

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	R21-11

Document Change History			
Date	Release	Changed by	Description
2021-11-25	R21-11	AUTOSAR Release Management	<ul style="list-style-type: none"> • New runtime error and return code handling modified • Detailing added in buffer handling • Editorial changes
2020-11-30	R20-11	AUTOSAR Release Management	<ul style="list-style-type: none"> • EthTrcvGetTransceiverModeApi and EthTrcvSetTransceiverModeApi obsolete • Eth_GeneralTypes removed from imported module list • Introduction WakeOnDataLine • Introduction 10BASE-T1S
2019-11-29	R19-11	AUTOSAR Release Management	<ul style="list-style-type: none"> • 2500Mbit Ethernet Support • Cable Diagnostic • Uniformisation Eth_Mode Type • Changed Document Status from final to published
2018-10-31	4.4.0	AUTOSAR Release Management	<ul style="list-style-type: none"> • Explicite transceiver link control • Support of host controllers with multiple cores
2017-12-08	4.3.1	AUTOSAR Release Management	<ul style="list-style-type: none"> • Minor corrections and adaptations • Improved Switch integration
2016-11-30	4.3.0	AUTOSAR Release Management	<ul style="list-style-type: none"> • Configuration via Switch or Mii • 100BASE-T1 test mode support

2015-07-31	4.2.2	AUTOSAR Release Management	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Change from Synchronous to Asynchronous API Ethernet Wakeup Support
2013-10-31	4.1.2	AUTOSAR Release Management	<ul style="list-style-type: none"> Introduction of Eth_GeneralTypes.h Support of API deviation for asynchronous implementation Editorial changes Removed chapter(s) on change documentation
2013-03-15	4.1.1	AUTOSAR Administration	<ul style="list-style-type: none"> 1000 kbit Ethernet Support
2011-12-22	4.0.3	AUTOSAR Administration	<ul style="list-style-type: none"> EthTrcv_GetVersionInfo revised
2010-09-30	3.1.5	AUTOSAR Administration	<ul style="list-style-type: none"> Further post-build configurable parameters Configuration enhanced by additional parameter EthTrcvWaitCount 'Instance ID' removed from Version Info (concerns EthTrcv_GetVersionInfo API) Additional development error in EthTrcv_GetVersionInfo API Improved description of 'XxxCtrlIdx' semantics Specification of behaviour for state switch into already active state
2010-02-02	3.1.4	AUTOSAR Administration	<ul style="list-style-type: none"> 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.

Table of Contents

1	Introduction and functional overview	8
2	Acronyms and Abbreviations	10
3	Related documentation	11
3.1	Input documents	11
3.2	Related standards and norms	11
3.3	Related specification	11
4	Constraints and assumptions	12
4.1	Limitations	12
4.2	Applicability to car domains	12
5	Dependencies to other modules	13
6	Requirements Tracing	14
7	Functional specification	16
7.1	Ethernet BSW stack	16
7.1.1	Indexing scheme	16
7.1.2	Requirements	17
7.1.3	Configuration description	18
7.1.4	Wake-up support	18
7.1.4.1	Wake-up and Sleep on dataline	19
7.1.5	PLCA support	20
7.1.6	Handling of cable diagnostic	20
7.2	Error Classification	21
7.2.1	Development Errors	21
7.2.2	Runtime Errors	22
7.2.3	Transient Faults	22
7.2.4	Production Errors	22
7.2.5	Extended Production Errors	22
8	API specification	23
8.1	Imported types	23
8.2	Type definitions	23
8.2.1	EthTrcv_ConfigType	23
8.2.2	EthTrcv_LinkStateType	23
8.2.3	EthTrcv_StateType	24
8.2.4	EthTrcv_BaudRateType	24
8.2.5	EthTrcv_DuplexModeType	24
8.2.6	EthTrcv_WakeupModeType	25
8.2.7	EthTrcv_PhyTestModeType	26
8.2.8	EthTrcv_PhyLoopbackModeType	26
8.2.9	EthTrcv_PhyTxModeType	26

8.2.10	EthTrcv_CableDiagResultType	27
8.2.11	EthTrcv_MacMethodType	27
8.3	Function definitions	28
8.3.1	EthTrcv_Init	28
8.3.1.1	Initialization of OA TC10 PHYs	29
8.3.2	EthTrcv_SetTransceiverMode	30
8.3.2.1	ETH_MODE_DOWN request	31
8.3.2.2	ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST or ETH_MODE_ACTIVE request	32
8.3.2.3	Error handling	34
8.3.3	EthTrcv_GetTransceiverMode	34
8.3.4	EthTrcv_GetBusWuReason	35
8.3.5	EthTrcv_CheckWakeup	36
8.3.6	EthTrcv_StartAutoNegotiation	38
8.3.7	EthTrcv_TransceiverLinkStateRequest	39
8.3.7.1	Point to point network topology	40
8.3.7.2	Bus topology	40
8.3.8	EthTrcv_GetLinkState	40
8.3.8.1	Point to point network topology	41
8.3.8.2	Point to point network topology	42
8.3.9	EthTrcv_GetBaudRate	42
8.3.10	EthTrcv_GetDuplexMode	43
8.3.11	EthTrcv_SetPhyTestMode	44
8.3.12	EthTrcv_SetPhyLoopbackMode	45
8.3.13	EthTrcv_GetPhySignalQuality	46
8.3.14	EthTrcv_SetPhyTxMode	46
8.3.15	EthTrcv_RunCableDiagnostic	47
8.3.16	EthTrcv_GetCableDiagnosticsResult	48
8.3.17	EthTrcv_GetPhyIdentifier	48
8.3.18	EthTrcv_GetMacMethod	49
8.3.19	EthTrcv_GetVersionInfo	50
8.4	Callback notifications	50
8.4.1	EthTrcv_ReadMiiIndication	50
8.4.2	EthTrcv_WriteMiiIndication	51
8.5	Scheduled functions	52
8.5.1	EthTrcv_MainFunction	52
8.6	Expected interfaces	53
8.6.1	Mandatory Interfaces	53
8.6.2	Optional Interfaces	54
8.6.3	Configurable interfaces	54
8.7	InterruptServiceRoutines	55
9	Sequence diagrams	56
10	Configuration specification	57
10.1	How to read this chapter	57
10.2	Containers and configuration parameters	57

10.2.1	EthTrcv	57
10.2.2	EthTrcvConfigSet	57
10.2.3	EthTrcvConfig	58
10.2.4	EthTrcvDemEventParameterRefs	76
10.2.5	EthTrcvMgmtInterface	76
10.2.5.1	EthTrcvMiiInterface	77
10.2.5.2	EthTrcvSwitchInterface	78
10.2.6	EthTrcvWakeupMap	79
10.2.7	EthTrcvGeneral	80
10.3	Published Information	89
A	Not applicable requirements	90

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.

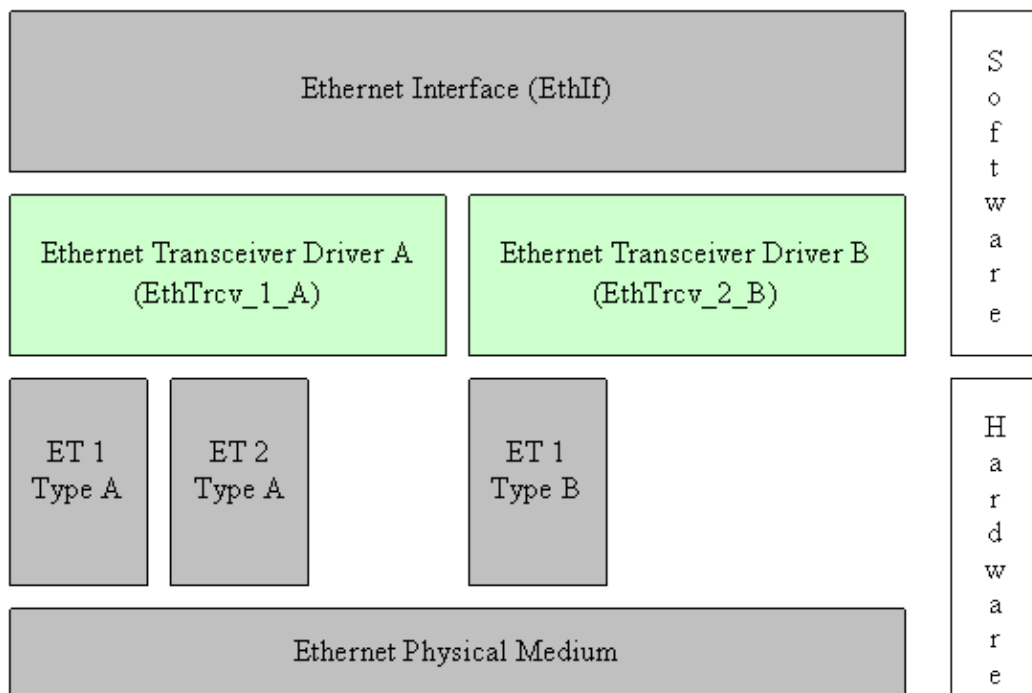


Figure 1.1: Ethernet stack module overview

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_SWS_EthernetInterface
- [2] Glossary
AUTOSAR_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_SWS_BSWGeneral
- [5] Requirements on Ethernet Support in AUTOSAR
AUTOSAR_SRS_Ethernet
- [6] Specification of Default Error Tracer
AUTOSAR_SWS_DefaultErrorTracer

3.2 Related standards and norms

- [7] IEC 7498-1 The Basic Model, IEC Norm, 1994
- [8] IEEE 802.3-2006
- [9] IEEE 802.3cg-2019

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_00106]	The Ethernet Transceiver Driver shall switch on/off wake up functionality at pre compile time.	[SWS_EthTrcv_00139]
[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]

Requirement	Description	Satisfied by
[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]

7 Functional specification

7.1 Ethernet BSW stack

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

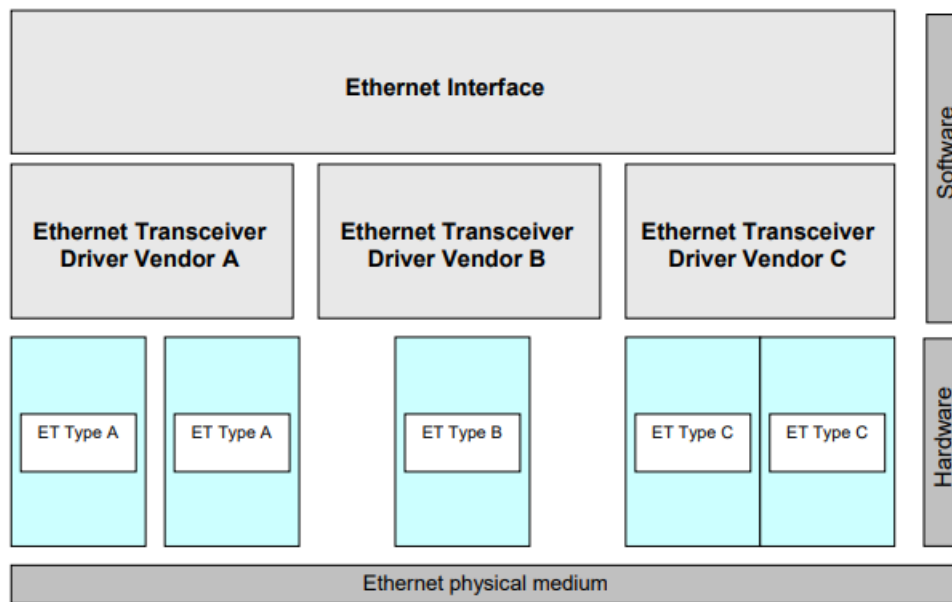


Figure 7.1: Basic Structure of the Ethernet BSW stack

7.1.1 Indexing scheme

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

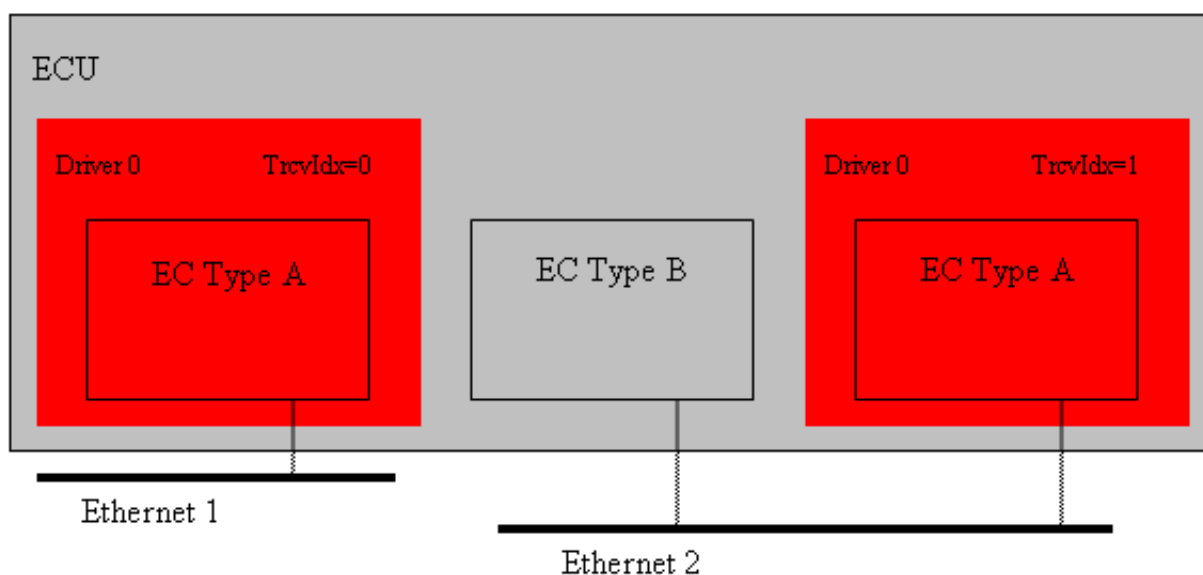


Figure 7.2: Ethernet Transceiver Driver indexing scheme

[SWS_EthTrcv_00003] [The Ethernet Transceiver Driver is using a zero-based index to abstract the access for upper software layers. The parameter 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] [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).] ([SRS_Eth_00108](#))

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] [The wake up interrupt handler (if present) shall clear the interrupt and identify the wake up reason and store it.]([SRS_Eth_00107](#))

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).

Additionally 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) 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, autonegotiation 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 than 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 fulfilled 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 measurement 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] [

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

There are no runtime errors.

7.2.3 Transient Faults

There are no transient faults.

7.2.4 Production Errors

There are no production errors.

7.2.5 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 succeeds 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] [

<i>Module</i>	<i>Header File</i>	<i>Imported Type</i>
Dem	Rte_Dem_Type.h	Dem_EventIdType
	Rte_Dem_Type.h	Dem_EventStatusType
EcuM	EcuM.h	EcuM_WakeupSourceType
Eth	Eth_GeneralTypes.h	Eth_ModeType (draft)
Icu	Icu.h	Icu_ChannelType
Std	Std_Types.h	Std_ReturnType
	Std_Types.h	Std_VersionInfoType

]()

8.2 Type definitions

8.2.1 EthTrcv_ConfigType

[SWS_EthTrcv_00098] [

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] [

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] [

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] [

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] [

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] [

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		

] (SRS_Eth_00108)

8.2.7 EthTrcv_PhyTestModeType

[SWS_EthTrcv_91002] [

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		

](SRS_Eth_00117)

8.2.8 EthTrcv_PhyLoopbackModeType

[SWS_EthTrcv_91004] [

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		

](SRS_Eth_00117)

8.2.9 EthTrcv_PhyTxModeType

[SWS_EthTrcv_91006] [

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		

]([SRS_Eth_00117](#))

8.2.10 EthTrcv_CableDiagResultType

[SWS_EthTrcv_91008] [

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		

]([SRS_Eth_00117](#))

8.2.11 EthTrcv_MacMethodType

[SWS_EthTrcv_91013]{DRAFT} [

Name	EthTrcv_MacMethodType (draft)		
Kind	Enumeration		
Range	ETHTRCV_MAC_TYPE_ CSMA_CD	0x00	Carrier sense 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] [

Service Name	EthTrcv_Init	
Syntax	<pre>void EthTrcv_Init (const EthTrcv_ConfigType* CfgPtr)</pre>	
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]{DRAFT} [The function shall store the access to the configuration structure for subsequent API calls.] ([SRS_Eth_00039](#))

[SWS_EthTrcv_00203]{DRAFT} [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](#)].

]([SRS_Eth_00039](#))

[SWS_EthTrcv_00115] [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.] ([SRS_Eth_00108](#))

[SWS_EthTrcv_00204]{DRAFT} [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.] ([SRS_Eth_00039](#))

[SWS_EthTrcv_00035] [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]).

]([SRS_Eth_00039](#))

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] [The function shall change the state of the component from ETHTRCV_STATE_UNINIT to ETHTRCV_STATE_INIT.] ([SRS_Eth_00039](#))

[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]{DRAFT} [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] ([SRS_Eth_00039](#))

[SWS_EthTrcv_00201]{DRAFT} [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 EthTrcvSleep-ModeExecutionDelay shall be started. |(SRS_Eth_00039)

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] [

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 Ethernet 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 Ethernet 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	

|(SRS_Eth_00154)

[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: Further 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]{DRAFT} [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] [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.]([SRS_Eth_00151](#))

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] [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`.] ([SRS_Eth_00151](#))

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

[SWS_EthTrcv_00182] [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 [\[3\]](#)).] ([SRS_Eth_00151](#))

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] [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.

]([SRS_Eth_00108](#))

[SWS_EthTrcv_00207] [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.

]([SRS_Eth_00108](#))

[SWS_EthTrcv_00183] [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.

]([SRS_Eth_00108](#))

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 Transceiver 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] [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.

]([SRS_Eth_00108](#), [SRS_Eth_00154](#))

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) implicitly 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 repetitions 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] [

Service Name	<code>EthTrcv_GetTransceiverMode</code>
Syntax	<pre>Std_ReturnType EthTrcv_GetTransceiverMode (uint8 TrcvIdx, Eth_ModeType* TrcvModePtr)</pre>
Service ID [hex]	<code>0x04</code>





Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
Parameters (inout)	None	
Parameters (out)	TrcvModePtr	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] [

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	

]([SRS_Eth_00107](#))

[SWS_EthTrcv_00186] [The function EthTrcv_GetBusWuReason shall read the stored wake-up reason and provide the information in WakeupReasonPtr.]([SRS_Eth_00107](#))

[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.]([SRS_Eth_00107](#))

[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.]([SRS_Eth_00107](#))

[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.]([SRS_Eth_00107](#))

[SWS_EthTrcv_00190] [The function shall be pre compile time configurable On/Off by the configuration parameter: EthTrcvGetBusWuReasonApi [ECUC_EthTrcv_00061].]([SRS_Eth_00107](#))

8.3.5 EthTrcv_CheckWakeup

[SWS_EthTrcv_00134] [

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 Ethernet 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] [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.] ([SRS_Eth_00107](#))

[SWS_EthTrcv_00202] [If the function EthTrcv_CheckWakeup() is called, EthTrcvWakeUpSleepOnDatelineEnabled 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.] ([SRS_Eth_00107](#))

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] [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.] ([SRS_Eth_00108](#), [SRS_Eth_00107](#))

Note: Asynchronous 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 Eth Trcv_CheckWakeUp()) to support timing constraints regarding runtime of the host ECU.

[SWS_EthTrcv_00137] [If development error detection is enabled: The function Eth Trcv_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 Eth Trcv_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_00139]{OBSOLETE} [The function EthTrcv_CheckWakeUp() shall be only available if EthTrcvWakeUpSupport is something else than ETHTRCV_WAKEUP_NOT_SUPPORTED.] ([SRS_Eth_00106](#))

[SWS_EthTrcv_00140] [*EthTrcv_Init()* shall be called before *EthTrcv_Check Wakeup()*.] ()

8.3.6 EthTrcv_StartAutoNegotiation

[SWS_EthTrcv_00055] [

Service Name	EthTrcv_StartAutoNegotiation	
Syntax	Std_ReturnType EthTrcv_StartAutoNegotiation (uint8 TrcvIdx)	
Service ID [hex]	0x05	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: success E_NOT_OK: transceiver could not be initialized
Description	Restarts the negotiation of the transmission parameters used by the indexed transceiver	
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] [The function shall be pre compile time configurable On/Off by the configuration parameter: EthTrcvStartAutoNegotiationApi [ECUC_EthTrcv_00008] and EthTrcvEnablePLCA [ECUC_EthTrcv_00055].] ([SRS_Eth_00149](#))

[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] [

Service Name	EthTrcv_TransceiverLinkStateRequest	
Syntax	<pre>Std_ReturnType EthTrcv_TransceiverLinkStateRequest (uint8 TrcvIdx, EthTrcv_LinkStateType LinkState)</pre>	
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 Ethernet Transceiver Driver
	LinkState	The Ethernet link state of a physical Ethernet 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 Ethernet 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]{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] [

Service Name	EthTrcv_GetLinkState	
Syntax	<pre>Std_ReturnType EthTrcv_GetLinkState (uint8 TrcvIdx, EthTrcv_LinkStateType* LinkStatePtr)</pre>	
Service ID [hex]	0x06	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
Parameters (inout)	None	
Parameters (out)	LinkStatePtr	ETHTRCV_LINK_STATE_DOWN: transceiver is 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] [The function shall read the current transceiver link state, except for the following condition:

If EthTrcvWakeupSleepOnDatelineEnabled 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.] ([SRS_Eth_00040](#))

Note: OA TC10 compliant Ethernet hardware is not able to transfer data on the data 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 Point to point network topology

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

[SWS_EthTrcv_00198]{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]{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]{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] [

Service Name	EthTrcv_GetBaudRate	
Syntax	<pre>Std_ReturnType EthTrcv_GetBaudRate (uint8 TrcvIdx, EthTrcv_BaudRateType* BaudRatePtr)</pre>	
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] [

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] [

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	

](SRS_Eth_00117)

[SWS_EthTrcv_00166] [If the internal cable diagnostic state is ETHTRCV_CABLE-DIAG_PENDING, EthTrcv_SetPhyTestMode() shall return E_NOT_OK.]()

[SWS_EthTrcv_00147] [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.](SRS_Eth_00117)

[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] [

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	

](SRS_Eth_00117)

[SWS_EthTrcv_00167] [If the internal cable diagnostic state is ETHTRCV_CABLE-DIAG_PENDING, the EthTrcv_SetPhyLoopbackMode() shall return E_NOT_OK.]()

[SWS_EthTrcv_00149] [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.](SRS_Eth_00117)

8.3.13 EthTrcv_GetPhySignalQuality

[SWS_EthTrcv_91001] [

Service Name	EthTrcv_GetPhySignalQuality	
Syntax	<pre>Std_ReturnType EthTrcv_GetPhySignalQuality (uint8 TrcvIdx, uint32* SignalQualityPtr)</pre>	
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	

](SRS_Eth_00117)

8.3.14 EthTrcv_SetPhyTxMode

[SWS_EthTrcv_91007] [

Service Name	EthTrcv_SetPhyTxMode	
Syntax	<pre>Std_ReturnType EthTrcv_SetPhyTxMode (uint8 TrcvIdx, EthTrcv_PhyTxModeType Mode)</pre>	
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	

](SRS_Eth_00117)

[SWS_EthTrcv_00168] [If the internal cable diagnostic state is ETHTRCV_CABLE-DIAG_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] [

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] [

Service Name	EthTrcv_GetCableDiagnosticsResult	
Syntax	<pre>Std_ReturnType EthTrcv_GetCableDiagnosticsResult (uint8 TrcvIdx, EthTrcv_CableDiagResultType* ResultPtr)</pre>	
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	

](SRS_Eth_00117)

[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] [

Service Name	EthTrcv_GetPhyIdentifier	
Syntax	<pre>Std_ReturnType EthTrcv_GetPhyIdentifier (uint8 TrcvIdx, uint32* OrgUniqueIdPtr, uint8* ModelNrPtr, uint8* RevisionNrPtr)</pre>	
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 Identifier.	
Available via	EthTrcv.h	

](SRS_Eth_00117)

[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] [

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 collision 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] [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.](SRS_Eth_00149)

[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] [

Service Name	EthTrcv_GetVersionInfo	
Syntax	<pre>void EthTrcv_GetVersionInfo (Std_VersionInfoType* VersionInfoPtr)</pre>	
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].]()

8.4 Callback notifications

8.4.1 EthTrcv_ReadMiiIndication

[SWS_EthTrcv_00108] [

Service Name	EthTrcv_ReadMiiIndication	
Syntax	<pre>void EthTrcv_ReadMiiIndication (uint8 CtrlIdx, uint8 TrcvIdx, uint8 RegIdx, uint8 RegVal)</pre>	
Service ID [hex]	0x09	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant for the same CtrlIdx, reentrant for different	
Parameters (in)	CtrlIdx	Index of the controller within the context of the Ethernet Driver
	TrcvIdx	Index of the transceiver on the MII
	RegIdx	Index of the transceiver register on the MII
	RegVal	Value contained in the indexed register
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Called when information has been read out via MII interface. Triggered by previous Eth_ReadMii call. Can directly be called within Eth_ReadMii.	
Available via	EthTrcv.h	

]()

8.4.2 EthTrcv_WriteMiiIndication

[SWS_EthTrcv_00109] [

Service Name	EthTrcv_WriteMiiIndication	
Syntax	<pre>void EthTrcv_WriteMiiIndication (uint8 CtrlIdx, uint8 TrcvIdx, uint8 RegIdx)</pre>	
Service ID [hex]	0x0a	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant for the same CtrlIdx, reentrant for different	
Parameters (in)	CtrlIdx	Index of the controller within the context of the Ethernet Driver
	TrcvIdx	Index of the transceiver on the MII
	RegIdx	Index of the transceiver register on the MII
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Called when information has been written via MII interface. Triggered by previous Eth_WriteMii call. Can directly be called within Eth_WriteMii.	
Available via	EthTrcv.h	

]()

8.5 Scheduled functions

8.5.1 EthTrcv_MainFunction

[SWS_EthTrcv_00106] [

Service Name	EthTrcv_MainFunction
Syntax	<pre>void EthTrcv_MainFunction (void)</pre>
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] [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.]([SRS_Eth_00108](#))

[SWS_EthTrcv_00192]{DRAFT} [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.]([SRS_ModeMgm_09267](#), [SRS_Eth_00152](#))

[SWS_EthTrcv_00193] [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]).]([SRS_Eth_00151](#))

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] [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.] ([SRS_Eth_00155](#))

[SWS_EthTrcv_00195] [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 Ethernet hardware (PHY) to go to sleep.] ([SRS_Eth_00155](#))

[SWS_EthTrcv_00196] [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.] ([SRS_Eth_00154](#))

8.6 Expected interfaces

This chapter lists all interfaces required from other modules.

8.6.1 Mandatory Interfaces

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

[SWS_EthTrcv_00085] [

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/Dem ConfigSet/DemEventParameter/DemEvent ReportingType) == STANDARD_REPORTING)





API Function	Header File	Description
EthIf_TrvcModeIndication	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.

]()

8.6.2 Optional Interfaces

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

[SWS_EthTrcv_00086] [

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_WriteMii	Eth.h	Configures a transceiver register or triggers a function offered by the receiver
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
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.

]()

8.6.3 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] [

Service Name	<EthTrcvWakeUpCallout>	
Syntax	<pre>void <EthTrcvWakeUpCallout> (uint8 TrcvIdx, Eth_ModeType TrcvMode)</pre>	
Service ID [hex]	0x11	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant Dont care	
Parameters (in)	TrcvIdx	Index of the Ethernet Transceiver
	TrcvMode	<p>ETH_MODE_ACTIVE: enable the transceiver</p> <p>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.</p>
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	<p>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).</p>	
Available via	EthTrcv_Externals.h	

]()

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

8.7 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

Module SWS Item	ECUC_EthTrcv_00034	
Module Name	EthTrcv	
Module 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

SWS Item	[ECUC_EthTrcv_00016]
Container Name	EthTrcvConfigSet

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

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

10.2.3 EthTrcvConfig

SWS Item	[ECUC_EthTrcv_00012]
Container Name	EthTrcvConfig
Parent Container	EthTrcvConfigSet
Description	Configuration of the individual transceiver
Configuration Parameters	

Name	EthTrcvActAsSlavePassiveEnabled [ECUC_EthTrcv_00071]		
Parent Container	EthTrcvConfig		
Description	<p>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).</p> <p>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.</p> <p>Tags: atp.Status=draft</p>		
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 EthTrcvWakeupSleepOnDatelineEnabled is set to TRUE and the corresponding ComM channel has set ComMNMVariant to SLAVE_PASSIVE.		

Name	EthTrcvConnNeg [ECUC_EthTrcv_00025]		
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.		

Name	EthTrcvDuplexMode [ECUC_EthTrcv_00023]		
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		

Name	EthTrcvForceSleepEnabled [ECUC_EthTrcv_00074]		
Parent Container	EthTrcvConfig		
Description	<p>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.</p> <p>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.</p>		
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 EthTrcvWakeupSleepOnDatelineEnabled is set to TRUE.		

Name	EthTrcvIdx [ECUC_EthTrcv_00013]		
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		

Name	EthTrcvMacLayerSpeed [ECUC_EthTrcv_00044]		
Parent Container	EthTrcvConfig		
Description	Defines the baud rate of the MAC layer.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	ETH_MAC_LAYER_SPEED_100M		

Post-Build Variant Multiplicity	ETH_MAC_LAYER_SPEED_10G		
	ETH_MAC_LAYER_SPEED_10M		
	ETH_MAC_LAYER_SPEED_1G		
	ETH_MAC_LAYER_SPEED_2500M 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		

Name	EthTrcvMacLayerSubType [ECUC_EthTrcv_00043]		
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		

Name	EthTrcvMacLayerType [ECUC_EthTrcv_00035]		
Parent Container	EthTrcvConfig		
Description	Defines the MAC layer type of the ethernet transceiver.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	TRCV_MAC_LAYER_TYP E_XGMII	MAC layer interface (data) bandwidth class 1Gbit/s (e.g. GMII, RGMII, SGMII, RvGMII, USGMII)	
	TRCV_MAC_LAYER_TYP E_XMII	MAC layer interface (data) bandwidth class 10-100Mbit/s (e.g. RMII, RvMII, SMII, MII)	
	TRCV_MAC_LAYER_TYP E_XXGMII	MAC layer interface (data) bandwidth 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		

Name	EthTrcvPhysLayerPlcaLocalNodeId [ECUC_EthTrcv_00056]		
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.
---------------------------	--

Name	EthTrcvPhysLayerPlcaMaxBurstCount [ECUC_EthTrcv_00059]		
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.		

Name	EthTrcvPhysLayerPlcaMaxBurstTimer [ECUC_EthTrcv_00060]		
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.		

Name	EthTrcvPhysLayerPlcaNodeCount [ECUC_EthTrcv_00058]		
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.		

Name	EthTrcvPhysLayerPlcaTransmitOpportunityTimer [ECUC_EthTrcv_00057]		
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.		

Name	EthTrcvPhysLayerType [ECUC_EthTrcv_00024]		
Parent Container	EthTrcvConfig		
Description	Specifies the physical layer type of the Ethernet transceiver link.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	TRCV_PHYS_LAYER_TY PE_1000BASE_T		physical layer interface 1000BASE-T (1Gbit/s, 4 pairs). Used for consumer electronic.
	TRCV_PHYS_LAYER_TY PE_1000BASE_T1		physical layer interface 1000BASE-T1 (1Gbit/s, 1 pair). Used for automotive.
	TRCV_PHYS_LAYER_TY PE_100BASE_T1		physical layer interface 100BASE-T1 (100Mbit/s, 1 pair). Used for automotive.
	TRCV_PHYS_LAYER_TY PE_100BASE_TX		physical layer interface 100BASE-TX (100Mbit/s, 2 pairs). Used for consumer electronic.
	TRCV_PHYS_LAYER_TY PE_10BASE_T1S		Physical layer interface 10BASE-T1S (10Mbit/s, 1 pair). 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		

Name	EthTrcvSleepModeExecutionDelay [ECUC_EthTrcv_00063]		
Parent Container	EthTrcvConfig		
Description	<p>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.</p> <p>The value shall be an integral multiple of EthTrcvMainFunctionPeriod.</p>		
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.		

Name	EthTrcvSleepRequestNumberOfRepetitions [ECUC_EthTrcv_00072]		
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.		

Name	EthTrcvSleepRequestRepetitionPeriod [ECUC_EthTrcv_00073]		
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.		

Name	EthTrcvSpeed [ECUC_EthTrcv_00022]		
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	
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		

Name	EthTrcvWakeUpCallout [ECUC_EthTrcv_00028]		
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.		

Name	EthTrcvWakeupForwardLocalEnabled [ECUC_EthTrcv_00067]		
Parent Container	EthTrcvConfig		
Description	<p>Specifies if remote wake up forwarding is enabled (TRUE) or disabled (FALSE) for OA TC10 compliant Ethernet Transceiver.</p> <p>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).</p>		
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.		

Name	EthTrcvWakeupForwardRemoteEnabled [ECUC_EthTrcv_00068]		
Parent Container	EthTrcvConfig		
Description	<p>Specifies if local wake up forwarding is enabled (TRUE) or disabled (FALSE) for OA TC10 compliant Ethernet Transceiver.</p> <p>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).</p>		
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.		

Name	EthTrcvWakeupLocalDetectionTime [ECUC_EthTrcv_00069]		
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 evaluated 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.		

Name	EthTrcvWakeupLocalDurationTime [ECUC_EthTrcv_00070]		
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
---------------------------	---

Name	EthTrcvWakeupLocalEnabled [ECUC_EthTrcv_00065]		
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.		

Name	EthTrcvWakeupRemoteEnabled [ECUC_EthTrcv_00066]		
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.		

Name	EthTrcvWakeupRequestNumberOfRepetitions [ECUC_EthTrcv_00075]		
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.		

Name	EthTrcvWakeUpRequestRepetitionPeriod [ECUC_EthTrcv_00062]		
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.		

Name	EthTrcvWakeupSleepOnDatalineEnabled [ECUC_EthTrcv_00064]		
Parent Container	EthTrcvConfig		
Description	<p>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.</p> <p>If the parameter is configured, wake-up and sleep on data line is supported according to OA TC10 by the used Ethernet hardware (PHY).</p> <p>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.</p> <p>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.</p>		
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		

Name	EthTrcvConfigEcucPartitionRef [ECUC_EthTrcv_00051]		
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		

Name	EthTrcvIcuChannelRef [ECUC_EthTrcv_00026]		
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		

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 EthTrcvWakeUpSupport is not ETHTRCV_WAKEUP_NOT_SUPPORTED.

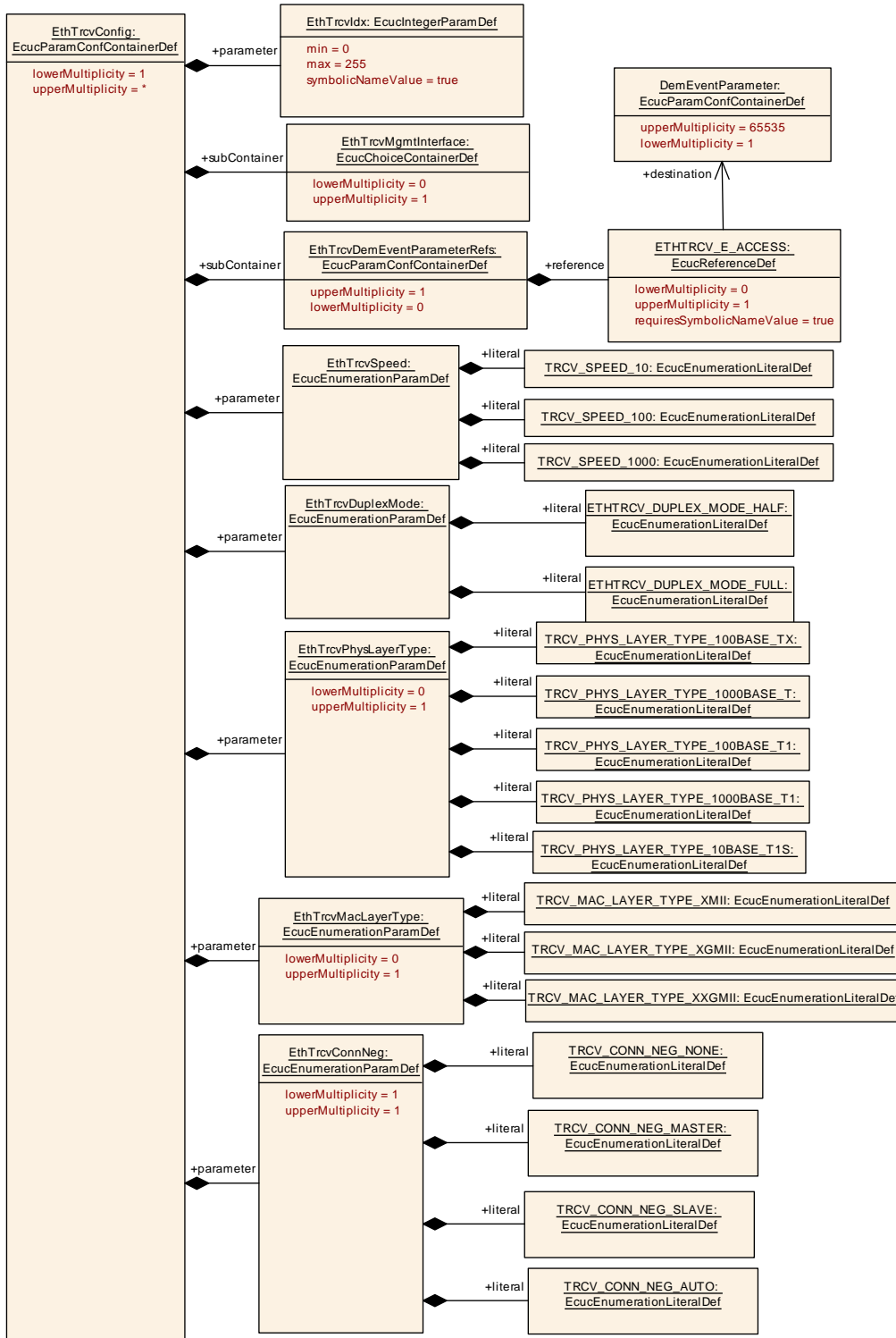


Figure 10.1: EthTrcvConfig (1/2)

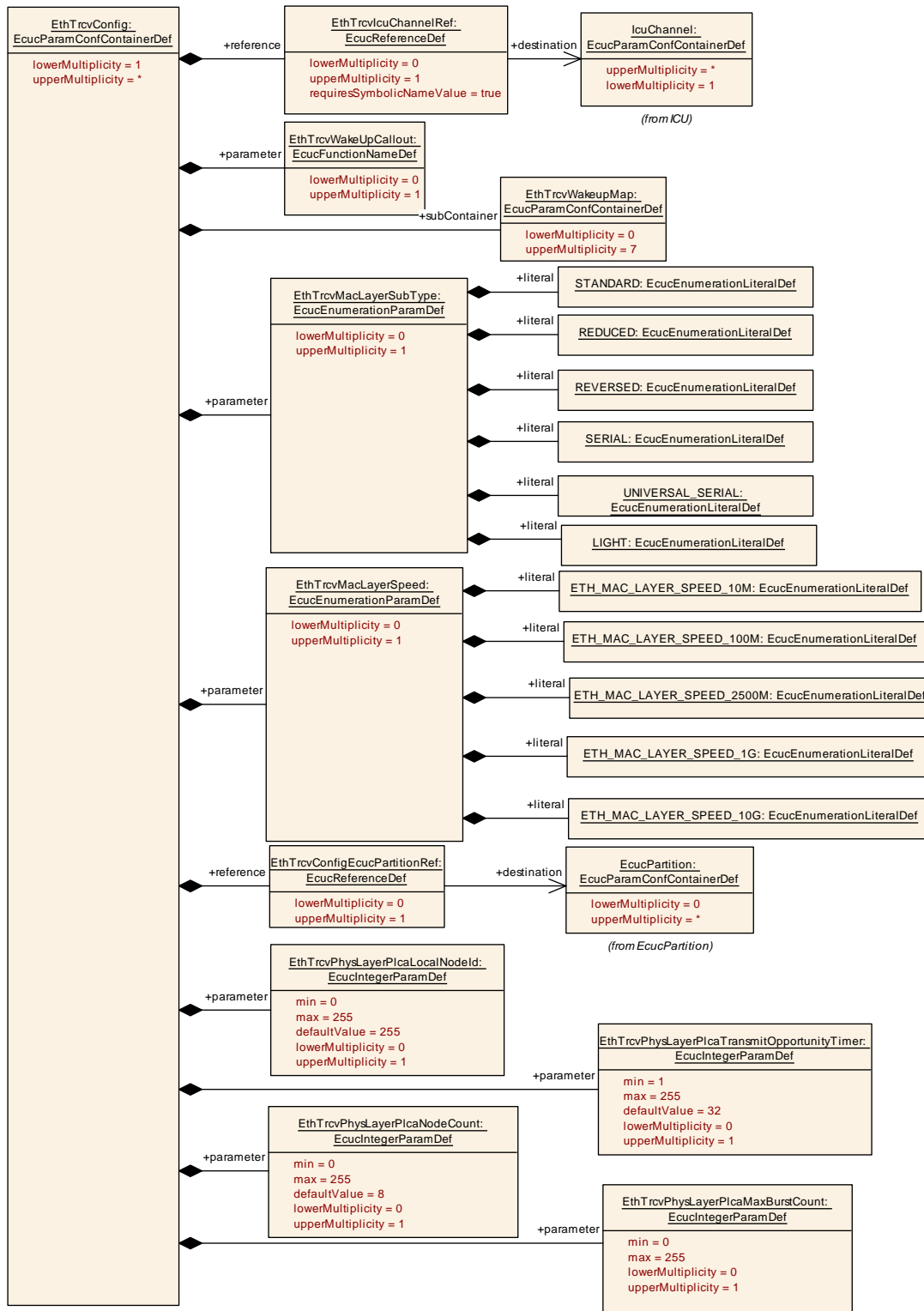


Figure 10.2: EthTrcvConfig (2/2)

[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

SWS Item	[ECUC_EthTrcv_00017]
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 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.
Configuration Parameters	

Name	ETHTRCV_E_ACCESS [ECUC_EthTrcv_00018]		
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		

No Included Containers

10.2.5 EthTrcvMgmtInterface

SWS Item	[ECUC_EthTrcv_00036]
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	–	
Configuration 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 tranceiver.

10.2.5.1 EthTrcvMiiInterface

SWS Item	[ECUC_EthTrcv_00037]		
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			

Name	EthTrcvCtrlIdx [ECUC_EthTrcv_00014]		
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		

Name	EthTrcvMiidx [ECUC_EthTrcv_00038]		
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		

No Included Containers

10.2.5.2 EthTrcvSwitchInterface

SWS Item	[ECUC_EthTrcv_00040]		
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

Name	EthTrcvSwitchPortRef [ECUC_EthTrcv_00042]		
Parent Container	EthTrcvSwitchInterface		
Description	Reference to a switch port.		
Multiplicity	1		
Type	Symbolic name reference to EthSwPort		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	EthTrcvSwitchRef [ECUC_EthTrcv_00041]		
Parent Container	EthTrcvSwitchInterface		
Description	Reference to a switch configuration container.		
Multiplicity	1		
Type	Symbolic name reference to EthSwConfig		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.2.6 EthTrcvWakeupMap

SWS Item	[ECUC_EthTrcv_00027]
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	

Name	EthTrcvWakeupReason [ECUC_EthTrcv_00033]		
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_SYSTEM_ERROR	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		

Name	EthTrcvWakeupSourceRef [ECUC_EthTrcv_00029]		
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		

No Included Containers

10.2.7 EthTrcvGeneral

SWS Item	[ECUC_EthTrcv_00001]
Container Name	EthTrcvGeneral
Parent Container	EthTrcv
Description	General configuration of Ethernet Transceiver Driver module

Configuration Parameters

Name	EthTrcvDevErrorDetect [ECUC_EthTrcv_00003]		
Parent Container	EthTrcvGeneral		
Description	Switches the development error detection and notification on or off. <ul style="list-style-type: none"> • 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		

Name	EthTrcvEnableCableDiagnosticApi [ECUC_EthTrcv_00054]		
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.		

Name	EthTrcvEnablePLCA [ECUC_EthTrcv_00055]		
Parent Container	EthTrcvGeneral		
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.		

Name	EthTrcvGetBaudRateApi [ECUC_EthTrcv_00010]		
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		

Name	EthTrcvGetBusWuReasonApi [ECUC_EthTrcv_00061]		
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		

Name	EthTrcvGetDuplexModeApi [ECUC_EthTrcv_00011]		
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		

Name	EthTrcvGetLinkStateApi [ECUC_EthTrcv_00009]		
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		

Name	EthTrcvGetPhyIdentifierApi [ECUC_EthTrcv_00046]		
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		

Name	EthTrcvGetTransceiverWakeupModeApi [ECUC_EthTrcv_00031]		
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		

Name	EthTrcvIndex [ECUC_EthTrcv_00020]		
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		

Name	EthTrcvMainFunctionPeriod [ECUC_EthTrcv_00032]		
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		

Name	EthTrcvMaxTrcvSupported [ECUC_EthTrcv_00002]		
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		

Name	EthTrcvSetPhyTestModeApi [ECUC_EthTrcv_00047]		
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		

Name	EthTrcvStartAutoNegotiationApi [ECUC_EthTrcv_00008]		
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		

Name	EthTrcvVersionInfoApi [ECUC_EthTrcv_00004]		
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		

Name	EthTrcvVersionInfoApiMacro [ECUC_EthTrcv_00005]		
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		

Name	EthTrcvWakeUpSupport [ECUC_EthTrcv_00030]		
Parent Container	EthTrcvGeneral		
Description	Configures how to detect a signaled wake-up by hardware: polling, asynchronous 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_ASYNC HRONOUS_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.	
Post-Build Variant Value	ETHTRCV_WAKEUP_BY_INTERRUPT	Wake-up detection is signaled by interrupt	
	ETHTRCV_WAKEUP_BY_POLLING	Wake-up detection is done by polling continuously within the main function	
	ETHTRCV_WAKEUP_NOT_SUPPORTED	Wake up is not supported	
	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	EthTrcvEcucPartitionRef [ECUC_EthTrcv_00050]		
Parent Container	EthTrcvGeneral		
Description	Maps the Ethernet transceiver driver to zero or multiple ECUC partitions to make the modules API available in this partition. The Ethernet transceiver driver will operate as an independent instance in each of the partitions.		
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

No Included Containers

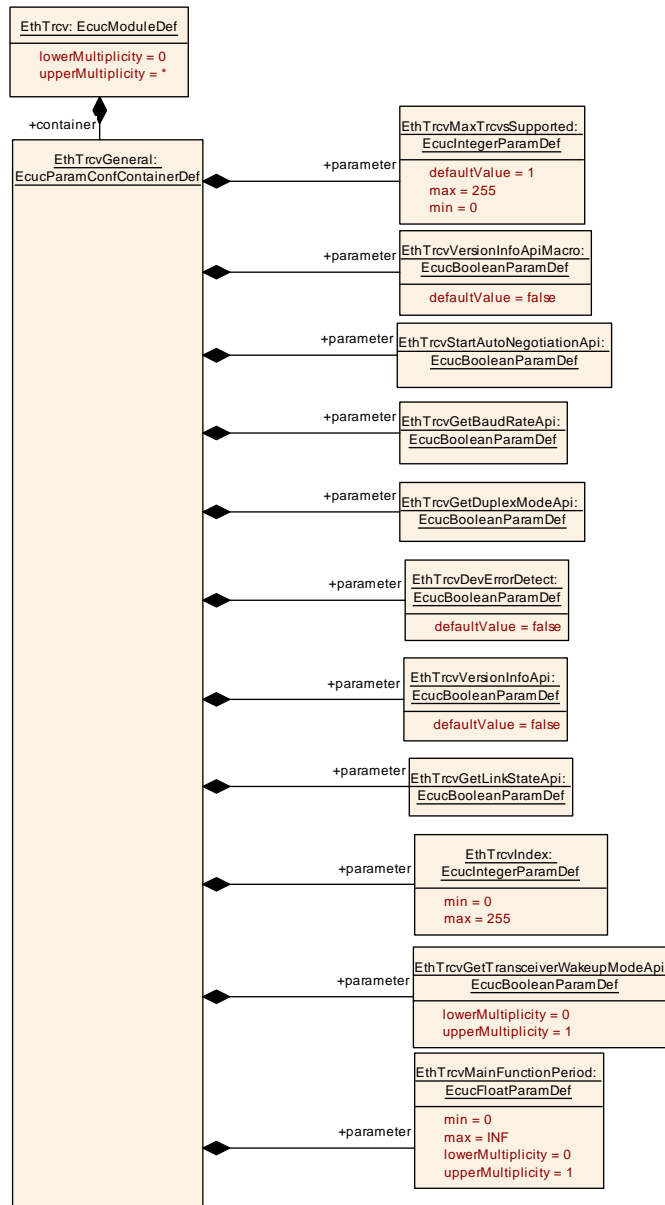


Figure 10.3: EthTrcvGeneral (1/2)

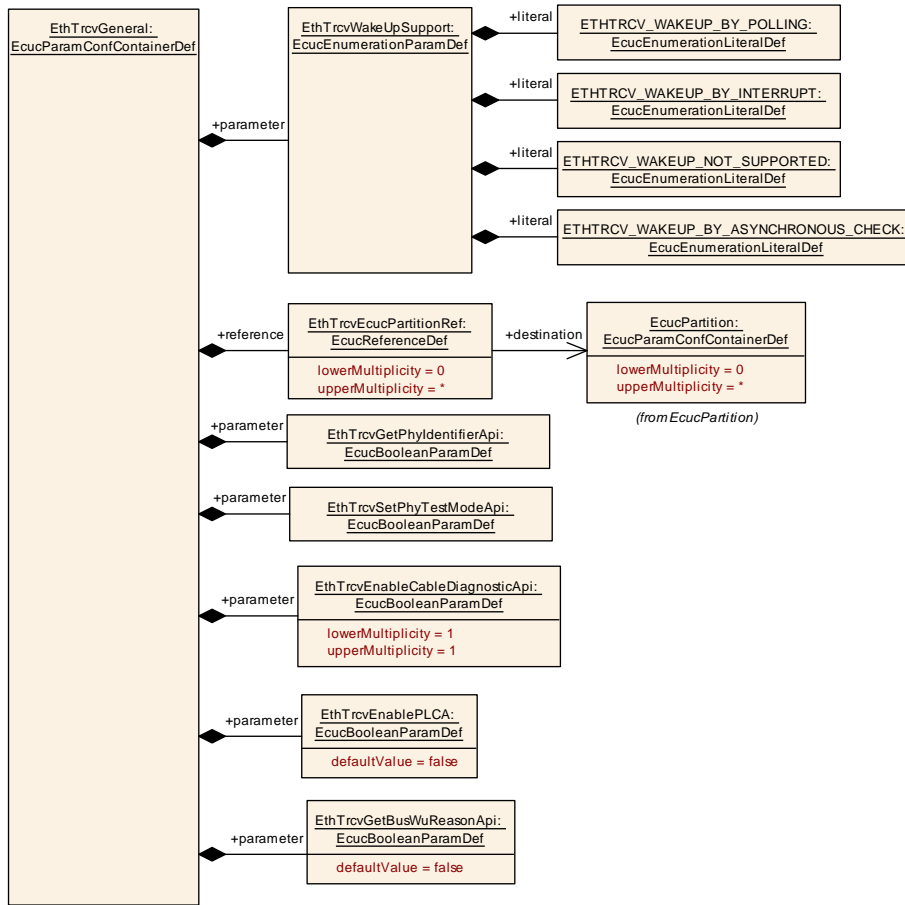


Figure 10.4: EthTrcvGeneral (2/2)

[SWS_EthTrcv_00156] [The module will operate as an independent instance in each of the partitions, means the called API will only target the partition it is called in.] ()

10.3 Published Information

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

A Not applicable requirements