

<b>Document Title</b>	Specification of Ethernet Transceiver Driver
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	431
<b>Document Status</b>	Final
<b>Part of AUTOSAR Standard</b>	Classic Platform
<b>Part of Standard Release</b>	4.4.0

<b>Document Change History</b>			
<b>Date</b>	<b>Release</b>	<b>Changed by</b>	<b>Change Description</b>
2018-10-31	4.4.0	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Explicite transceiver link control</li> <li>• Support of host controllers with multiple cores</li> </ul>
2017-12-08	4.3.1	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Minor corrections and adaptations</li> <li>• Improved Switch integration</li> </ul>
2016-11-30	4.3.0	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Configuration via Switch or Mii</li> <li>• 100BASE-T1 test mode support</li> </ul>
2015-07-31	4.2.2	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• EthTrcv_TransceiverInit functionality merged into EthTrcv_Init API</li> <li>• Development Error Tracer renamed to Default Error Tracer</li> </ul>
2014-10-31	4.2.1	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Change from Synchronous to Asynchronous API</li> <li>• Ethernet Wakeup Support</li> </ul>
2013-10-31	4.1.2	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Introduction of Eth_GeneralTypes.h</li> <li>• Support of API deviation for asynchronous implementation</li> <li>• Editorial changes</li> <li>• Removed chapter(s) on change documentation</li> </ul>
2013-03-15	4.1.1	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• 1000 kbit Ethernet Support</li> </ul>
2011-12-22	4.0.3	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• EthTrcv_GetVersionInfo revised</li> </ul>

<b>Document Change History</b>			
<b>Date</b>	<b>Release</b>	<b>Changed by</b>	<b>Change Description</b>
2010-09-30	3.1.5	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Further post-build configurable parameters</li> <li>• Configuration enhanced by additional parameter EthTrcvWaitCount</li> <li>• 'Instance ID' removed from Version Info (concerns EthTrcv_GetVersionInfo API)</li> <li>• Additional development error in EthTrcv_GetVersionInfo API</li> <li>• Improved description of 'XxxCtrlIdx' semantics</li> <li>• Specification of behaviour for state switch into already active state</li> </ul>
2010-02-02	3.1.4	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Initial Release</li> </ul>

## 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 .....	7
2	Acronyms and abbreviations .....	9
3	Related documentation.....	10
3.1	Input documents .....	10
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 traceability .....	14
7	Functional specification .....	15
7.1	Ethernet BSW stack .....	15
7.1.1	Indexing scheme.....	15
7.1.2	Requirements.....	16
7.1.3	Configuration description .....	17
7.1.4	Wake-up support.....	18
7.1.5	Handling of cable diagnostic .....	18
7.2	Error classification .....	19
7.2.1	Development Errors .....	19
7.2.2	Runtime Errors.....	19
7.2.3	Transient Faults .....	19
7.2.4	Production Errors .....	19
7.2.5	Extended Production Errors .....	19
8	API specification.....	21
8.1	Imported types.....	21
8.2	Type definitions .....	21
8.2.1	EthTrcv_ConfigType .....	21
8.2.2	EthTrcv_ModeType.....	21
8.2.3	EthTrcv_LinkStateType.....	22
8.2.4	EthTrcv_StateType .....	22
8.2.5	EthTrcv_BaudRateType.....	22
8.2.6	EthTrcv_DuplexModeType .....	22
8.2.7	EthTrcv_WakeupModeType.....	22
8.2.8	EthTrcv_WakeupReasonType .....	23
8.2.9	EthTrcv_PhyTestModeType.....	23
8.2.10	EthTrcv_PhyLoopbackModeType.....	23
8.2.11	EthTrcv_PhyTxModeType .....	24
8.2.12	EthTrcv_CableDiagResultType .....	24
8.3	Function definitions.....	24
8.3.1	EthTrcv_Init.....	24

8.3.2	EthTrcv_SetTransceiverMode.....	25
8.3.3	EthTrcv_GetTransceiverMode .....	27
8.3.4	EthTrcv_SetTransceiverWakeupMode .....	28
8.3.5	EthTrcv_GetTransceiverWakeupMode .....	29
8.3.6	EthTrcv_CheckWakeup .....	30
8.3.7	EthTrcv_StartAutoNegotiation.....	31
8.3.8	EthTrcv_TransceiverLinkStateRequest.....	32
8.3.9	EthTrcv_GetLinkState .....	33
8.3.10	EthTrcv_GetBaudRate.....	34
8.3.11	EthTrcv_GetDuplexMode .....	35
8.3.12	EthTrcv_SetPhyTestMode.....	36
8.3.13	EthTrcv_SetPhyLoopbackMode .....	36
8.3.14	EthTrcv_GetPhySignalQuality .....	37
8.3.15	EthTrcv_SetPhyTxMode.....	37
8.3.16	EthTrcv_RunCableDiagnostic.....	38
8.3.17	EthTrcv_GetCableDiagnosticsResult.....	38
8.3.18	EthTrcv_GetPhyIdentifier.....	38
8.3.19	EthTrcv_GetVersionInfo .....	39
8.4	Callback notifications.....	39
8.4.1	EthTrcv_ReadMiiIndication .....	39
8.4.2	EthTrcv_WriteMiiIndication .....	40
8.5	Interrupt service routines .....	40
8.6	Scheduled functions .....	41
8.6.1	EthTrcv_MainFunction .....	41
8.7	Expected Interfaces.....	41
8.7.1	Mandatory Interfaces .....	41
8.7.2	Optional Interfaces.....	41
8.7.3	Configurable interfaces .....	42
9	Sequence diagrams .....	43
10	Configuration specification.....	44
10.1	Containers and configuration parameters .....	45
10.1.1	EthTrcv .....	48
10.1.2	EthTrcvConfigSet .....	48
10.1.3	EthTrcvConfig.....	48
10.1.4	EthTrcvDemEventParameterRefs .....	53
10.1.5	EthTrcvMgmtInterface .....	54
10.1.6	EthTrcvMiiInterface.....	54
10.1.7	EthTrcvSwitchInterface.....	55
10.1.8	EthTrcvGeneral .....	56
11	Not applicable requirements .....	63

## Known Limitations

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

# 1 Introduction and functional overview

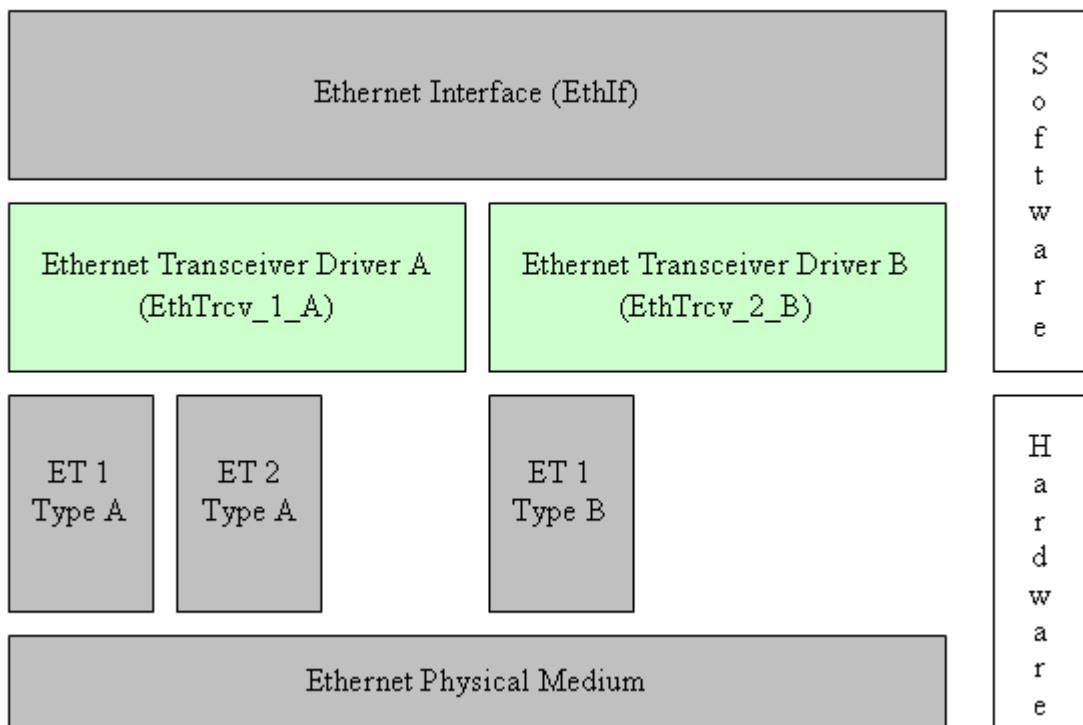
This specification specifies the functionality, API and the configuration of the AUTOSAR Basic Software module Ethernet Transceiver Driver.

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

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

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

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



**Figure 1.1: Ethernet stack module overview**

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



## 2 Acronyms and abbreviations

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

## 3 Related documentation

### 3.1 Input documents

- [1] List of Basic Software Modules  
AUTOSAR\_TR\_BSWModuleList.pdf
- [2] Layered Software Architecture  
AUTOSAR\_EXP\_LayeredSoftwareArchitecture.pdf
- [3] AUTOSAR General Requirements on Basic Software Modules  
AUTOSAR\_SRS\_BSWGeneral.pdf
- [4] Specification of Communication  
AUTOSAR\_SWS\_COM.pdf
- [5] Requirements on Ethernet Support in AUTOSAR  
AUTOSAR\_SRS\_Ethernet.pdf
- [6] Specification of Ethernet Interface  
AUTOSAR\_SWS\_EthernetInterface.pdf
- [7] Specification of Ethernet State Manager  
AUTOSAR\_SWS\_EthernetStateManager.pdf
- [8] Specification of Ethernet Interface  
AUTOSAR\_SWS\_EthernetInterface.pdf
- [9] Specification of Socket Adapter  
AUTOSAR\_SWS\_SocketAdapter.pdf
- [10] Specification of UDP Network Management  
AUTOSAR\_SWS\_UDPNetworkManagement.pdf
- [11] Specification of PDU Router  
AUTOSAR\_SWS\_PDURouter.pdf
- [12] BSW Scheduler Specification  
AUTOSAR\_SWS\_Scheduler.pdf
- [13] Specification of ECU Configuration  
AUTOSAR\_TPS\_ECUConfiguration.pdf
- [14] Specification of Memory Mapping  
AUTOSAR\_SWS\_MemoryMapping.pdf
- [15] Specification of Standard Types  
AUTOSAR\_SWS\_StandardTypes.pdf

[16] Specification of Default Error Tracer  
AUTOSAR\_SWS\_DefaultErrorTracer.pdf

[17] Specification of Diagnostics Event Manager  
AUTOSAR\_SWS\_DiagnosticEventManager

[18] Specification of ECU State Manager  
AUTOSAR\_SWS\_ECUSTateManager.pdf

[19] General Specification of Basic Software Modules  
AUTOSAR\_SWS\_BSWGeneral.pdf

### **3.2 Related standards and norms**

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

[21] IEEE 802.3-2006

### **3.3 Related specification**

AUTOSAR provides a General Specification on Basic Software modules [19] (SWS BSW General), which is also valid for Ethernet Transceiver Driver.

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

## **4 Constraints and assumptions**

### **4.1 Limitations**

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

### **4.2 Applicability to car domains**

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

## 5 Dependencies to other modules

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

Modules that use Ethernet Transceiver Driver module:

- Ethernet Interface (EthIf)

Modules used by the Ethernet Transceiver Driver module:

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

Dependencies to other Modules:

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

## 6 Requirements traceability

Requirement	Description	Satisfied by
SRS_Eth_00106	The Ethernet Transceiver Driver shall switch on/off wake up functionality at pre compile time.	SWS_EthTrcv_00124, SWS_EthTrcv_00139
SRS_Eth_00107	The Ethernet Transceiver Driver shall support access to the wake up reason.	SWS_EthTrcv_00135
SRS_Eth_00108	The Ethernet Transceiver Driver shall be able to wake-up the bus.	SWS_EthTrcv_00118
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

## 7 Functional specification

### 7.1 Ethernet BSW stack

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

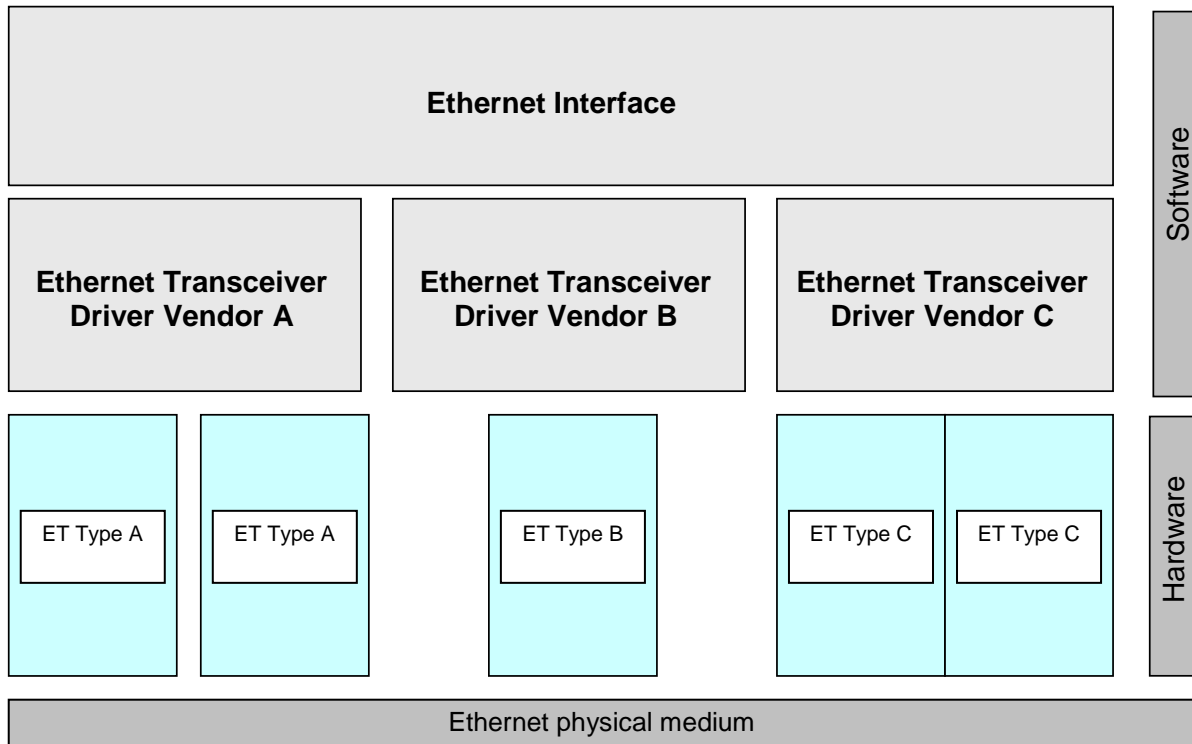
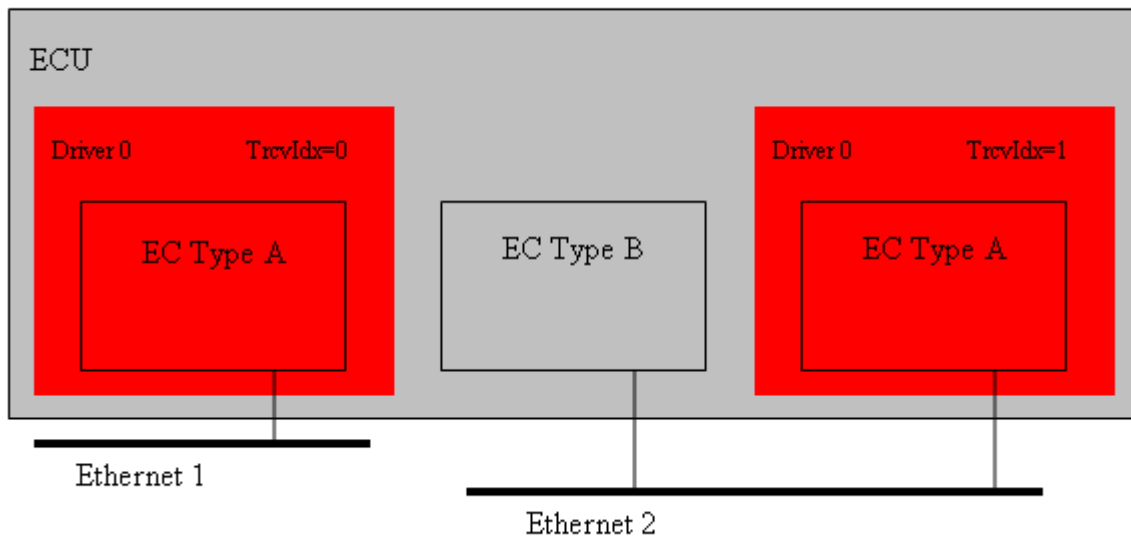


Figure 7.1: Basic Structure of the Ethernet BSW stack

#### 7.1.1 Indexing scheme

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



**Figure 7.2: Ethernet Transceiver Driver indexing scheme**

[SWS\_EthTrcv\_00003] [

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

### 7.1.2 Requirements

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

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

[SWS\_EthTrcv\_00004] [

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

[SWS\_EthTrcv\_00005] [

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

[SWS\_EthTrcv\_00006] [

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

[SWS\_EthTrcv\_00007] [



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

DET API functions are specified in [16].

[SWS\_EthTrcv\_00009] ]

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

[SWS\_EthTrcv\_00010] ]

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

### 7.1.3 Configuration description

[SWS\_EthTrcv\_00011] ]

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

[SWS\_EthTrcv\_00012] ]

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

[SWS\_EthTrcv\_00013] ]

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

[SWS\_EthTrcv\_00014] ]

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

[SWS\_EthTrcv\_00015] ]

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

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

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

#### 7.1.4 Wake-up support

[SWS\_EthTrcv\_00110] [

The Ethernet Transceiver driver shall support wake up depending on the configuration parameter EthTrcvWakeUpSupport either not at all (ETHTRCV\_WAKEUP\_NOT\_SUPPORTED) or by Interrupt (ETHTRCV\_WAKEUP\_BY\_INTERRUPT) or by polling (ETHTRCV\_WAKEUP\_BY\_POLLING). ]()

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

- Sleeping ECU and sleeping bus -> wake up detection via EthTrcv\_Init (called during Power On)
- Awake ECU and sleeping bus -> wake up detection via EthTrcv\_MainFunction or Wake up interrupt handler (checked by EcuM within CheckWakeup)

[SWS\_EthTrcv\_00111] [

If the wake-up mode of the corresponding transceiver is ETHTRCV\_WUM\_ENABLE and transceiver is requested to low power mode (ETHTRCV\_MODE\_DOWN), the transceiver driver shall enable the corresponding ICU channel (see EthTrcvIcuChannelRef) by calling Icu\_EnableNotification. ]()

[SWS\_EthTrcv\_00112] [

If the wake-up mode of the corresponding transceiver is ETHTRCV\_WUM\_ENABLE and transceiver is requested to active (ETHTRCV\_MODE\_ACTIVE), the transceiver driver shall disable the corresponding ICU channel (see EthTrcvIcuChannelRef) by calling Icu\_DisableNotification. ]()

[SWS\_EthTrcv\_00146] [

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

#### 7.1.5 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.

Its up to the caller to re-trigger cable diagnostic again, if the measurement failed by returning ETHTRCV\_CABLEDIAG\_ERROR.

[SWS\_EthTrcv\_00159][

Calling EthTrcv\_RunCableDiagnostic shall trigger cable diagnostic measurement. ]()

[SWS\_EthTrcv\_00160][

Calling EthTrcv\_GetCableDiagnosticsResult shall return the current state of the cable diagnostic measurement. ]()

Note: Cable diagnostic measurement is triggered by a CDD that maintain the cable diagnostic result. For a single Ethernet transceiver (not referenced by Ethernet switch port), the CDD directly calls the EthTrcv APIs. For a Ethernet transceiver which is referenced by a Ethernet switch port, the EthSwt forward the API calls to the EthTrcv. Thus, the upper layer is either a CDD or a Ethernet switch.

## 7.2 Error classification

### 7.2.1 Development Errors

[SWS\_EthTrcv\_00017] [

Type or error	Relevance	Related error code	Value [hex]
Invalid transceiver index	Development error	ETHTRCV_E_INV_TRCV_IDX	0x01
EthTrcv module was not initialized	Development error	ETHTRCV_E_UNINIT	0x02
Invalid pointer in parameter list	Development error	ETHTRCV_E_PARAM_POINTER	0x03

]0

### 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] [

<b>Error Name:</b>	ETHTRCV_E_ACCESS	
<b>Short Description:</b>	Ethernet Transceiver Access Failure.	
<b>Long Description:</b>	Monitors the access to the Ethernet Transceiver.	
<b>Detection Criteria:</b>	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.
<b>Secondary Parameters:</b>	None.	

<i>Time Required:</i>	None.
<i>Monitor Frequency</i>	None.

10

## 8 API specification

### 8.1 Imported types

This chapter lists all types included from the following modules:

#### [SWS\_EthTrcv\_00027] [

Module	Header File	Imported Type
Dem	Rte_Dem_Type.h	Dem_EventIdType
	Rte_Dem_Type.h	Dem_EventStatusType
EcuM	EcuM.h	EcuM_WakeupSourceType
Eth_GeneralTypes	Eth_GeneralTypes.h	EthTrcv_BaudRateType
	Eth_GeneralTypes.h	EthTrcv_CableDiagResultType
	Eth_GeneralTypes.h	EthTrcv_ConfigType
	Eth_GeneralTypes.h	EthTrcv_DuplexModeType
	Eth_GeneralTypes.h	EthTrcv_LinkStateType
	Eth_GeneralTypes.h	EthTrcv_ModeType
	Eth_GeneralTypes.h	EthTrcv_PhyLoopbackModeType
	Eth_GeneralTypes.h	EthTrcv_PhyTestModeType
	Eth_GeneralTypes.h	EthTrcv_PhyTxModeType
	Eth_GeneralTypes.h	EthTrcv_WakeupModeType
Icu	Icu.h	Icu_ChannelType
Std_Types	StandardTypes.h	Std_ReturnType
	StandardTypes.h	Std_VersionInfoType

] ()

### 8.2 Type definitions

#### 8.2.1 EthTrcv\_ConfigType

##### [SWS\_EthTrcv\_00098] [

<b>Name:</b>	EthTrcv_ConfigType
<b>Type:</b>	Structure
<b>Range:</b>	Implementation specific.
<b>Description:</b>	Implementation specific structure of the post build configuration
<b>Available via:</b>	Eth_GeneralTypes.h

] ()

#### 8.2.2 EthTrcv\_ModeType

##### [SWS\_EthTrcv\_00099] [

<b>Name:</b>	EthTrcv_ModeType		
<b>Type:</b>	Enumeration		
<b>Range:</b>	ETHTRCV_MODE_DOWN	0x00	Transceiver disabled
	ETHTRCV_MODE_ACTIVE	0x01	Transceiver enabled
<b>Description:</b>	This type defines the transceiver modes		
<b>Available via:</b>	Eth_GeneralTypes.h		

] ()

### 8.2.3 EthTrcv\_LinkStateType

#### [SWS\_EthTrcv\_00100] [

<b>Name:</b>	EthTrcv_LinkStateType		
<b>Type:</b>	Enumeration		
<b>Range:</b>	ETHTRCV_LINK_STATE_DOWN	0x00	No physical Ethernet connection established
	ETHTRCV_LINK_STATE_ACTIVE	0x01	Physical Ethernet connection established
<b>Description:</b>	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)		
<b>Available via:</b>	Eth_GeneralTypes.h		

] ()

### 8.2.4 EthTrcv\_StateType

#### [SWS\_EthTrcv\_00101] [

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

] ()

### 8.2.5 EthTrcv\_BaudRateType

#### [SWS\_EthTrcv\_00102] [

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

] ()

### 8.2.6 EthTrcv\_DuplexModeType

#### [SWS\_EthTrcv\_00103] [

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

] ()

### 8.2.7 EthTrcv\_WakeupModeType

#### [SWS\_EthTrcv\_00113] [

<b>Name:</b>	EthTrcv_WakeupModeType		
<b>Type:</b>	Enumeration		
<b>Range:</b>	ETHTRCV_WUM_DISABLE	0x00	Transceiver wake up disabled
	ETHTRCV_WUM_ENABLE	0x01	Transceiver wake up enabled
	ETHTRCV_WUM_CLEAR	0x02	Transceiver wake up reason cleared.
<b>Description:</b>	This type controls the transceiver wake up modes and/or clears the wake-up reason.		
<b>Available via:</b>	Eth_GeneralTypes.h		

] ()

## 8.2.8 EthTrcv\_WakeupReasonType

[SWS\_EthTrcv\_00114] [

<b>Name:</b>	EthTrcv_WakeupReasonType		
<b>Type:</b>	Enumeration		
<b>Range:</b>	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_BUS	0x02	Bus wake up detected. Available if 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.
<b>Description:</b>	This type defines the transceiver wake up reasons.		
<b>Available via:</b>	Eth_GeneralTypes.h		

] ()

## 8.2.9 EthTrcv\_PhyTestModeType

[SWS\_EthTrcv\_91002] [

<b>Name:</b>	EthTrcv_PhyTestModeType		
<b>Type:</b>	Enumeration		
<b>Range:</b>	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
<b>Description:</b>	Describes the possible PHY test modes		
<b>Available via:</b>	Eth_GeneralTypes.h		

] (SRS\_Eth\_00117)

## 8.2.10 EthTrcv\_PhyLoopbackModeType

[SWS\_EthTrcv\_91004] [

<b>Name:</b>	EthTrcv_PhyLoopbackModeType		
<b>Type:</b>	Enumeration		
<b>Range:</b>	ETHTRCV_PHYLOOPBACK_NONE	0x00	normal operation

	ETHTRCV_PHYLOOPBACK_INTERNAL	0x01	internal loopback
	ETHTRCV_PHYLOOPBACK_EXTERNAL	0x02	external loopback
	ETHTRCV_PHYLOOPBACK_REMOTE	0x03	remote loopback
<b>Description:</b>	Describes the possible PHY loopback modes		
<b>Available via:</b>	Eth GeneralTypes.h		

](SRS\_Eth\_00117)

### 8.2.11 EthTrcv\_PhyTxModeType

[SWS\_EthTrcv\_91006] [

<b>Name:</b>	EthTrcv_PhyTxModeType		
<b>Type:</b>	Enumeration		
<b>Range:</b>	ETHTRCV_PHYTXMODE_NORMAL	0x00	normal operation
	ETHTRCV_PHYTXMODE_TX_OFF	0x01	transmitter disabled
	ETHTRCV_PHYTXMODE_SCRAMBLER_OFF	0x02	scrambler disabled
<b>Description:</b>	Describes the possible PHY transmit modes		
<b>Available via:</b>	Eth GeneralTypes.h		

](SRS\_Eth\_00117)

### 8.2.12 EthTrcv\_CableDiagResultType

[SWS\_EthTrcv\_91008] [

<b>Name:</b>	EthTrcv_CableDiagResultType		
<b>Type:</b>	Enumeration		
<b>Range:</b>	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
<b>Description:</b>	Describes the results of the cable diagnostics.		
<b>Available via:</b>	Eth GeneralTypes.h		

](SRS\_Eth\_00117)

## 8.3 Function definitions

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

### 8.3.1 EthTrcv\_Init

[SWS\_EthTrcv\_00028] [

<b>Service name:</b>	EthTrcv_Init		
<b>Syntax:</b>	void EthTrcv_Init( const EthTrcv_ConfigType* CfgPtr )		
<b>Service ID[hex]:</b>	0x01		
<b>Sync/Async:</b>	Synchronous		
<b>Reentrancy:</b>	Non Reentrant		
<b>Parameters (in):</b>	CfgPtr	Points to the implementation specific structure	
<b>Parameters</b>	None		



<b>(inout):</b>	
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	Initializes the Ethernet Transceiver Driver
<b>Available via:</b>	EthTrcv.h

] ()

[SWS\_EthTrcv\_00029] [

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

[SWS\_EthTrcv\_00035] [

The function shall:

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

[SWS\_EthTrcv\_00030] [

The function shall change the state of the component from ETHTRCV\_STATE\_UNINIT to ETHTRCV\_STATE\_INIT. ]()

[SWS\_EthTrcv\_00115] [

If the wake-up mode of the corresponding transceiver is ETHTRCV\_WUM\_ENABLE the function shall check for wake-up reasons and propagate the corresponding wake-up source (see EthTrcvWakeupMap configuration) to the EcuM by calling EcuM\_SetWakeupEvent. ]()

[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 and return E\_NOT\_OK, otherwise pass the production error ETHTRCV\_E\_ACCESS and return E\_OK. ]()

[SWS\_EthTrcv\_00032] [

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

### 8.3.2 EthTrcv\_SetTransceiverMode

[SWS\_EthTrcv\_00042] [

<b>Service name:</b>	EthTrcv_SetTransceiverMode	
<b>Syntax:</b>	Std_ReturnType EthTrcv_SetTransceiverMode( uint8 TrcvIdx, EthTrcv_ModeType CtrlMode )	
<b>Service ID[hex]:</b>	0x03	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
	CtrlMode	ETHTRCV_MODE_DOWN: disable the transceiver ETHTRCV_MODE_ACTIVE: enable the transceiver
<b>Parameters (inout):</b>	None	

<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: Service accepted E_NOT_OK: Service denied
<b>Description:</b>	Enables / disables the indexed transceiver	
<b>Available via:</b>	EthTrcv.h	

] ()

[SWS\_EthTrcv\_00043] ]

The function shall put the index transceiver in the specified mode and indicate the new mode by the API EthIf\_TrcvModeIndication latest during the next EthTrcv\_MainFunction. ]()

[SWS\_EthTrcv\_00117] ]

If the wake up mode of the corresponding transceiver is ETHTRCV\_WUM\_ENABLE and the function is called with ETHTRCV\_MODE\_DOWN, it shall set the transceiver into a mode (e.g. sleep mode) where wakeups can be detected. ]()

[SWS\_EthTrcv\_00118] ]

If EthTrcv\_SetTransceiverMode() is called with parameter ETHTRCV\_MODE\_ACTIVE, the Ethernet Transceiver driver shall  
 (\*) check for wake-up reasons when entering the transceiver's active mode.  
 (\*) In case 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\_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 otherwise (if DET is disabled) return E\_NOT\_OK. ]()

[SWS\_EthTrcv\_00045] ]

If development error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV\_E\_INV\_TRCV\_IDX otherwise (if DET is disabled) return E\_NOT\_OK. ]()

[SWS\_EthTrcv\_00046] ]

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

[SWS\_EthTrcv\_00094] ]

If the transceiver is already in the requested mode E\_OK shall be returned and no development error shall be raised. ]()

[SWS\_EthTrcv\_00104] ]

The function 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] [

Caveat: The function requires previous transceiver initialization (EthTrcv\_Init). ]()

### 8.3.3 EthTrcv\_GetTransceiverMode

[SWS\_EthTrcv\_00048] [

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

] () [SWS\_EthTrcv\_00049] [

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

[SWS\_EthTrcv\_00050] [

If development error detection is enabled: the function shall check that the service EthTrcv\_Init was previously called. If the check fails, the function shall raise the development error ETHTRCV\_E\_UNINIT otherwise (if DET is disabled) return E\_NOT\_OK. ]()

[SWS\_EthTrcv\_00051] [

If development error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV\_E\_INV\_TRCV\_IDX otherwise (if DET is disabled) return E\_NOT\_OK. ]()

[SWS\_EthTrcv\_00052] [

If development error detection is enabled: the function shall check the parameter TrcvModePtr for being valid. If the check fails, the function shall raise the development error ETHTRCV\_E\_PARAM\_POINTER otherwise (if DET is disabled) return E\_NOT\_OK. ]()

[SWS\_EthTrcv\_00053] [

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

[SWS\_EthTrcv\_00054] [

Caveat: The function requires previous transceiver initialization (EthTrcv\_Init). ]()

### 8.3.4 EthTrcv\_SetTransceiverWakeupMode

[SWS\_EthTrcv\_00119] [

<b>Service name:</b>	EthTrcv_SetTransceiverWakeupMode	
<b>Syntax:</b>	Std_ReturnType EthTrcv_SetTransceiverWakeupMode (uint8 TrcvIdx, EthTrcv_WakeupModeType TrcvWakeupMode)	
<b>Service ID[hex]:</b>	0x0d	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
	TrcvWakeupMode	ETHTRCV_WUM_DISABLE: disable transceiver wake up ETHTRCV_WUM_ENABLE: enable transceiver wake up ETHTRCV_WUM_CLEAR: clears transceiver wake up reason
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: transceiver wake up mode has been changed. E_NOT_OK: transceiver wake up mode could not be changed or the wake-up reason could not be cleared.
<b>Description:</b>	Enables / disables the wake-up mode or clear the wake-up reason of the indexed transceiver	
<b>Available via:</b>	EthTrcv.h	

] ()

[SWS\_EthTrcv\_00120] [

If function EthTrcv\_SetTransceiverWakeupMode() is called with ETHTRCV\_WUM\_DISABLE or ETHTRCV\_WUM\_ENABLE it shall put the indexed transceiver in the specified wake up mode. ]()

[SWS\_EthTrcv\_00121] [

If function EthTrcv\_SetTransceiverWakeupMode() is called with ETHTRCV\_WUM\_CLEAR it shall clear stored wakeup events on the indexed transceiver. ]()

[SWS\_EthTrcv\_00122] [

If development error detection is enabled: The function EthTrcv\_SetTransceiverWakeupMode() shall check that the service EthTrcv\_Init was previously called. If the check fails, the function shall raise the development error ETHTRCV\_E\_UNINIT otherwise (if DET is disabled) return E\_NOT\_OK. ]()

[SWS\_EthTrcv\_00123] [

If development error detection is enabled: The function EthTrcv\_SetTransceiverWakeupMode() shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV\_E\_INV\_TRCV\_IDX otherwise (if DET is disabled) return E\_NOT\_OK. ]()

[SWS\_EthTrcv\_00124] [

The function `EthTrcv_SetTransceiverWakeupMode()` shall be only available if `EthTrcvWakeUpSupport` is not disabled (set to `ETHTRCV_WAKEUP_NOT_SUPPORTED`). ] (SRS\_Eth\_00106)

[SWS\_EthTrcv\_00125] ]

If the transceiver is already in the requested wake-up mode, `E_OK` shall be returned and no development error shall be raised. ]()

[SWS\_EthTrcv\_00126] ]

Caveat: The function `EthTrcv_SetTransceiverWakeupMode()` requires previous transceiver initialization (`EthTrcv_Init`). ]()

### 8.3.5 EthTrcv\_GetTransceiverWakeupMode

[SWS\_EthTrcv\_00127] ]

<b>Service name:</b>	EthTrcv_GetTransceiverWakeupMode	
<b>Syntax:</b>	Std_ReturnType EthTrcv_GetTransceiverWakeupMode (uint8 TrcvIdx, EthTrcv_WakeupModeType* TrcvWakeupModePtr)	
<b>Service ID[hex]:</b>	0x0e	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	TrcvWakeupModePtr	ETHTRCV_WUM_DISABLE: transceiver wake up is disabled ETHTRCV_WUM_ENABLE: transceiver wake up is enabled
<b>Return value:</b>	Std_ReturnType	E_OK: success E_NOT_OK: transceiver wake up mode could not be obtained
<b>Description:</b>	Returns the wake up mode of the indexed transceiver	
<b>Available via:</b>	EthTrcv.h	

] ()

[SWS\_EthTrcv\_00128] ]

The function `EthTrcv_GetTransceiverWakeupMode()` shall read the current transceiver wake up mode and provide it into `TrcvWakeupModePtr`. ]()

[SWS\_EthTrcv\_00129] ]

If development error detection is enabled: The function `EthTrcv_GetTransceiverWakeupMode()` shall check that the service `EthTrcv_Init` was previously called. If the check fails, the function shall raise the development error `ETHTRCV_E_UNINIT` otherwise (if DET is disabled) return `E_NOT_OK`. ]()

[SWS\_EthTrcv\_00130] ]

If development error detection is enabled: The function `EthTrcv_GetTransceiverWakeupMode()` shall check the parameter `TrcvIdx` for being valid. If the check fails, the function shall raise the development error `ETHTRCV_E_INV_TRCV_IDX` otherwise (if DET is disabled) return `E_NOT_OK`. ]()

[SWS\_EthTrcv\_00131] ]

If development error detection is enabled: The function `EthTrcv_GetTransceiverWakeupMode()` shall check the parameter `TrcvWakeupModePtr` for being valid. If the check fails, the function shall raise the development error `ETHTRCV_E_PARAM_POINTER` otherwise (if DET is disabled) return `E_NOT_OK`. `()`

[SWS\_EthTrcv\_00132] `()`

The function `EthTrcv_GetTransceiverWakeupMode()` shall be only available if `EthTrcvGetTransceiverWakeupModeApi` is set to `TRUE`. `()`

[SWS\_EthTrcv\_00133] `()`

Caveat: The function `EthTrcv_GetTransceiverWakeupMode()` requires previous transceiver initialization (`EthTrcv_Init`). `()`

### 8.3.6 EthTrcv\_CheckWakeup

[SWS\_EthTrcv\_00134] `()`

<b>Service name:</b>	EthTrcv_CheckWakeup	
<b>Syntax:</b>	Std_ReturnType EthTrcv_CheckWakeup( uint8 TrcvIdx )	
<b>Service ID[hex]:</b>	0x0f	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: The function has been successfully executed E_NOT_OK: The function could not be successfully executed
<b>Description:</b>	Service is called by EthIf in case a wake-up interrupt is detected.	
<b>Available via:</b>	EthTrcv.h	

`()`

[SWS\_EthTrcv\_00135] `()`

If the wake-up mode of the corresponding transceiver is `ETHTRCV_WUM_ENABLE` 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) to the EcuM by calling `EcuM_SetWakeupEvent`. `(SRS_Eth_00107)`

[SWS\_EthTrcv\_00136] `()`

If the wake-up mode of the corresponding transceiver is not `ETHTRCV_WUM_ENABLE`, the function `EthTrcv_CheckWakeup()` shall return `E_OK`. `()`

[SWS\_EthTrcv\_00137] `()`

If development error detection is enabled: The function `EthTrcv_CheckWakeup()` shall check that the service `EthTrcv_Init` was previously called. If the check fails, the function shall raise the development error `ETHTRCV_E_UNINIT` otherwise (if DET is disabled) return `E_NOT_OK`. `()`

[SWS\_EthTrcv\_00138] [

If development error detection is enabled: The function EthTrcv\_CheckWakeup() shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV\_E\_INV\_TRCV\_IDX otherwise (if DET is disabled) return E\_NOT\_OK. ]()

[SWS\_EthTrcv\_00139] [

The function EthTrcv\_CheckWakeup() shall be only available if EthTrcvWakeUpSupport is something else than ETHTRCV\_WAKEUP\_NOT\_SUPPORTED. ](SRS\_Eth\_00106)

[SWS\_EthTrcv\_00140] [

Caveat: The function EthTrcv\_CheckWakeup() requires previous transceiver initialization (EthTrcv\_Init). ]()

### 8.3.7 EthTrcv\_StartAutoNegotiation

[SWS\_EthTrcv\_00055] [

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

] () [SWS\_EthTrcv\_00056] [

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

[SWS\_EthTrcv\_00057] [

If development error detection is enabled: the function shall check that the service EthTrcv\_Init was previously called. If the check fails, the function shall raise the development error ETHTRCV\_E\_UNINIT otherwise (if DET is disabled) return E\_NOT\_OK. ]()

[SWS\_EthTrcv\_00058] [

If development error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV\_E\_INV\_TRCV\_IDX otherwise (if DET is disabled) return E\_NOT\_OK. ]()



[SWS\_EthTrcv\_00059] [

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

[SWS\_EthTrcv\_00060] [

Caveat: The function requires previous transceiver initialization (EthTrcv\_Init). ]()

[SWS\_EthTrcv\_00088] [

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

### 8.3.8 EthTrcv\_TransceiverLinkStateRequest

[SWS\_EthTrcv\_91025] [

<b>Service name:</b>	EthTrcv_TransceiverLinkStateRequest	
<b>Syntax:</b>	Std_ReturnType EthTrcv_TransceiverLinkStateRequest (uint8 TrcvIdx, EthTrcv_LinkStateType LinkState)	
<b>Service ID[hex]:</b>	--	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.	
<b>Parameters (in):</b>	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
	LinkState	The Ethernet link state of a physical Ethernet connection.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: The request has been accepted E_NOT_OK: The request has not been accepted
<b>Description:</b>	Request the given link state for the given Ethernet transceiver	
<b>Available via:</b>	EthTrcv.h	

] () [SWS\_EthTrcv\_00151] [

The function shall start link training of the indexed transceiver if the requested link state is ETHTRCV\_LINK\_STATE\_ACTIVE 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\_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. ]()



### 8.3.9 EthTrcv\_GetLinkState

#### [SWS\_EthTrcv\_00061] [

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

#### ] () [SWS\_EthTrcv\_00062] [

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

#### [SWS\_EthTrcv\_00063] [

If development error detection is enabled: the function shall check that the service EthTrcv\_Init was previously called. If the check fails, the function shall raise the development error ETHTRCV\_E\_UNINIT otherwise (if DET is disabled) return E\_NOT\_OK. ]()

#### [SWS\_EthTrcv\_00064] [

If development error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV\_E\_INV\_TRCV\_IDX otherwise (if DET is disabled) return E\_NOT\_OK. ]()

#### [SWS\_EthTrcv\_00065] [

If development error detection is enabled: the function shall check the parameter LinkStatePtr for being valid. If the check fails, the function shall raise the development error ETHTRCV\_E\_PARAM\_POINTER otherwise (if DET is disabled) return E\_NOT\_OK. ]()

#### [SWS\_EthTrcv\_00066] [

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

#### [SWS\_EthTrcv\_00067] [

Caveat: The function requires previous transceiver initialization (EthTrcv\_Init). ]()

### 8.3.10 EthTrcv\_GetBaudRate

#### [SWS\_EthTrcv\_00068] [

<b>Service name:</b>	EthTrcv_GetBaudRate	
<b>Syntax:</b>	Std_ReturnType EthTrcv_GetBaudRate( uint8 TrcvIdx, EthTrcv_BaudRateType* BaudRatePtr )	
<b>Service ID[hex]:</b>	0x07	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	BaudRatePtr	ETHTRCV_BAUD_RATE_10MBIT: 10MBit connection ETHTRCV_BAUD_RATE_100MBIT: 100MBit connection ETHTRCV_BAUD_RATE_1000MBIT: 1000MBit connection
<b>Return value:</b>	Std_ReturnType	E_OK: success E_NOT_OK: transceiver could not be initialized
<b>Description:</b>	Obtains the baud rate of the indexed transceiver	
<b>Available via:</b>	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 otherwise (if DET is disabled) return E\_NOT\_OK. ]()

#### [SWS\_EthTrcv\_00071] [

If development error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV\_E\_INV\_TRCV\_IDX otherwise (if DET is disabled) return E\_NOT\_OK. ]()

#### [SWS\_EthTrcv\_00072] [

If development error detection is enabled: the function shall check the parameter BaudRatePtr for being valid. If the check fails, the function shall raise the development error ETHTRCV\_E\_PARAM\_POINTER otherwise (if DET is disabled) return E\_NOT\_OK. ]()

#### [SWS\_EthTrcv\_00073] [

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

#### [SWS\_EthTrcv\_00074] [

Caveat: The function requires previous transceiver initialization (EthTrcv\_Init). ]()

#### [SWS\_EthTrcv\_00089] [

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

### 8.3.11 EthTrcv\_GetDuplexMode

#### [SWS\_EthTrcv\_00075] [

<b>Service name:</b>	EthTrcv_GetDuplexMode	
<b>Syntax:</b>	Std_ReturnType EthTrcv_GetDuplexMode ( uint8 TrcvIdx, EthTrcv_DuplexModeType* DuplexModePtr )	
<b>Service ID[hex]:</b>	0x08	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	DuplexModePtr	ETHTRCV_DUPLEX_MODE_HALF: half duplex connections ETHTRCV_DUPLEX_MODE_FULL: full duplex connection
<b>Return value:</b>	Std_ReturnType	E_OK: success E_NOT_OK: transceiver could not be initialized
<b>Description:</b>	Obtains the duplex mode of the indexed transceiver	
<b>Available via:</b>	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 otherwise (if DET is disabled) return E\_NOT\_OK. ]()

#### [SWS\_EthTrcv\_00078] [

If development error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV\_E\_INV\_TRCV\_IDX otherwise (if DET is disabled) return E\_NOT\_OK. ]()

#### [SWS\_EthTrcv\_00079] [

If development error detection is enabled: the function shall check the parameter DuplexModePtr for being valid. If the check fails, the function shall raise the development error ETHTRCV\_E\_PARAM\_POINTER otherwise (if DET is disabled) return E\_NOT\_OK. ]()

#### [SWS\_EthTrcv\_00080] [

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

#### [SWS\_EthTrcv\_00081] [

Caveat: The function requires previous transceiver initialization (EthTrcv\_Init). ]()

[SWS\_EthTrcv\_00090] [

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

### 8.3.12 EthTrcv\_SetPhyTestMode

[SWS\_EthTrcv\_91003] [

<b>Service name:</b>	EthTrcv_SetPhyTestMode	
<b>Syntax:</b>	Std_ReturnType EthTrcv_SetPhyTestMode ( uint8 TrcvIdx, EthTrcv_PhyTestModeType Mode )	
<b>Service ID[hex]:</b>	0x11	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.	
<b>Parameters (in):</b>	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
	Mode	Test mode to be activated
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: The request has been accepted E_NOT_OK: The request has not been accepted.
<b>Description:</b>	Activates a given test mode.	
<b>Available via:</b>	EthTrcv.h	

] (SRS\_Eth\_00117)

[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)

### 8.3.13 EthTrcv\_SetPhyLoopbackMode

[SWS\_EthTrcv\_91005] [

<b>Service name:</b>	EthTrcv_SetPhyLoopbackMode	
<b>Syntax:</b>	Std_ReturnType EthTrcv_SetPhyLoopbackMode ( uint8 TrcvIdx, EthTrcv_PhyLoopbackModeType Mode )	
<b>Service ID[hex]:</b>	0x12	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.	
<b>Parameters (in):</b>	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
	Mode	Loopback mode to be activated
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: The request has been accepted E_NOT_OK: The request has not been accepted.

<b>Description:</b>	Activates a given loopback mode.
<b>Available via:</b>	EthTrcv.h

] (SRS\_Eth\_00117)

[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)

### 8.3.14 EthTrcv\_GetPhySignalQuality

[SWS\_EthTrcv\_91001] [

<b>Service name:</b>	EthTrcv_GetPhySignalQuality	
<b>Syntax:</b>	Std_ReturnType EthTrcv_GetPhySignalQuality( uint8 TrcvIdx, uint32* SignalQualityPtr )	
<b>Service ID[hex]:</b>	0x10	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.	
<b>Parameters (in):</b>	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	SignalQualityPtr	Pointer to the memory where the signal quality shall be stored.
<b>Return value:</b>	Std_ReturnType	E_OK: The request has been accepted E_NOT_OK: The request has not been accepted.
<b>Description:</b>	Obtains the current signal quality of the link of the indexed transceiver	
<b>Available via:</b>	EthTrcv.h	

] (SRS\_Eth\_00117)

### 8.3.15 EthTrcv\_SetPhyTxMode

[SWS\_EthTrcv\_91007] [

<b>Service name:</b>	EthTrcv_SetPhyTxMode	
<b>Syntax:</b>	Std_ReturnType EthTrcv_SetPhyTxMode( uint8 TrcvIdx, EthTrcv_PhyTxModeType Mode )	
<b>Service ID[hex]:</b>	0x13	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.	
<b>Parameters (in):</b>	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
	Mode	Transmission mode to be activated
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: The request has been accepted E_NOT_OK: The request has not been accepted
<b>Description:</b>	Activates a given transmission mode.	
<b>Available via:</b>	EthTrcv.h	

] (SRS\_Eth\_00117)

[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.] ()

### 8.3.16 EthTrcv\_RunCableDiagnostic

[SWS\_EthTrcv\_91011]

<b>Service name:</b>	EthTrcv_RunCableDiagnostic	
<b>Syntax:</b>	Std_ReturnType EthTrcv_RunCableDiagnostic( uint8 TrcvIdx )	
<b>Service ID[hex]:</b>	0x16	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.	
<b>Parameters (in):</b>	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Transceiver Driver.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: The trigger has been accepted. E_NOT_OK: The trigger has not been accepted.
<b>Description:</b>	Trigger the cable diagnostics for the given Ethernet transceiver.	
<b>Available via:</b>	EthTrcv.h	

] ()

### 8.3.17 EthTrcv\_GetCableDiagnosticsResult

[SWS\_EthTrcv\_91009]

<b>Service name:</b>	EthTrcv_GetCableDiagnosticsResult	
<b>Syntax:</b>	Std_ReturnType EthTrcv_GetCableDiagnosticsResult( uint8 TrcvIdx, EthTrcv_CableDiagResultType* ResultPtr )	
<b>Service ID[hex]:</b>	0x14	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.	
<b>Parameters (in):</b>	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	ResultPtr	Pointer to the location where the cable diagnostics result shall be stored
<b>Return value:</b>	Std_ReturnType	E_OK: The request has been accepted E_NOT_OK: The request has not been accepted
<b>Description:</b>	Retrieves the cable diagnostics result of a given transceiver.	
<b>Available via:</b>	EthTrcv.h	

] (SRS\_Eth\_00117)

### 8.3.18 EthTrcv\_GetPhyIdentifier

[SWS\_EthTrcv\_91010]

<b>Service name:</b>	EthTrcv_GetPhyIdentifier	
<b>Syntax:</b>	Std_ReturnType EthTrcv_GetPhyIdentifier( uint8 TrcvIdx,	

	<pre>uint32* OrgUniqueIdPtr, uint8* ModelNrPtr, uint8* RevisionNrPtr )</pre>	
<b>Service ID[hex]:</b>	0x15	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.	
<b>Parameters (in):</b>	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	E_OK: The request has been accepted E_NOT_OK: The request has not been accepted
<b>Description:</b>	Obtains the PHY identifier of the Ethernet Transceiver according to IEEE 802.3-2015 chapter 22.2.4.3.1 PHY Identifier.	
<b>Available via:</b>	EthTrcv.h	

] (SRS\_Eth\_00117)

### 8.3.19 EthTrcv\_GetVersionInfo

[SWS\_EthTrcv\_00082] [

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

## 8.4 Callback notifications

### 8.4.1 EthTrcv\_ReadMiiIndication

[SWS\_EthTrcv\_00108] [



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

] ()

## 8.4.2 EthTrcv\_WriteMiiIndication

[SWS EthTrcv\_00109] [

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

] ()

## 8.5 Interrupt service routines

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



## 8.6 Scheduled functions

### 8.6.1 EthTrcv\_MainFunction

[SWS\_EthTrcv\_00106] [

<b>Service name:</b>	EthTrcv_MainFunction
<b>Syntax:</b>	void EthTrcv_MainFunction( void )
<b>Service ID[hex]:</b>	0x0c
<b>Description:</b>	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.
<b>Available via:</b>	SchM_EthTrcv.h

] () [SWS\_EthTrcv\_00107] [

Used for polling state changes. Calls EthIf\_TrcvModeIndication when the transceiver mode changed. ]()

[SWS\_EthTrcv\_00141] [

The function EthTrcv\_MainFunction() shall check for wake up reasons and shall store wakeup events if EthTrcvWakeUpSupport is set to ETHTRCV\_WAKEUP\_BY\_POLLING. ]()

## 8.7 Expected Interfaces

This chapter lists all interfaces required from other modules.

### 8.7.1 Mandatory Interfaces

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

[SWS\_EthTrcv\_00085] [

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.
EthIf_TrcvModeIndication	EthIf.h	Called asynchronously when a mode change has been read out. If the function is triggered by previous call of EthTrcv_SetTransceiverMode it can directly be called within the trigger function.
SchM_Enter_EthTrcv	SchM_<Mip>.h	Invokes the SchM_Enter function to enter a module local exclusive area.
SchM_Exit_EthTrcv	SchM_<Mip>.h	Invokes the SchM_Exit function to exit an exclusive area.

] ()

### 8.7.2 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
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.

] ()

### 8.7.3 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] [

<b>Service name:</b>	<EthTrcvWakeUpCallout>	
<b>Syntax:</b>	void <EthTrcvWakeUpCallout>(uint8 TrcvIdx)	
<b>Service ID[hex]:</b>	0x11	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant Dont care	
<b>Parameters (in):</b>	TrcvIdx	Index of the Ethernet Transceiver
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	Indicates an wake-up request for the specified Ethernet Transceiver. Can be used to trigger integrator code that initiates a remote wake-up.	
<b>Available via:</b>	EthTrcv_Externals.h	

] ()

[SWS\_EthTrcv\_00145] [

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

## 9 Sequence diagrams

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

## 10 Configuration specification

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

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

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

## 10.1 Containers and configuration parameters

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

[SWS\_EthTrcv\_00155] DRAFT [

The Ethernet Transceiver Driver module shall reject configurations with partition mappings which are not supported by the implementation.] ()

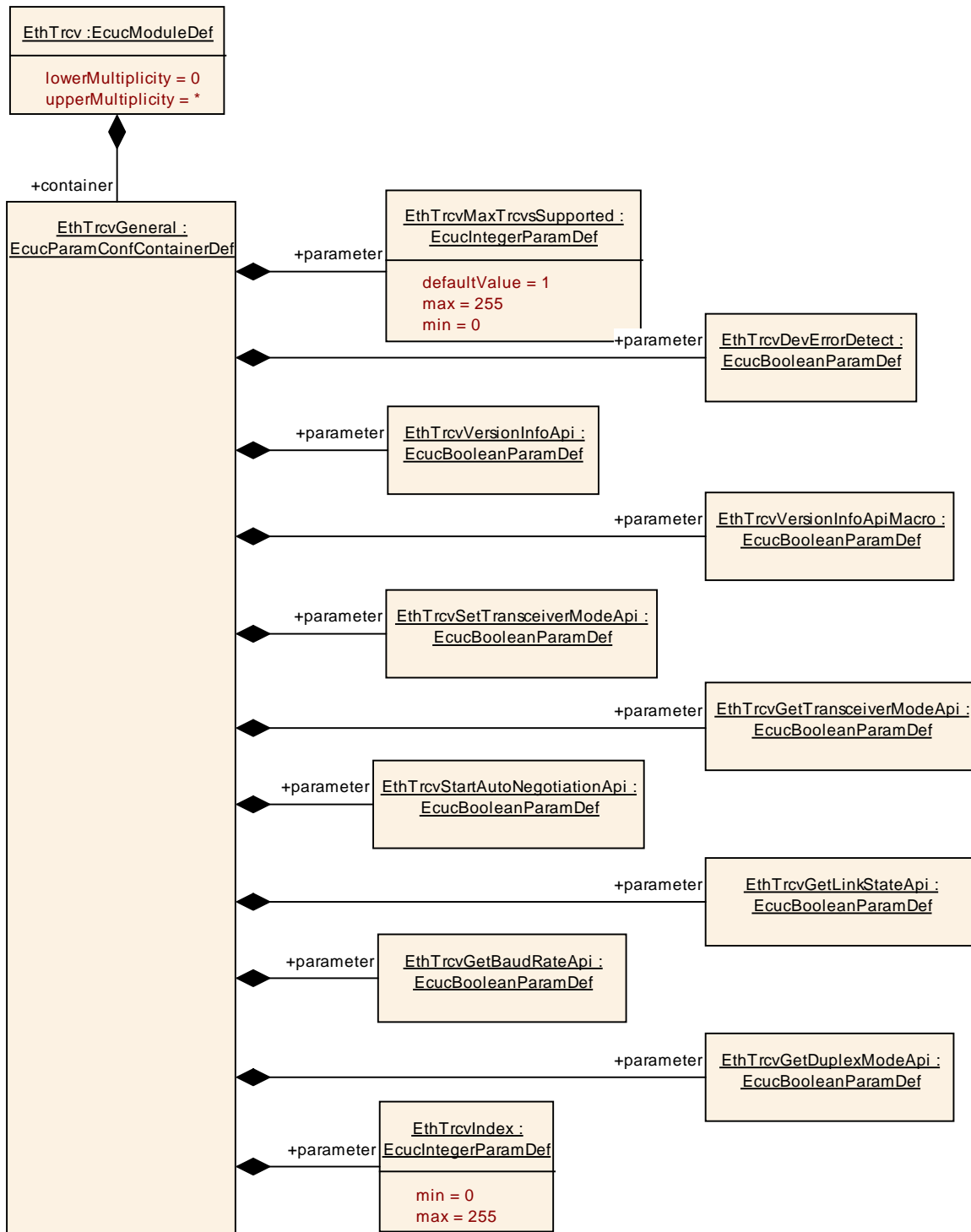


Figure 10.1: Ethernet Transceiver Driver configuration structure

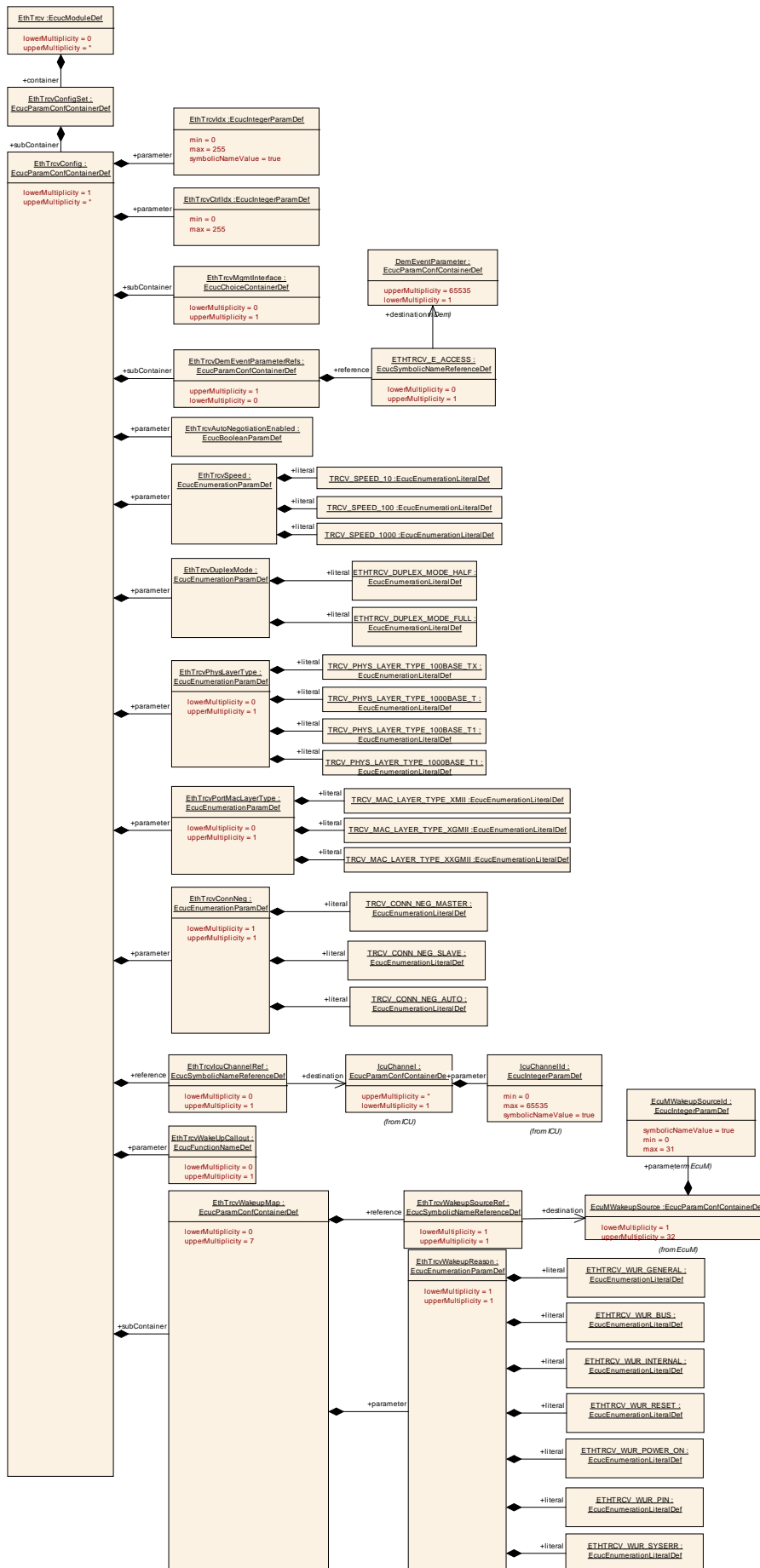


Figure 10.2: Ethernet Transceiver Driver Transceiver configuration structure

### 10.1.1 EthTrcv

<b>SWS Item</b>	<b>ECUC_EthTrcv_00034 :</b>
<b>Module Name</b>	EthTrcv
<b>Module Description</b>	Configuration of Ethernet Transceiver Driver module
<b>Post-Build Variant Support</b>	true
<b>Supported Config Variants</b>	VARIANT-LINK-TIME, VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
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

### 10.1.2 EthTrcvConfigSet

<b>SWS Item</b>	<b>ECUC_EthTrcv_00016 :</b>
<b>Container Name</b>	EthTrcvConfigSet
<b>Description</b>	This container contains the configuration parameters and sub containers of the AUTOSAR EthTrcv module.
<b>Configuration Parameters</b>	

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

### 10.1.3 EthTrcvConfig

<b>SWS Item</b>	<b>ECUC_EthTrcv_00012 :</b>
<b>Container Name</b>	EthTrcvConfig
<b>Description</b>	Configuration of the individual transceiver
<b>Configuration Parameters</b>	

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

<b>SWS Item</b>	<b>ECUC_EthTrcv_00025 :</b>
<b>Name</b>	EthTrcvConnNeg
<b>Parent Container</b>	EthTrcvConfig
<b>Description</b>	Specifies the connection negotiation of the Ethernet transceiver link.



<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	TRCV_CONN_NEG_AUTO	Automatic Negotiation	
	TRCV_CONN_NEG_MASTER	Master	
	TRCV_CONN_NEG_SLAVE	Slave	
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

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

<b>SWS Item</b>	<b>ECUC_EthTrcv_00013 :</b>		
<b>Name</b>	EthTrcvIdx		
<b>Parent Container</b>	EthTrcvConfig		
<b>Description</b>	Specifies the instance ID of the configured transceiver.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>	0 .. 255		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: ECU		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00044 :</b>		
<b>Name</b>	EthTrcvMacLayerSpeed		
<b>Parent Container</b>	EthTrcvConfig		
<b>Description</b>	Defines the baud rate of the MAC layer.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	ETH_MAC_LAYER_SPEED_100M	--	
	ETH_MAC_LAYER_SPEED_10G	--	
	ETH_MAC_LAYER_SPEED_10M	--	
	ETH_MAC_LAYER_SPEED_1G	--	
<b>Post-Build Variant Multiplicity</b>	true		

<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: ECU		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00043 :</b>		
<b>Name</b>	EthTrcvMacLayerSubType		
<b>Parent Container</b>	EthTrcvConfig		
<b>Description</b>	Defines the MAC layer subtype of a switch port		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	LIGHT	--	
	REDUCED	--	
	REVERSED	--	
	SERIAL	--	
	STANDARD	--	
	UNIVERSAL_SERIAL	--	
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: ECU		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00035 :</b>		
<b>Name</b>	EthTrcvMacLayerType		
<b>Parent Container</b>	EthTrcvConfig		
<b>Description</b>	Defines the MAC layer type of the ethernet transceiver.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	TRCV_MAC_LAYER_TYPE_XGMII		MAC layer interface (data) bandwidth class 1Gbit/s (e.g. GMII, RGMII, SGMII, RvGMII, USGMII)
	TRCV_MAC_LAYER_TYPE_XMII		MAC layer interface (data) bandwidth class 100Mbit/s (e.g. RMII, RvMII, SMII, RvMII)
	TRCV_MAC_LAYER_TYPE_XXGMII		MAC layer interface (data) bandwidth class 10Gbit/s
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		

<b>Value</b>			
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: ECU		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00024 :</b>		
<b>Name</b>	EthTrcvPhysLayerType		
<b>Parent Container</b>	EthTrcvConfig		
<b>Description</b>	Specifies the physical layer type of the Ethernet transceiver link.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	TRCV_PHYS_LAYER_TYPE_1000BASE_T	physical layer interface 1000BASE-T (1Gbit/s, 4 pairs). Used for consumer electronic.	
	TRCV_PHYS_LAYER_TYPE_1000BASE_T1	physical layer interface 1000BASE-T1 (1Gbit/s, 1 pair). Used for automotive.	
	TRCV_PHYS_LAYER_TYPE_100BASE_T1	physical layer interface 100BASE-T1 (100Mbit/s, 1 pair). Used for automotive.	
	TRCV_PHYS_LAYER_TYPE_100BASE_TX	physical layer interface 100BASE-TX (100Mbit/s, 2 pairs). Used for consumer electronic.	
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00022 :</b>		
<b>Name</b>	EthTrcvSpeed		
<b>Parent Container</b>	EthTrcvConfig		
<b>Description</b>	Specifies the speed of the Ethernet transceiver link in [MBit/s]. If AutoNegotiation is enabled this is the maximum speed advertised for Auto-Negotiation.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	TRCV_SPEED_10	10 MBit/s	
	TRCV_SPEED_100	100 MBit/s	
	TRCV_SPEED_1000	1000 MBit/s	
<b>Post-Build Variant Value</b>	true		
<b>Value</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE

<b>Configuration Class</b>	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local dependency: EthTrcvAutoNegotiationEnabled		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00028 :</b>		
<b>Name</b>	EthTrcvWakeUpCallout		
<b>Parent Container</b>	EthTrcvConfig		
<b>Description</b>	Configuration of the call-out name.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default value</b>	--		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	dependency: Only valid if EthTrcvWakeUpSupport is not ETHTRCV_WAKEUP_NOT_SUPPORTED.		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00051 :</b>		
<b>Name</b>	EthTrcvConfigEcucPartitionRef		
<b>Parent Container</b>	EthTrcvConfig		
<b>Description</b>	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. <b>Tags:</b> atp.Status=draft		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to [ EcucPartition ]		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: ECU		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00026 :</b>		
<b>Name</b>	EthTrcvIcuChannelRef		
<b>Parent Container</b>	EthTrcvConfig		
<b>Description</b>	Reference to the IcuChannel to enable/disable the interrupts for wakeups.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Symbolic name reference to [ IcuChannel ]		
<b>Post-Build Variant Multiplicity</b>	false		

<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
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.

[SWS\_EthTrcv\_00157] DRAFT [

The ECUC partitions referenced by EthTrcvConfigEcucPartitionRef shall be a subset of the ECUC partitions referenced by EthTrcvEcucPartitionRef.] ()

[SWS\_EthTrcv\_00158] DRAFT [

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

#### 10.1.4 EthTrcvDemEventParameterRefs

<b>SWS Item</b>	<b>ECUC_EthTrcv_00017 :</b>
<b>Container Name</b>	EthTrcvDemEventParameterRefs
<b>Description</b>	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.
<b>Configuration Parameters</b>	

<b>SWS Item</b>	<b>ECUC_EthTrcv_00018 :</b>		
<b>Name</b>	ETHTRCV_E_ACCESS		
<b>Parent Container</b>	EthTrcvDemEventParameterRefs		
<b>Description</b>	Reference to the DemEventParameter which shall be issued when the error "Transceiver access failed" has occurred.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Symbolic name reference to [ DemEventParameter ]		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE

<b>Class</b>	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

**No Included Containers**

### 10.1.5 EthTrcvMgmtInterface

<b>SWS Item</b>	<b>ECUC_EthTrcv_00036 :</b>		
<b>Choice container Name</b>	EthTrcvMgmtInterface		
<b>Description</b>	The choice container allow to configure either the EthTrcv is accessed by a MII interface or Switch interface.		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	

<b>Container Choices</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
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.
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.

### 10.1.6 EthTrcvMiiInterface

<b>SWS Item</b>	<b>ECUC_EthTrcv_00037 :</b>		
<b>Container Name</b>	EthTrcvMiiInterface		
<b>Description</b>	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.		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>ECUC_EthTrcv_00014 :</b>		
<b>Name</b>	EthTrcvCtrlIdx		
<b>Parent Container</b>	EthTrcvMiiInterface		
<b>Description</b>	Specifies the controller used for MII access to the transceiver		
<b>Multiplicity</b>	1		
<b>Type</b>	EcuIntegerParamDef		

<b>Range</b>	0 .. 255		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00038 :</b>		
<b>Name</b>	EthTrcvMiiIdx		
<b>Parent Container</b>	EthTrcvMiiInterface		
<b>Description</b>	Specifies the transceiver index used for MII access to the transceiver.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

**No Included Containers**

### 10.1.7 EthTrcvSwitchInterface

<b>SWS Item</b>	<b>ECUC_EthTrcv_00040 :</b>		
<b>Container Name</b>	EthTrcvSwitchInterface		
<b>Description</b>	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.		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Configuration Parameters</b>			

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

<b>SWS Item</b>	<b>ECUC_EthTrcv_00041 :</b>		
<b>Name</b>	EthTrcvSwitchRef		



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

**No Included Containers**

### 10.1.8 EthTrcvGeneral

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

<b>SWS Item</b>	<b>ECUC_EthTrcv_00003 :</b>		
<b>Name</b>	EthTrcvDevErrorDetect		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Switches the development error detection and notification on or off. <ul style="list-style-type: none"> <li>true: detection and notification is enabled.</li> <li>false: detection and notification is disabled.</li> </ul>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00054 :</b>		
<b>Name</b>	EthTrcvEnableCableDiagnosticApi		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Enable/disable the APIs for cable diagnostic: EthTrcv_RunCableDiagnostic, EthTrcv_GetCableDiagnosticsResult		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00010 :</b>		
<b>Name</b>	EthTrcvGetBaudRateApi		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Enables / Disables EthTrcv_GetBaudRate API		



<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00052 :</b>		
<b>Name</b>	EthTrcvGetCableDiagnosticsResultApi		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Enables / Disables EthTrcv_GetCableDiagnosticsResult API.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00011 :</b>		
<b>Name</b>	EthTrcvGetDuplexModeApi		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Enables / Disables EthTrcv_GetDuplexMode API		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00009 :</b>		
<b>Name</b>	EthTrcvGetLinkStateApi		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Enables / Disables EthTrcv_GetLinkState API		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00046 :</b>		
<b>Name</b>	EthTrcvGetPhylIdentifierApi		

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

<b>SWS Item</b>	<b>ECUC_EthTrcv_00045 :</b>		
<b>Name</b>	EthTrcvGetPhySignalQualityApi		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Enables / Disables EthTrcv_GetPhySignalQuality API.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00007 :</b>		
<b>Name</b>	EthTrcvGetTransceiverModeApi		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Enables / Disables EthTrcv_GetTransceiverMode API		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00031 :</b>		
<b>Name</b>	EthTrcvGetTransceiverWakeupModeApi		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Enables / Disables EthTrcv_GetTransceiverWakeupMode API		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		

<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local dependency: Only valid if EthTrcvWakeUpSupport is not ETHTRCV_WAKEUP_NOT_SUPPORTED		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00020 :</b>		
<b>Name</b>	EthTrcvIndex		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00032 :</b>		
<b>Name</b>	EthTrcvMainFunctionPeriod		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Specifies the period of main function EthTrcv_MainFunction in seconds.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	]0 .. INF[		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00002 :</b>		
<b>Name</b>	EthTrcvMaxTrcvsSupported		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	--		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default value</b>	1		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	

	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00047 :</b>		
<b>Name</b>	EthTrcvSetPhyTestModeApi		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Enables / Disables EthTrcv_SetPhyTestMode API.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00048 :</b>		
<b>Name</b>	EthTrcvSetPhyTxModeApi		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Enables / Disables EthTrcv_SetPhyTxMode API.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00006 :</b>		
<b>Name</b>	EthTrcvSetTransceiverModeApi		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Enables / Disables EthTrcv_SetTransceiverMode API		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00008 :</b>		
<b>Name</b>	EthTrcvStartAutoNegotiationApi		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Enables / Disables EthTrcv_StartAutoNegotiation API		

<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00004 :</b>		
<b>Name</b>	EthTrcvVersionInfoApi		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Enables / Disables version info API		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00005 :</b>		
<b>Name</b>	EthTrcvVersionInfoApiMacro		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Enables / Disables version info API macro implementation		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00030 :</b>		
<b>Name</b>	EthTrcvWakeUpSupport		
<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	Configures wake-up to polling or interrupt or to not used/not supported. In case no wake up is supported by the hardware, the BSWMD pre-configuration shall be set to ETHTRCV_WAKEUP_NOT_SUPPORTED.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	ETHTRCV_WAKEUP_BY_INTERRUPT	Wake up by interrupt	
	ETHTRCV_WAKEUP_BY_POLLING	Wake up by polling	
	ETHTRCV_WAKEUP_NOT_SUPPORTED	Wake up is not supported	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_EthTrcv_00050 :</b>		
<b>Name</b>	EthTrcvEcucPartitionRef		

<b>Parent Container</b>	EthTrcvGeneral		
<b>Description</b>	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. <b>Tags:</b> atp.Status=draft		
<b>Multiplicity</b>	0..*		
<b>Type</b>	Reference to [ EcucPartition ]		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: ECU		

**No Included Containers**

[SWS\_EthTrcv\_00156] DRAFT [

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.] ()

## 11 Not applicable requirements

### [SWS\_EthTrcv\_00999]

These requirements are not applicable to this specification (BSW00170).