConfig		
Description	Specifies the time delay in seconds to execute a sleep (see OA TC10) for a Ethernet hardware (PHY), if a pending wake-up was detected while a ETH_MODE_DOWN was requested. The value shall be an integral multiple of EthTrcvMainFunctionPeriod.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	]0 .. INF[		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: This parameter is only valid and mandatory, if EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE.		

⌋

[ECUC_EthTrcv_00072] Definition of EcucIntegerParamDef EthTrcvSleepRequestNumberOfRepetitions ⌈

Parameter Name	EthTrcvSleepRequestNumberOfRepetitions
Parent Container	EthTrcvConfig
Description	Specifies the repetitions to trigger a Sleep.Request (according to OA TC10) if a release of the communication channel was triggered by the upper layer (ETH_MODE_DOWN) and a SleepFail.Indication was signaled. Thus, the Ethernet hardware (PHY) of the connected communication partner did not accept the Sleep.Request.

▽

△

Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	0		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: This parameter is only valid, if EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE.		

⌋

[ECUC_EthTrcv_00073] Definition of EcucFloatParamDef EthTrcvSleepRequestRepetitionPeriod ⌈

Parameter Name	EthTrcvSleepRequestRepetitionPeriod		
Parent Container	EthTrcvConfig		
Description	Specifies the repetition period in seconds of repetitions for a Sleep Request (according to OA TC10). The value shall be a integral multiple of EthTrcvMainFunctionPeriod.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	]0 .. INF[		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: This parameter is only valid, if EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE and EthTrcvSleepRequestNumberOfRepetitions is set greater than 0.		

⌋

[ECUC_EthTrcv_00022] Definition of EcucEnumerationParamDef EthTrcvSpeed ⌈

Parameter Name	EthTrcvSpeed
Parent Container	EthTrcvConfig
Description	Specifies the speed of the Ethernet transceiver link in [MBit/s]. If AutoNegotiation is enabled (EthTrcvConnNeg=TRCV_CONN_NEG_AUTO) 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	
	TRCV_SPEED_10000	10000 MBit/s	
	TRCV_SPEED_2500	2500 MBit/s	
	TRCV_SPEED_5000	5000 MBit/s	
Post-Build Variant Value	true		
Value Configuration Class	**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: EthTrcvConnNeg=TRCV_CONN_NEG_AUTO		

⌋

[ECUC_EthTrcv_00028] Definition of EcucFunctionNameDef EthTrcvWakeUpCallout ⌈

Parameter Name	EthTrcvWakeUpCallout		
Parent Container	EthTrcvConfig		
Description	Configuration of the call-out name.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default value	–		
Regular Expression	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	dependency: Only valid if EthTrcvWakeUpSupport is not ETHTRCV_WAKEUP_NOT_SUPPORTED.		

⌋

[ECUC_EthTrcv_00067] Definition of EcucBooleanParamDef EthTrcvWakeupForwardLocalEnabled ⌈

Parameter Name	EthTrcvWakeupForwardLocalEnabled		
Parent Container	EthTrcvConfig		
Description	Specifies if remote wake up forwarding is enabled (TRUE) or disabled (FALSE) for OA TC10 compliant Ethernet Transceiver. If the parameter is set to TRUE, the Ethernet hardware (PHY) activate a local wake up (e.g. via I/O pin), if a remote wake-up on data line (e.g. 100Base-T1) was received (either WUP or WUR).		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: This parameter is only valid, if EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE and if EthTrcvWakeupLocalEnabled is set to TRUE.		

⌋

[ECUC_EthTrcv_00068] Definition of EcucBooleanParamDef EthTrcvWakeupForwardRemoteEnabled ⌈

Parameter Name	EthTrcvWakeupForwardRemoteEnabled		
Parent Container	EthTrcvConfig		
Description	Specifies if local wake up forwarding is enabled (TRUE) or disabled (FALSE) for OA TC10 compliant Ethernet Transceiver. If the parameter is set to TRUE, the Ethernet hardware (PHY) transmit a wake-up (WUP or WUR on the data line (e.g. 100Base-T1), when a local wake-up occurred (e.g. via I/O pin), which was triggered by neighboring Ethernet hardware (PHY).		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: This parameter is only valid, if EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE and if EthTrcvWakeupRemoteEnabled set to TRUE.		

⌋

[ECUC_EthTrcv_00069] Definition of EcucFloatParamDef EthTrcvWakeupLocalDetectionTime ⌈

Parameter Name	EthTrcvWakeupLocalDetectionTime		
Parent Container	EthTrcvConfig		
Description	Defines the time in seconds when a local wake-up (e.g. via I/O pin) triggered by a neighboring PHY is evalutated as a valid wake-up.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[1E-5 .. 0.1]		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local dependency: This parameter is only valid, if EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE.		

⌋

[ECUC_EthTrcv_00070] Definition of EcucFloatParamDef EthTrcvWakeupLocalDurationTime ⌈

Parameter Name	EthTrcvWakeupLocalDurationTime		
Parent Container	EthTrcvConfig		
Description	Defines the duration time in seconds how long a local wake-up should be present on the local wake-up connection (e.g. via I/O pin) to indicate the neighboring PHYs about a wake-up.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[4E-5 .. 0.1]		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	

▽

△

Scope / Dependency	scope: local dependency: This parameter is only valid, if EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE. This parameter has to be configured, if EthTrcvWakeupForwardLocalEnabled is set to TRUE

⌋

[ECUC_EthTrcv_00065] Definition of EcucBooleanParamDef EthTrcvWakeupLocalEnabled ⌈

Parameter Name	EthTrcvWakeupLocalEnabled		
Parent Container	EthTrcvConfig		
Description	Specifies if local wake-up is enabled (TRUE) or disabled (FALSE) for OA TC10 compliant Ethernet hardware (PHY). If the parameter is set to TRUE, the Ethernet hardware (PHY) is able to detect and react on a wake-up received by a neighboring PHY (e.g. via I/O pin).		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: This parameter is only valid, if EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE.		

⌋

[ECUC_EthTrcv_00066] Definition of EcucBooleanParamDef EthTrcvWakeupRemoteEnabled ⌈

Parameter Name	EthTrcvWakeupRemoteEnabled		
Parent Container	EthTrcvConfig		
Description	Specifies if remote wake-up is enabled (TRUE) or disabled (FALSE) for OA TC10 compliant Ethernet hardware (PHY). If the parameter is set to TRUE, the Ethernet hardware (PHY) wake up when receiving a remote wake-up (e.g. via 100Base-T1 data line) triggered by the connected communication partner.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	

▽

△

Scope / Dependency	scope: local dependency: This parameter is only valid, if EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE.		

⌋

[ECUC_EthTrcv_00075] Definition of EcucIntegerParamDef EthTrcvWakeupRequestNumberOfRepetitions ⌈

Parameter Name	EthTrcvWakeupRequestNumberOfRepetitions		
Parent Container	EthTrcvConfig		
Description	Specifies the repetitions to trigger a wake-up request (according to OA TC10), if an active communication request (ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST) was triggered by the upperlayer.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	0		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: This parameter is only valid, if EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE.		

⌋

[ECUC_EthTrcv_00062] Definition of EcucFloatParamDef EthTrcvWakeUpRequestRepetitionPeriod ⌈

Parameter Name	EthTrcvWakeUpRequestRepetitionPeriod		
Parent Container	EthTrcvConfig		
Description	Specifies the repetition period in seconds of a wake-up request, if an active communication request (ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST) was triggered by the upperlayer. The value shall be an integral multiple of EthTrcvMainFunctionPeriod.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	]0 .. INF[		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants

▽

△

	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: This parameter is only valid, if EthTrcvWakeupSleepOnDatalineEnabled is set to TRUE and EthTrcvSleepRequestNumberOfRepetitions is set greater than 0.		

⌋

[ECUC_EthTrcv_00064] Definition of EcucBooleanParamDef EthTrcvWakeupSleepOnDatalineEnabled ⌈

Parameter Name	EthTrcvWakeupSleepOnDatalineEnabled		
Parent Container	EthTrcvConfig		
Description	Specifies if wake-up on data line according to OA TC10 is supported by the used Ethernet hardware (PHY) and if the functionality (wake-up and sleep on dataline) is enabled. If the parameter is configured, wake-up and sleep on data line is supported according to OA TC10 by the used Ethernet hardware (PHY). If the parameter is set to TRUE, the wake-up and sleep on data line functionality according to OA TC10 is enabled. If the parameter is set to FALSE, the wake-up and sleep on data line functionality according to OA TC10 is disabled. Note: Disabling of the wake-up and sleep functionality is used if Ethernet hardware (PHY) parts are connected, where one PHY do not support the OA TC10 wake-up and sleep on dataline functionality. This may needed for legacy scenarios, where ECUs are taken over from previous car lines and no OA TC10 Ethernet hardware (PHY) were used.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00051] Definition of EcucReferenceDef EthTrcvConfigEcucPartitionRef ⌈

Parameter Name	EthTrcvConfigEcucPartitionRef		
Parent Container	EthTrcvConfig		
Description	Maps the Ethernet transceiver configuration to zero or one ECUC partitions. The ECUC partition referenced is a subset of the ECUC partitions where the Ethernet transceiver driver is mapped to.		
Multiplicity	0..1		
Type	Reference to EcucPartition		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

⌋

[ECUC_EthTrcv_00026] Definition of EcucReferenceDef EthTrcvIcuChannelRef ⌈

Parameter Name	EthTrcvIcuChannelRef		
Parent Container	EthTrcvConfig		
Description	Reference to the IcuChannel to enable/disable the interrupts for wakeups.		
Multiplicity	0..1		
Type	Symbolic name reference to IcuChannel		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

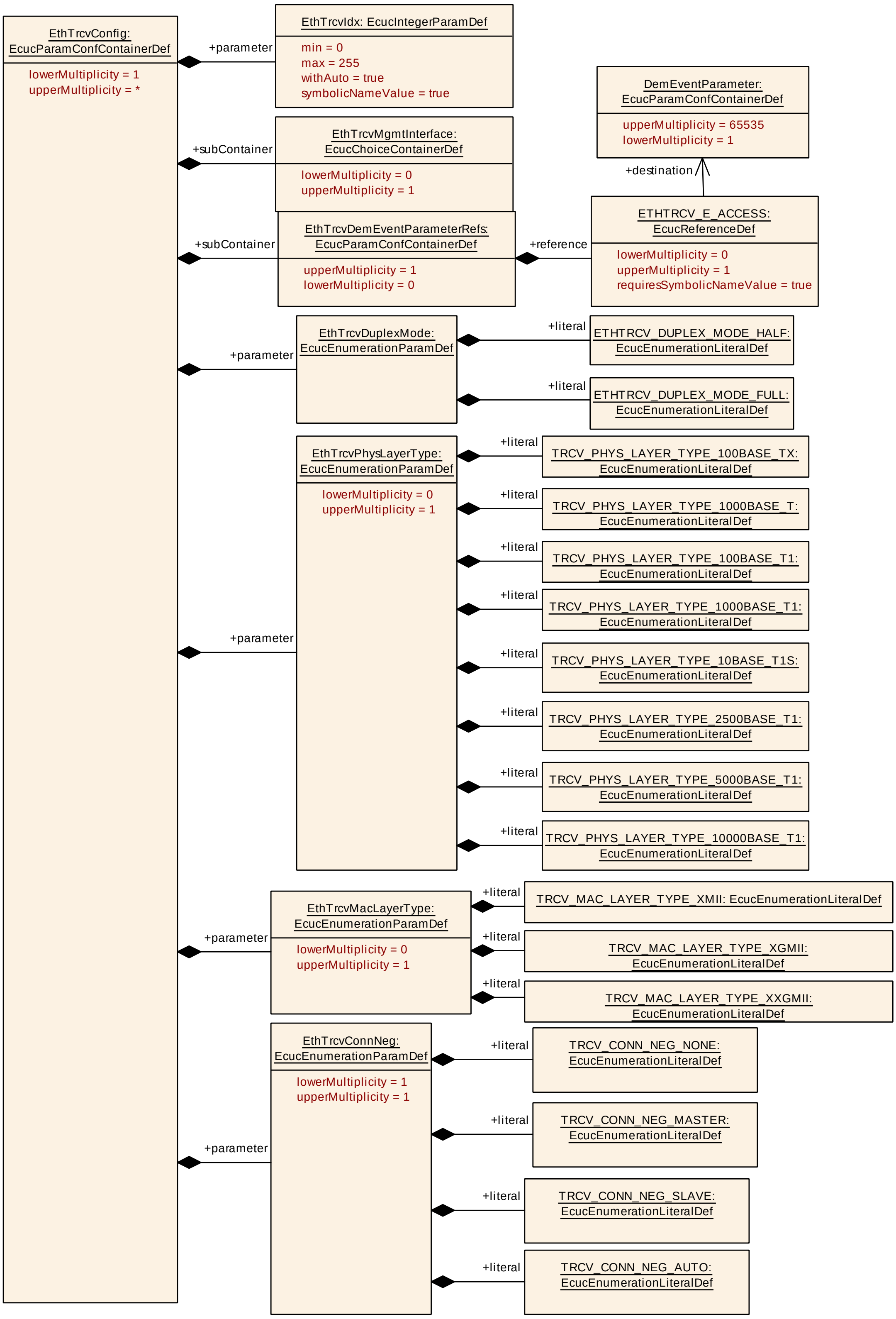
EthTrcvConfig: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+parameter
EthTrcvIdx: EcucIntegerParamDef
min = 0
max = 255
withAuto = true
symbolicNameValue = true
+subContainer
EthTrcvMgmtInterface: EcucChoiceContainerDef
lowerMultiplicity = 0
upperMultiplicity = 1
DemEventParameter: EcucParamConfContainerDef
upperMultiplicity = 65535
lowerMultiplicity = 1
+destination
+subContainer
EthTrcvDemEventParameterRefs: EcucParamConfContainerDef
upperMultiplicity = 1
lowerMultiplicity = 0
+reference
ETHTRCV_E_ACCESS: EcucReferenceDef
lowerMultiplicity = 0
upperMultiplicity = 1
requiresSymbolicNameValue = true
+parameter
EthTrcvDuplexMode: EcucEnumerationParamDef
+literal
ETHTRCV_DUPLEX_MODE_HALF: EcucEnumerationLiteralDef
+literal
ETHTRCV_DUPLEX_MODE_FULL: EcucEnumerationLiteralDef
+parameter
EthTrcvPhysLayerType: EcucEnumerationParamDef
lowerMultiplicity = 0
upperMultiplicity = 1
+literal
TRCV_PHYS_LAYER_TYPE_100BASE_TX: EcucEnumerationLiteralDef
+literal
TRCV_PHYS_LAYER_TYPE_1000BASE_T: EcucEnumerationLiteralDef
+literal
TRCV_PHYS_LAYER_TYPE_100BASE_T1: EcucEnumerationLiteralDef
+literal
TRCV_PHYS_LAYER_TYPE_1000BASE_T1: EcucEnumerationLiteralDef
+literal
TRCV_PHYS_LAYER_TYPE_10BASE_T1S: EcucEnumerationLiteralDef
+literal
TRCV_PHYS_LAYER_TYPE_2500BASE_T1: EcucEnumerationLiteralDef
+literal
TRCV_PHYS_LAYER_TYPE_5000BASE_T1: EcucEnumerationLiteralDef
+literal
TRCV_PHYS_LAYER_TYPE_10000BASE_T1: EcucEnumerationLiteralDef
+parameter
EthTrcvMacLayerType: EcucEnumerationParamDef
lowerMultiplicity = 0
upperMultiplicity = 1
+literal
TRCV_MAC_LAYER_TYPE_XMII: EcucEnumerationLiteralDef
+literal
TRCV_MAC_LAYER_TYPE_XGMII: EcucEnumerationLiteralDef
+literal
TRCV_MAC_LAYER_TYPE_XXGMII: EcucEnumerationLiteralDef
+parameter
EthTrcvConnNeg: EcucEnumerationParamDef
lowerMultiplicity = 1
upperMultiplicity = 1
+literal
TRCV_CONN_NEG_NONE: EcucEnumerationLiteralDef
+literal
TRCV_CONN_NEG_MASTER: EcucEnumerationLiteralDef
+literal
TRCV_CONN_NEG_SLAVE: EcucEnumerationLiteralDef
+literal
TRCV_CONN_NEG_AUTO: EcucEnumerationLiteralDef

Figure 10.1: EthTrcvConfig (1/2)

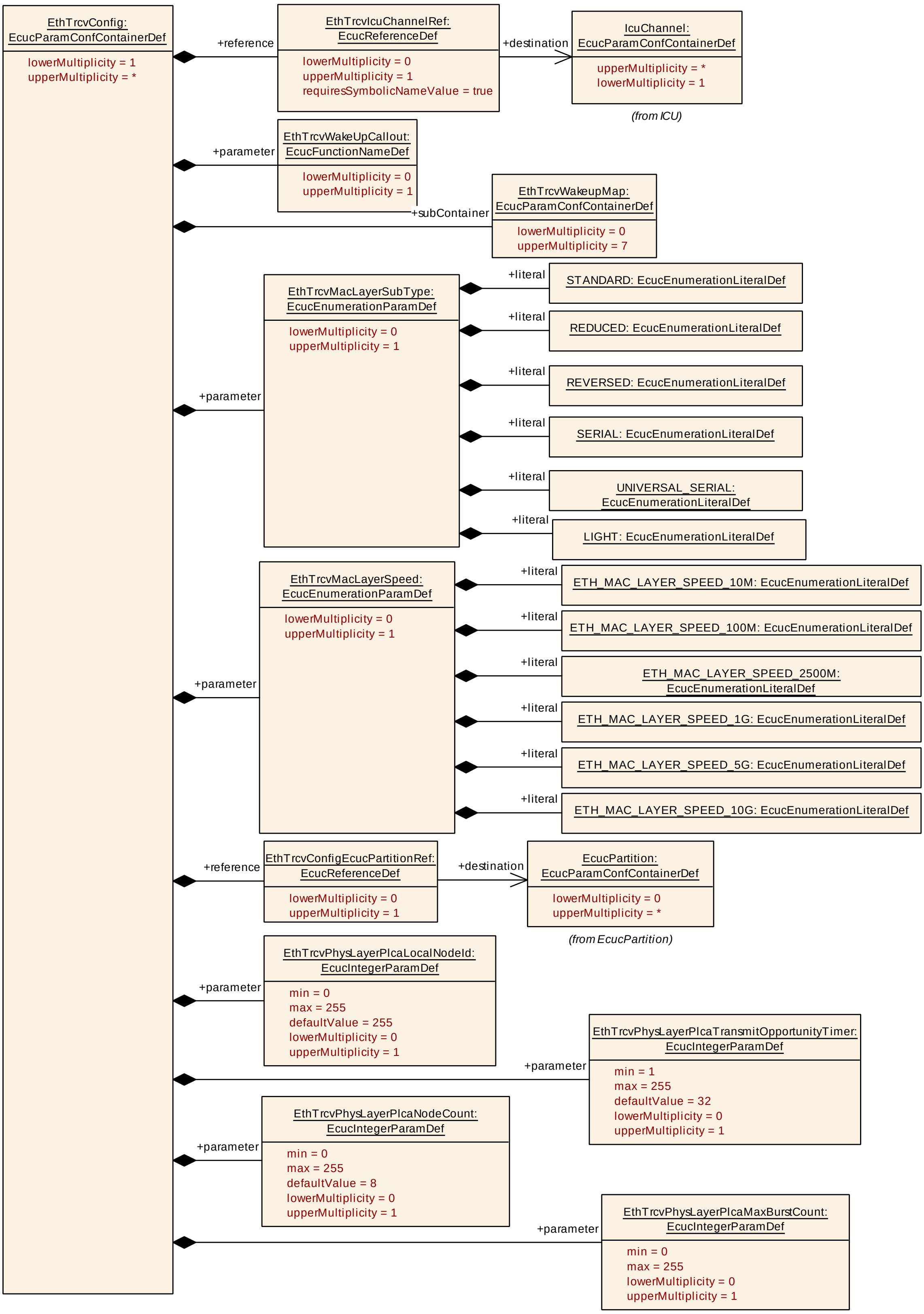
EthTrcvConfig: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+reference
EthTrcvIcuChannelRef: EcucReferenceDef
lowerMultiplicity = 0
upperMultiplicity = 1
requiresSymbolicNameValue = true
+destination
IcuChannel: EcucParamConfContainerDef
upperMultiplicity = *
lowerMultiplicity = 1
(from ICU)
+parameter
EthTrcvWakeUpCallout: EcucFunctionNameDef
lowerMultiplicity = 0
upperMultiplicity = 1
+subContainer
EthTrcvWakeupMap: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 7
+parameter
EthTrcvMacLayerSubType: EcucEnumerationParamDef
lowerMultiplicity = 0
upperMultiplicity = 1
+literal
STANDARD: EcucEnumerationLiteralDef
+literal
REDUCED: EcucEnumerationLiteralDef
+literal
REVERSED: EcucEnumerationLiteralDef
+literal
SERIAL: EcucEnumerationLiteralDef
+literal
UNIVERSAL_SERIAL: EcucEnumerationLiteralDef
+literal
LIGHT: EcucEnumerationLiteralDef
+parameter
EthTrcvMacLayerSpeed: EcucEnumerationParamDef
lowerMultiplicity = 0
upperMultiplicity = 1
+literal
ETH_MAC_LAYER_SPEED_10M: EcucEnumerationLiteralDef
+literal
ETH_MAC_LAYER_SPEED_100M: EcucEnumerationLiteralDef
+literal
ETH_MAC_LAYER_SPEED_2500M: EcucEnumerationLiteralDef
+literal
ETH_MAC_LAYER_SPEED_1G: EcucEnumerationLiteralDef
+literal
ETH_MAC_LAYER_SPEED_5G: EcucEnumerationLiteralDef
+literal
ETH_MAC_LAYER_SPEED_10G: EcucEnumerationLiteralDef
+reference
EthTrcvConfigEcucPartitionRef: EcucReferenceDef
lowerMultiplicity = 0
upperMultiplicity = 1
+destination
EcucPartition: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = *
(from EcucPartition)
+parameter
EthTrcvPhysLayerPlcaLocalNodeId: EcucIntegerParamDef
min = 0
max = 255
defaultValue = 255
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
EthTrcvPhysLayerPlcaTransmitOpportunityTimer: EcucIntegerParamDef
min = 1
max = 255
defaultValue = 32
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
EthTrcvPhysLayerPlcaNodeCount: EcucIntegerParamDef
min = 0
max = 255
defaultValue = 8
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
EthTrcvPhysLayerPlcaMaxBurstCount: EcucIntegerParamDef
min = 0
max = 255
lowerMultiplicity = 0
upperMultiplicity = 1

Figure 10.2: EthTrcvConfig (2/2)

[SWS_EthTrcv_00157] ⌈The ECUC partitions referenced by EthTrcvConfigEcucPartitionRef shall be a subset of the ECUC partitions referenced by EthTrcvEcucPartitionRef.⌋

[SWS_EthTrcv_00158] ⌈EthTrcvConfig, EthCtrlConfig and EthSwtConfig (if existent in configuration) of one communication channel shall all reference the same ECUC partition.⌋

[SWS_EthTrcv_CONSTR_00001] ⌈If EthTrcvEcucPartitionRef references one or more ECUC partitions, EthTrcvConfigEcucPartitionRef shall have a multiplicity of one and reference one of these ECUC partitions as well.⌋

10.2.4 EthTrcvDemEventParameterRefs

[ECUC_EthTrcv_00017] Definition of EcucParamConfContainerDef EthTrcvDemEventParameterRefs ⌈

Container Name	EthTrcvDemEventParameterRefs
Parent Container	EthTrcvConfig
Description	Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The Event Id is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
ETHTRCV_E_ACCESS	0..1	[ECUC_EthTrcv_00018]

No Included Containers

⌋

[ECUC_EthTrcv_00018] Definition of EcucReferenceDef ETHTRCV_E_ACCESS ⌈

Parameter Name	ETHTRCV_E_ACCESS
Parent Container	EthTrcvDemEventParameterRefs
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
Post-Build Variant Multiplicity	true

▽

△

Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

⌋

10.2.5 EthTrcvMgmtInterface

[ECUC_EthTrcv_00036] Definition of EcucChoiceContainerDef EthTrcvMgmtInterface ⌈

Choice Container Name	EthTrcvMgmtInterface		
Parent Container	EthTrcvConfig		
Description	The choice container allow to configure either the EthTrcv is accessed by a MII interface or Switch interface.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	

No Included Parameters

Container Choices		
Container Name	**Multiplicity**	**Scope / Dependency**
EthTrcvMiiInterface	0..1	This container includes the MII interface configuration between an Ethernet Controller and the Ethernet Transceiver. If this container is configured the EthTrcv shall call Eth_WriteMii / Eth_ReadMii API to access the hardware ethernet tranceiver. Note: In case of external MACPHY use this parameter may be enabled.
EthTrcvSwitchInterface	0..1	This container includes the Switch interface configuration between an Ethernet Switch and an Ethernet Transceiver. If this container is configured the EthTrcv shall call EthSwt_WriteTrcvRegister / EthSwt_WriteTrcvRegister API to access the hardware ethernet transceiver.

⌋

10.2.5.1 EthTrcvMiiInterface

[ECUC_EthTrcv_00037] Definition of EcucParamConfContainerDef EthTrcvMiiInterface ⌈

Container Name	EthTrcvMiiInterface		
Parent Container	EthTrcvMgmtInterface		
Description	This container includes the MII interface configuration between an Ethernet Controller and the Ethernet Transceiver. If this container is configured the EthTrcv shall call Eth_WriteMii / Eth_ReadMii API to access the hardware ethernet tranceiver. Note: In case of external MACPHY use this parameter may be enabled.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTrcvCtrlIdx	1	[ECUC_EthTrcv_00014]
EthTrcvMiiIdx	1	[ECUC_EthTrcv_00038]

No Included Containers

⌋

[ECUC_EthTrcv_00014] Definition of EcucIntegerParamDef EthTrcvCtrlIdx ⌈

Parameter Name	EthTrcvCtrlIdx		
Parent Container	EthTrcvMiiInterface		
Description	Specifies the controller used for MII access to the transceiver		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	–		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00038] Definition of EcucIntegerParamDef EthTrcvMiiIdx ⌈

Parameter Name	EthTrcvMiiIdx		
Parent Container	EthTrcvMiiInterface		
Description	Specifies the transceiver index used for MII access to the transceiver.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

10.2.5.2 EthTrcvSwitchInterface

[ECUC_EthTrcv_00040] Definition of EcucParamConfContainerDef EthTrcvSwitchInterface ⌈

Container Name	EthTrcvSwitchInterface		
Parent Container	EthTrcvMgmtInterface		
Description	This container includes the Switch interface configuration between an Ethernet Switch and an Ethernet Transceiver. If this container is configured the EthTrcv shall call EthSwt_WriteTrcvRegister / EthSwt_WriteTrcvRegister API to access the hardware ethernet transceiver.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTrcvSwitchPortRef	1	[ECUC_EthTrcv_00042]
EthTrcvSwitchRef	1	[ECUC_EthTrcv_00041]

No Included Containers

⌋

[ECUC_EthTrcv_00042] Definition of EcucReferenceDef EthTrcvSwitchPortRef ⌈

Parameter Name	EthTrcvSwitchPortRef		
Parent Container	EthTrcvSwitchInterface		
Description	Reference to a switch port.		
Multiplicity	1		
Type	Symbolic name reference to EthSwtPort		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00041] Definition of EcucReferenceDef EthTrcvSwitchRef ⌈

Parameter Name	EthTrcvSwitchRef		
Parent Container	EthTrcvSwitchInterface		
Description	Reference to a switch configuration container.		
Multiplicity	1		
Type	Symbolic name reference to EthSwtConfig		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[SWS_EthTrcv_00208]

Status: DRAFT

⌈The EthTrcv shall collect the information of the EthSwt Ports referring to the each EthTrcv instance (e.g. TrcvIdx) during configuration in order to call the EthSwt callbacks of MKA (e.g. `EthSwt_MacSecUpdateSecYNotification`, `EthSwt_MacSecAddTxSaNotification`, `EthSwt_MacSecAddRxSaNotification` and `EthSwt_MacSecGetMacSecStatisticsNotification`). The EthSwtPort.EthSwtPortTrcvRef links an EthSwtPort to an EthTrcvIdx in the EthSwt configuration.⌋

10.2.6 EthTrcvWakeupMap

[ECUC_EthTrcv_00027] Definition of EcucParamConfContainerDef EthTrcvWakeupMap ⌈

Container Name	EthTrcvWakeupMap
Parent Container	EthTrcvConfig
Description	Container for the mapping of wake up reasons to wake up sources. At least one container is needed if EthTrcvWakeUpSupport is not ETHTRCV_WAKEUP_NOT_SUPPORTED.
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTrcvWakeupReason	1	[ECUC_EthTrcv_00033]
EthTrcvWakeupSourceRef	1	[ECUC_EthTrcv_00029]

No Included Containers

⌋

[ECUC_EthTrcv_00033] Definition of EcucEnumerationParamDef EthTrcvWakeupReason ⌈

Parameter Name	EthTrcvWakeupReason		
Parent Container	EthTrcvWakeupMap		
Description	This parameter defines the transceiver wake up reasons.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	ETHTRCV_WUR_BUS	0x02: Bus wake up detected. Available if supported by hardware.	
	ETHTRCV_WUR_GENERAL	0x01: General wake up detected, no distinct reason supported by hardware.	
	ETHTRCV_WUR_INTERNAL	0x03: Internal wake up detected. Available if supported by hardware.	
	ETHTRCV_WUR_PIN	0x06: Pin wake up detected. Available if supported by hardware.	
	ETHTRCV_WUR_POWER_ON	0x05: Power on wake up detected. Available if supported by hardware.	
	ETHTRCV_WUR_RESET	0x04: Reset wake up detected. Available if supported by hardware.	
	ETHTRCV_WUR_SYSERR	0x07: System error wake up detected. Available if supported by hardware.	
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00029] Definition of EcucReferenceDef EthTrcvWakeupSourceRef ⌈

Parameter Name	EthTrcvWakeupSourceRef		
Parent Container	EthTrcvWakeupMap		
Description	Configures the wake-up source defined in EcuM.		
Multiplicity	1		
Type	Symbolic name reference to EcuMWakeupSource		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

10.2.7 EthTrcvGeneral

[ECUC_EthTrcv_00001] Definition of EcucParamConfContainerDef EthTrcvGeneral ⌈

Container Name	EthTrcvGeneral
Parent Container	EthTrcv
Description	General configuration of Ethernet Transceiver Driver module
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTrcvDevErrorDetect	1	[ECUC_EthTrcv_00003]
EthTrcvEnableCableDiagnosticApi	1	[ECUC_EthTrcv_00054]
EthTrcvGetBaudRateApi	1	[ECUC_EthTrcv_00010]
EthTrcvGetBusWuReasonApi	1	[ECUC_EthTrcv_00061]
EthTrcvGetDuplexModeApi	1	[ECUC_EthTrcv_00011]
EthTrcvGetLinkStateApi	1	[ECUC_EthTrcv_00009]
EthTrcvGetPhyIdentifierApi	1	[ECUC_EthTrcv_00046]
EthTrcvGetTransceiverWakeupModeApi	0..1	[ECUC_EthTrcv_00031]
EthTrcvIndex	1	[ECUC_EthTrcv_00020]
EthTrcvMainFunctionPeriod	0..1	[ECUC_EthTrcv_00032]
EthTrcvMaxTrcvsSupported	1	[ECUC_EthTrcv_00002]
EthTrcvSetPhyTestModeApi	1	[ECUC_EthTrcv_00047]
EthTrcvStartAutoNegotiationApi	1	[ECUC_EthTrcv_00008]
EthTrcvVersionInfoApi	1	[ECUC_EthTrcv_00004]
EthTrcvVersionInfoApiMacro	1	[ECUC_EthTrcv_00005]
EthTrcvWakeUpSupport	1	[ECUC_EthTrcv_00030]
EthTrcvEcucPartitionRef	0..*	[ECUC_EthTrcv_00050]

No Included Containers

⌋

[ECUC_EthTrcv_00003] Definition of EcucBooleanParamDef EthTrcvDevErrorDetect ⌈

Parameter Name	EthTrcvDevErrorDetect		
Parent Container	EthTrcvGeneral		
Description	Switches the development error detection and notification on or off. • true: detection and notification is enabled. • false: detection and notification is disabled.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00054] Definition of EcucBooleanParamDef EthTrcvEnableCable DiagnosticApi ⌈

Parameter Name	EthTrcvEnableCableDiagnosticApi		
Parent Container	EthTrcvGeneral		
Description	Enable/disable the APIs for cable diagnostic: EthTrcv_RunCableDiagnostic, EthTrcv_GetCableDiagnosticsResult		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: If EthTrcvPhysLayerType is set to TRCV_PHYS_LAYER_TYPE_10BASE_T1S this parameter shall be set to FALSE.		

⌋

[ECUC_EthTrcv_00010] Definition of EcucBooleanParamDef EthTrcvGetBaudRateApi ⌈

Parameter Name	EthTrcvGetBaudRateApi		
Parent Container	EthTrcvGeneral		
Description	Enables / Disables EthTrcv_GetBaudRate API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00061] Definition of EcucBooleanParamDef EthTrcvGetBusWuReasonApi ⌈

Parameter Name	EthTrcvGetBusWuReasonApi		
Parent Container	EthTrcvGeneral		
Description	Enables / Disables API to obtain the recent wake-up reason detected by the used Ethernet hardware (e.g. PHY).		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00011] Definition of EcucBooleanParamDef EthTrcvGetDuplexModeApi ⌈

Parameter Name	EthTrcvGetDuplexModeApi		
Parent Container	EthTrcvGeneral		
Description	Enables / Disables EthTrcv_GetDuplexMode API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	

▽

△

	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00009] Definition of EcucBooleanParamDef EthTrcvGetLinkStateApi ⌈

Parameter Name	EthTrcvGetLinkStateApi		
Parent Container	EthTrcvGeneral		
Description	Enables / Disables EthTrcv_GetLinkState API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00046] Definition of EcucBooleanParamDef EthTrcvGetPhyIdentifierApi ⌈

Parameter Name	EthTrcvGetPhyIdentifierApi		
Parent Container	EthTrcvGeneral		
Description	Enables / Disables EthTrcv_GetPhyIdentifier API.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00031] Definition of EcucBooleanParamDef EthTrcvGetTransceiverWakeupModeApi ⌈

Parameter Name	EthTrcvGetTransceiverWakeupModeApi		
Parent Container	EthTrcvGeneral		
Description	Enables / Disables EthTrcv_GetTransceiverWakeupMode API		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: Only valid if EthTrcvWakeUpSupport is not ETHTRCV_WAKEUP_NOT_SUPPORTED		

⌋

[ECUC_EthTrcv_00020] Definition of EcucIntegerParamDef EthTrcvIndex ⌈

Parameter Name	EthTrcvIndex		
Parent Container	EthTrcvGeneral		
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	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00032] Definition of EcucFloatParamDef EthTrcvMainFunctionPeriod ⌈

Parameter Name	EthTrcvMainFunctionPeriod
Parent Container	EthTrcvGeneral
Description	Specifies the period of main function EthTrcv_MainFunction in seconds.
Multiplicity	0..1

▽

△

Type	EcucFloatParamDef		
Range	]0 .. INF[		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00002] Definition of EcucIntegerParamDef EthTrcvMaxTrcvsSupported ⌈

Parameter Name	EthTrcvMaxTrcvsSupported		
Parent Container	EthTrcvGeneral		
Description	–		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	1		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00047] Definition of EcucBooleanParamDef EthTrcvSetPhyTestModeApi ⌈

Parameter Name	EthTrcvSetPhyTestModeApi		
Parent Container	EthTrcvGeneral		
Description	Enables / Disables EthTrcv_SetPhyTestMode API.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	

▽

△

	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00008] Definition of EcucBooleanParamDef EthTrcvStartAutoNegotiationApi ⌈

Parameter Name	EthTrcvStartAutoNegotiationApi		
Parent Container	EthTrcvGeneral		
Description	Enables / Disables EthTrcv_StartAutoNegotiation API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00004] Definition of EcucBooleanParamDef EthTrcvVersionInfoApi ⌈

Parameter Name	EthTrcvVersionInfoApi		
Parent Container	EthTrcvGeneral		
Description	Enables / Disables version info API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00005] Definition of EcucBooleanParamDef EthTrcvVersionInfoApiMacro ⌈

Parameter Name	EthTrcvVersionInfoApiMacro		
Parent Container	EthTrcvGeneral		
Description	Enables / Disables version info API macro implementation		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00030] Definition of EcucEnumerationParamDef EthTrcvWakeUpSupport ⌈

Parameter Name	EthTrcvWakeUpSupport		
Parent Container	EthTrcvGeneral		
Description	Configures how to detect a signaled wake-up by hardware: polling, asychronous check via signal, interrupt or to not used/not supported. In case detection of a wake-up is not support (e.g. wake-up detection not needed or Ethernet hardware does not support wake-up detection), the BSWMD pre-configuration shall be set to ETHTRCV_WAKEUP_NOT_SUPPORTED.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	ETHTRCV_WAKEUP_BY_ASYNCHRONOUS_CHECK	Wake-up detection is done within the main function. Polling of the Ethernet hardware is only done, if a wake-up was signaled. Otherwise the Ethernet hardware is not checked for wake-up.Used e.g. if Ethernet hardware is OA TC10 compliant and the ECU maintain an Ethernet switch.	
	ETHTRCV_WAKEUP_BY_INTERRUPT	Wake-up detection is signaled by interrupt	
	ETHTRCV_WAKEUP_BY_POLLING	Wake-up detection is done by polling continuouslly within the main function	
	ETHTRCV_WAKEUP_NOT_SUPPORTED	Wake up is not supported	
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_EthTrcv_00050] Definition of EcucReferenceDef EthTrcvEcucPartitionRef

⌈

Parameter Name	EthTrcvEcucPartitionRef		
Parent Container	EthTrcvGeneral		
Description	Maps the Ethernet transceiver driver to zero or multiple ECUC partitions to make the modules API available in this partition.		
Multiplicity	0..*		
Type	Reference to EcucPartition		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

⌋

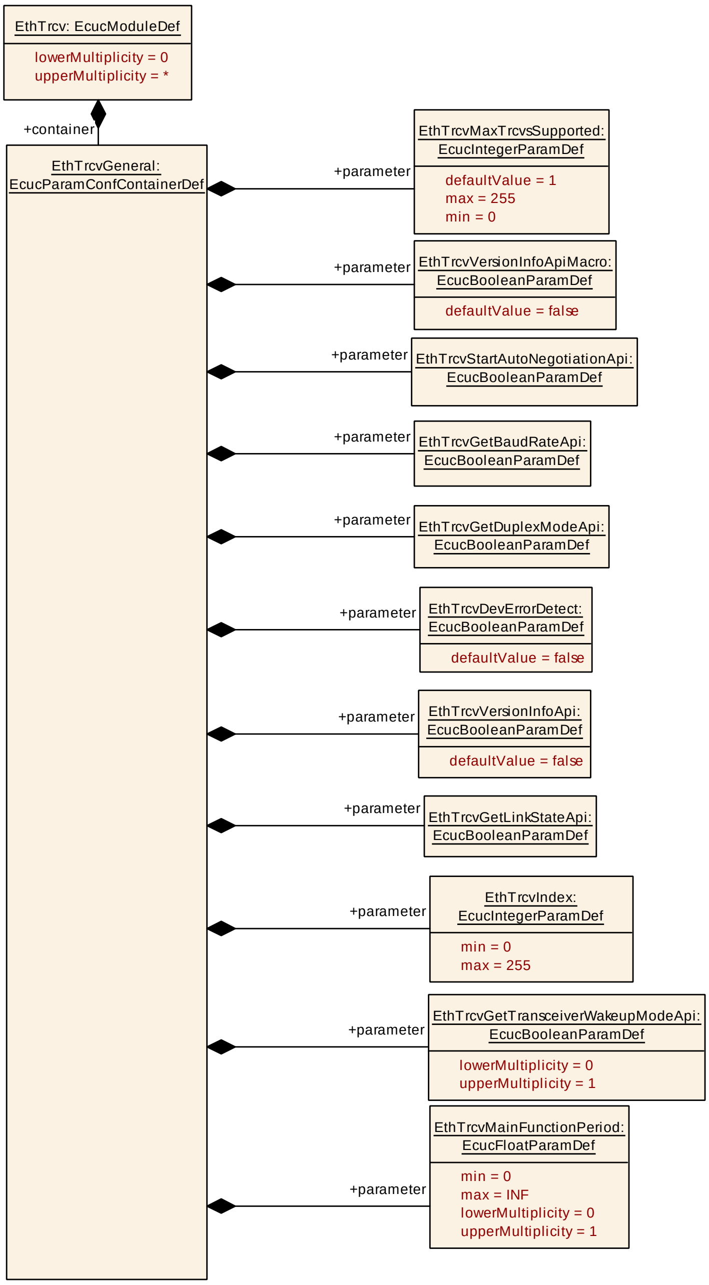

Figure 10.3: EthTrcvGeneral (1/2)

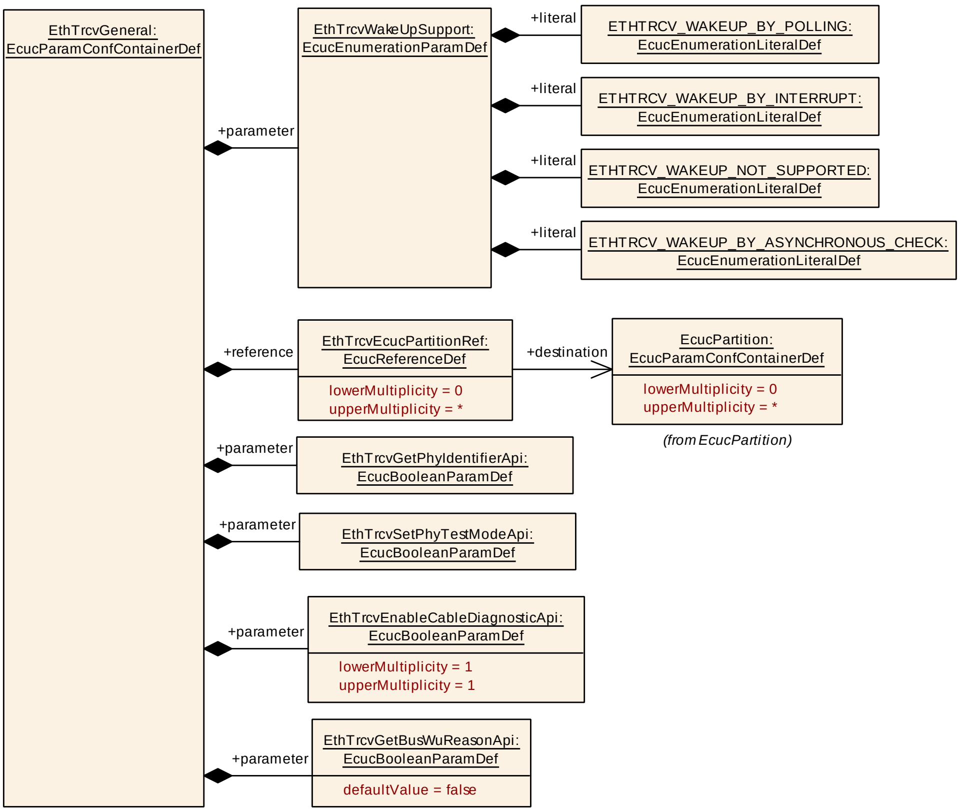

Figure 10.4: EthTrcvGeneral (2/2)

10.3 Published Information

For details refer to the *chapter 10.3, "Published Information"* in [4].

A Not applicable requirements

No items.

B Change history of AUTOSAR traceable items

B.1 Traceable item history of this document according to AUTOSAR Release R24-11

B.1.1 Added Specification Items in R24-11

none

B.1.2 Changed Specification Items in R24-11

Number	Heading
[ECUC_EthTrcv_00001]	Definition of EcucParamConfContainerDef EthTrcvGeneral
[ECUC_EthTrcv_00012]	Definition of EcucParamConfContainerDef EthTrcvConfig
[ECUC_EthTrcv_00022]	Definition of EcucEnumerationParamDef EthTrcvSpeed
[ECUC_EthTrcv_00024]	Definition of EcucEnumerationParamDef EthTrcvPhysLayerType
[ECUC_EthTrcv_00044]	Definition of EcucEnumerationParamDef EthTrcvMacLayerSpeed
[ECUC_EthTrcv_00050]	Definition of EcucReferenceDef EthTrcvEcucPartitionRef
[ECUC_EthTrcv_00055]	Definition of EcucBooleanParamDef EthTrcvEnablePLCA
[SWS_EthTrcv_00027]	Definition of imported datatypes of module EthTrcv
[SWS_EthTrcv_00086]	Definition of optional interfaces requested by module EthTrcv
[SWS_EthTrcv_00144]	Definition of configurable interface <EthTrcvWakeUpCallout>
[SWS_EthTrcv_00208]	
[SWS_EthTrcv_91026]	Definition of API function EthTrcv_MacSecUpdateSecY
[SWS_EthTrcv_91027]	Definition of API function EthTrcv_MacSecInitRxSc
[SWS_EthTrcv_91028]	Definition of API function EthTrcv_MacSecResetRxSc
[SWS_EthTrcv_91029]	Definition of API function EthTrcv_MacSecUpdateTxSa

▽

△

Number	Heading
[SWS_EthTrcv_91030]	Definition of API function EthTrcv_MacSecAddTxSa
[SWS_EthTrcv_91031]	Definition of API function EthTrcv_MacSecDeleteTxSa
[SWS_EthTrcv_91032]	Definition of API function EthTrcv_MacSecAddRxSa
[SWS_EthTrcv_91033]	Definition of API function EthTrcv_MacSecUpdateRxSa
[SWS_EthTrcv_91034]	Definition of API function EthTrcv_MacSecDeleteRxSa
[SWS_EthTrcv_91035]	Definition of API function EthTrcv_MacSecGetTxSaNextPn
[SWS_EthTrcv_91036]	Definition of API function EthTrcv_MacSecGetMacSecStatistics
[SWS_EthTrcv_91037]	Definition of API function EthTrcv_MacSecSetControlledPortEnabled
[SWS_EthTrcv_91038]	Definition of API function EthTrcv_EthSwitchMacSecUpdateSecY
[SWS_EthTrcv_91039]	Definition of API function EthTrcv_EthSwitchMacSecAddTxSa
[SWS_EthTrcv_91040]	Definition of API function EthTrcv_SwitchMacSecAddRxSa
[SWS_EthTrcv_91041]	Definition of API function EthTrcv_SwitchMacSecGetMacSecStatistics

Table B.1: Changed Specification Items in R24-11

B.1.3 Deleted Specification Items in R24-11

Number	Heading
[SWS_EthTrcv_00108]	Definition of callback function EthTrcv_ReadMiiIndication
[SWS_EthTrcv_00109]	Definition of callback function EthTrcv_WriteMiiIndication
[SWS_EthTrcv_00156]	

Table B.2: Deleted Specification Items in R24-11

B.1.4 Added Constraints in R24-11

none

B.1.5 Changed Constraints in R24-11

none

B.1.6 Deleted Constraints in R24-11

none

B.2 Traceable item history of this document according to AUTOSAR Release R23-11

B.2.1 Added Specification Items in R23-11

Number	Heading
[SWS_EthTrcv_-91015]	Definiton of runtime errors in module EthTrcv

Table B.3: Added Specification Items in R23-11

B.2.2 Changed Specification Items in R23-11

none

B.2.3 Deleted Specification Items in R23-11

none

B.2.4 Added Constraints in R23-11

none

B.2.5 Changed Constraints in R23-11

none

B.2.6 Deleted Constraints in R23-11

none

B.3 Traceable item history of this document according to AUTOSAR Release R22-11

B.3.1 Added Specification Items in R22-11

Number	Heading
[SWS_EthTrcv_00208]	
[SWS_EthTrcv_91026]	
[SWS_EthTrcv_91027]	
[SWS_EthTrcv_91028]	
[SWS_EthTrcv_91029]	
[SWS_EthTrcv_91030]	
[SWS_EthTrcv_91031]	
[SWS_EthTrcv_91032]	
[SWS_EthTrcv_91033]	
[SWS_EthTrcv_91034]	
[SWS_EthTrcv_91035]	
[SWS_EthTrcv_91036]	
[SWS_EthTrcv_91037]	
[SWS_EthTrcv_91038]	
[SWS_EthTrcv_91039]	
[SWS_EthTrcv_91040]	

▽

△

Number	Heading
[SWS_EthTrcv_-91041]	

Table B.4: Added Specification Items in R22-11

B.3.2 Changed Specification Items in R22-11

Number	Heading
[SWS_EthTrcv_-00017]	
[SWS_EthTrcv_-00027]	
[SWS_EthTrcv_-00028]	
[SWS_EthTrcv_-00042]	
[SWS_EthTrcv_-00048]	
[SWS_EthTrcv_-00055]	
[SWS_EthTrcv_-00061]	
[SWS_EthTrcv_-00068]	
[SWS_EthTrcv_-00075]	
[SWS_EthTrcv_-00082]	
[SWS_EthTrcv_-00085]	
[SWS_EthTrcv_-00086]	
[SWS_EthTrcv_-00098]	
[SWS_EthTrcv_-00100]	
[SWS_EthTrcv_-00101]	
[SWS_EthTrcv_-00102]	
[SWS_EthTrcv_-00103]	

▽

△

Number	Heading
[SWS_EthTrcv_00106]	
[SWS_EthTrcv_00108]	
[SWS_EthTrcv_00109]	
[SWS_EthTrcv_00114]	
[SWS_EthTrcv_00134]	
[SWS_EthTrcv_00144]	
[SWS_EthTrcv_00201]	
[SWS_EthTrcv_00203]	
[SWS_EthTrcv_00204]	
[SWS_EthTrcv_00205]	
[SWS_EthTrcv_00206]	
[SWS_EthTrcv_91001]	
[SWS_EthTrcv_91002]	
[SWS_EthTrcv_91003]	
[SWS_EthTrcv_91004]	
[SWS_EthTrcv_91005]	
[SWS_EthTrcv_91006]	
[SWS_EthTrcv_91007]	
[SWS_EthTrcv_91008]	
[SWS_EthTrcv_91009]	
[SWS_EthTrcv_91010]	
[SWS_EthTrcv_91011]	

▽

△

Number	Heading
[SWS_EthTrcv_-91012]	
[SWS_EthTrcv_-91013]	
[SWS_EthTrcv_-91014]	
[SWS_EthTrcv_-91025]	

Table B.5: Changed Specification Items in R22-11

B.3.3 Deleted Specification Items in R22-11

Number	Heading
[SWS_EthTrcv_-00139]	

Table B.6: Deleted Specification Items in R22-11

B.3.4 Added Constraints in R22-11

none

B.3.5 Changed Constraints in R22-11

none

B.3.6 Deleted Constraints in R22-11

none

B.4 Traceable item history of this document according to AUTOSAR Release R21-11

B.4.1 Added Specification Items "in R21-11"

Number	Heading
[SWS_EthTrcv_00003]	
[SWS_EthTrcv_00004]	
[SWS_EthTrcv_00005]	
[SWS_EthTrcv_00006]	
[SWS_EthTrcv_00007]	
[SWS_EthTrcv_00009]	
[SWS_EthTrcv_00010]	
[SWS_EthTrcv_00011]	
[SWS_EthTrcv_00012]	
[SWS_EthTrcv_00013]	
[SWS_EthTrcv_00014]	
[SWS_EthTrcv_00015]	
[SWS_EthTrcv_00017]	
[SWS_EthTrcv_00027]	
[SWS_EthTrcv_00028]	
[SWS_EthTrcv_00029]	
[SWS_EthTrcv_00030]	
[SWS_EthTrcv_00032]	

▽

△

Number	Heading
[SWS_EthTrcv_00035]	
[SWS_EthTrcv_00040]	
[SWS_EthTrcv_00042]	
[SWS_EthTrcv_00043]	
[SWS_EthTrcv_00044]	
[SWS_EthTrcv_00045]	
[SWS_EthTrcv_00047]	
[SWS_EthTrcv_00048]	
[SWS_EthTrcv_00049]	
[SWS_EthTrcv_00050]	
[SWS_EthTrcv_00051]	
[SWS_EthTrcv_00052]	
[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]	

▽

△

Number	Heading
[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]	

▽

△

Number	Heading
[SWS_EthTrcv_00089]	
[SWS_EthTrcv_00090]	
[SWS_EthTrcv_00093]	
[SWS_EthTrcv_00094]	
[SWS_EthTrcv_00098]	
[SWS_EthTrcv_00100]	
[SWS_EthTrcv_00101]	
[SWS_EthTrcv_00102]	
[SWS_EthTrcv_00103]	
[SWS_EthTrcv_00104]	
[SWS_EthTrcv_00105]	
[SWS_EthTrcv_00106]	
[SWS_EthTrcv_00107]	
[SWS_EthTrcv_00108]	
[SWS_EthTrcv_00109]	
[SWS_EthTrcv_00110]	
[SWS_EthTrcv_00111]	
[SWS_EthTrcv_00112]	
[SWS_EthTrcv_00114]	
[SWS_EthTrcv_00115]	
[SWS_EthTrcv_00117]	
[SWS_EthTrcv_00118]	

▽

△

Number	Heading
[SWS_EthTrcv_-00134]	
[SWS_EthTrcv_-00135]	
[SWS_EthTrcv_-00137]	
[SWS_EthTrcv_-00138]	
[SWS_EthTrcv_-00139]	
[SWS_EthTrcv_-00140]	
[SWS_EthTrcv_-00141]	
[SWS_EthTrcv_-00144]	
[SWS_EthTrcv_-00145]	
[SWS_EthTrcv_-00146]	
[SWS_EthTrcv_-00147]	
[SWS_EthTrcv_-00148]	
[SWS_EthTrcv_-00149]	
[SWS_EthTrcv_-00151]	
[SWS_EthTrcv_-00152]	
[SWS_EthTrcv_-00153]	
[SWS_EthTrcv_-00154]	
[SWS_EthTrcv_-00155]	
[SWS_EthTrcv_-00156]	
[SWS_EthTrcv_-00157]	
[SWS_EthTrcv_-00158]	
[SWS_EthTrcv_-00159]	

▽

△

Number	Heading
[SWS_EthTrcv_-00160]	
[SWS_EthTrcv_-00161]	
[SWS_EthTrcv_-00162]	
[SWS_EthTrcv_-00163]	
[SWS_EthTrcv_-00165]	
[SWS_EthTrcv_-00166]	
[SWS_EthTrcv_-00167]	
[SWS_EthTrcv_-00168]	
[SWS_EthTrcv_-00169]	
[SWS_EthTrcv_-00170]	
[SWS_EthTrcv_-00171]	
[SWS_EthTrcv_-00172]	
[SWS_EthTrcv_-00173]	
[SWS_EthTrcv_-00174]	
[SWS_EthTrcv_-00175]	
[SWS_EthTrcv_-00176]	
[SWS_EthTrcv_-00177]	
[SWS_EthTrcv_-00178]	
[SWS_EthTrcv_-00179]	
[SWS_EthTrcv_-00180]	
[SWS_EthTrcv_-00181]	
[SWS_EthTrcv_-00182]	

▽

△

Number	Heading
[SWS_EthTrcv_00183]	
[SWS_EthTrcv_00184]	
[SWS_EthTrcv_00185]	
[SWS_EthTrcv_00186]	
[SWS_EthTrcv_00187]	
[SWS_EthTrcv_00188]	
[SWS_EthTrcv_00189]	
[SWS_EthTrcv_00190]	
[SWS_EthTrcv_00191]	
[SWS_EthTrcv_00192]	
[SWS_EthTrcv_00193]	
[SWS_EthTrcv_00194]	
[SWS_EthTrcv_00195]	
[SWS_EthTrcv_00196]	
[SWS_EthTrcv_00197]	
[SWS_EthTrcv_00198]	
[SWS_EthTrcv_00199]	
[SWS_EthTrcv_00200]	
[SWS_EthTrcv_00201]	
[SWS_EthTrcv_00202]	
[SWS_EthTrcv_00203]	
[SWS_EthTrcv_00204]	

▽

△

Number	Heading
[SWS_EthTrcv_00205]	
[SWS_EthTrcv_00206]	
[SWS_EthTrcv_00207]	
[SWS_EthTrcv_91001]	
[SWS_EthTrcv_91002]	
[SWS_EthTrcv_91003]	
[SWS_EthTrcv_91004]	
[SWS_EthTrcv_91005]	
[SWS_EthTrcv_91006]	
[SWS_EthTrcv_91007]	
[SWS_EthTrcv_91008]	
[SWS_EthTrcv_91009]	
[SWS_EthTrcv_91010]	
[SWS_EthTrcv_91011]	
[SWS_EthTrcv_91012]	
[SWS_EthTrcv_91013]	
[SWS_EthTrcv_91014]	
[SWS_EthTrcv_91025]	
[SWS_EthTrcv_CONSTR_00001]	

Table B.7: Added Specification Items in R21-11

B.4.2 Changed Specification Items "in R21-11"

none

B.4.3 Deleted Specification Items "in R21-11"

none

B.4.4 Added Constraints "in R21-11"

none

B.4.5 Changed Constraints "in R21-11"

none

B.4.6 Deleted Constraints "in R21-11"

none