Document Title	Specification of Ethernet Transceiver Driver
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	431
Document Classification	Standard
Document Status	Final
Part of AUTOSAR Release	4.2.2

Document Change History		
Release	**Changed by**	**Change Description**
4.2.2	AUTOSAR Release Management	• EthTrcv_TransceiverInit functionality merged into EthTrcv_Init API • Development Error Tracer renamed to Default Error Tracer
4.2.1	AUTOSAR Release Management	• Change from Synchronous to Asynchronous API • Ethernet Wakeup Support
4.1.2	AUTOSAR Release Management	• Introduction of Eth_GeneralTypes.h • Support of API deviation for asynchronous implementation • Editorial changes • Removed chapter(s) on change documentation
4.1.1	AUTOSAR Administration	• 1000 kbit Ethernet Support
4.0.3	AUTOSAR Administration	• EthTrcv_GetVersionInfo revised
3.1.5	AUTOSAR Administration	• Further post-build configurable parameters • Configuration enhanced by additional parameter EthTrcvWaitCount • 'Instance ID' removed from Version Info (concerns EthTrcv_GetVersionInfo API) • Additional development error in EthTrcv_GetVersionInfo API • Improved description of 'XxxCtrlIdx' semantics • Specification of behaviour for state switch into already active state
3.1.4	AUTOSAR Administration	• Initial Release

Disclaimer

Advice for users

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

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

Table of Contents

Known Limitations

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

1 Introduction and functional overview

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

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

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

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

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

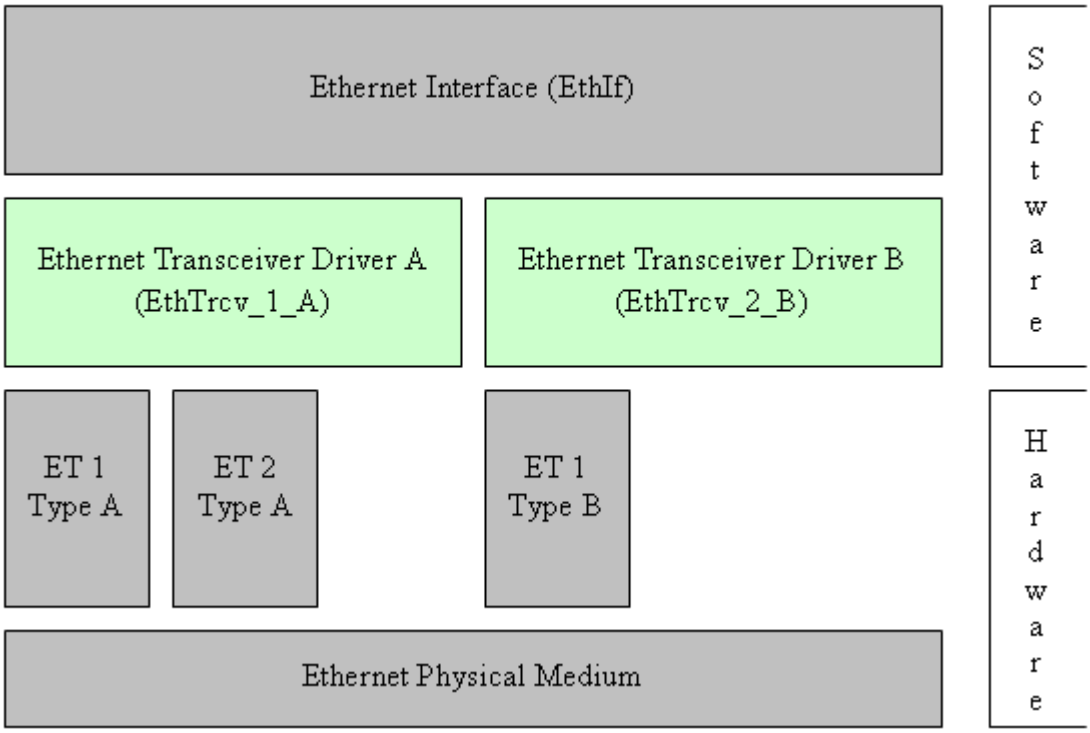

Figure 1.1: Ethernet stack module overview

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

2 Acronyms and abbreviations

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

3 Related documentation

3.1 Input documents

[1] List of Basic Software Modules
AUTOSAR_TR_BSWModuleList.pdf

[2] Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf

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

[4] Specification of Communication
AUTOSAR_SWS_COM.pdf

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

[6] Specification of Ethernet Interface
AUTOSAR_SWS_EthernetInterface.pdf

[7] Specification of Ethernet State Manager
AUTOSAR_SWS_EthernetStateManager.pdf

[8] Specification of Ethernet Interface
AUTOSAR_SWS_EthernetInterface.pdf

[9] Specification of Socket Adapter
AUTOSAR_SWS_SocketAdapter.pdf

[10] Specification of UDP Network Management
AUTOSAR_SWS_UDPNetworkManagement.pdf

[11] Specification of PDU Router
AUTOSAR_SWS_PDURouter.pdf

[12] BSW Scheduler Specification
AUTOSAR_SWS_Scheduler.pdf

[13] Specification of ECU Configuration
AUTOSAR_TPS_ECUConfiguration.pdf

[14] Specification of Memory Mapping
AUTOSAR_SWS_MemoryMapping.pdf

[15] Specification of Standard Types
AUTOSAR_SWS_StandardTypes.pdf

[16] Specification of Default Error Tracer
AUTOSAR_SWS_ DefaultErrorTracer.pdf

[17] Specification of Diagnostics Event Manager
AUTOSAR_SWS_DiagnosticEventManager

[18] Specification of C Implementation Rules
AUTOSAR_TR_CImplementationRules.pdf

[19] Specification of ECU State Manager
AUTOSAR_SWS_ECUStateManager.pdf

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

3.2 Related standards and norms

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

[21] IEEE 802.3-2006

3.3 Related specification

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

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

4 Constraints and assumptions

4.1 Limitations

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

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

4.2 Applicability to car domains

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

5 Dependencies to other modules

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

Modules that use Ethernet Transceiver Driver module:

- Ethernet Interface (EthIf)

Modules used by the Ethernet Transceiver Driver module:

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

Dependencies to other Modules:

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

5.1 File structure

5.1.1 Header file structure

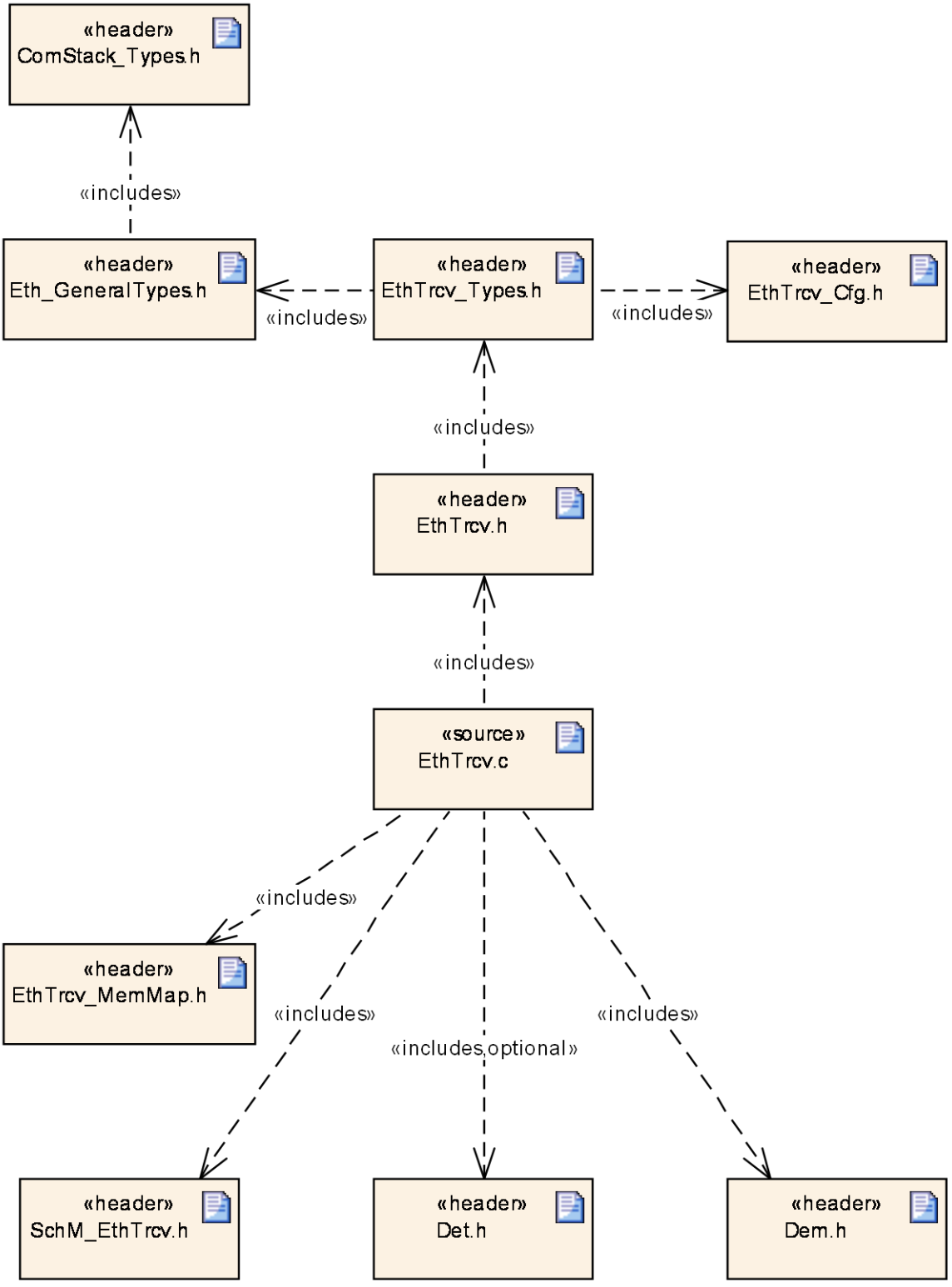

Figure 5.1: Ethernet Transceiver Driver file structure

6 Requirements traceability

Requirement	Description	Satisfied by
-	-	SWS_EthTrcv_00003
-	-	SWS_EthTrcv_00004
-	-	SWS_EthTrcv_00005
-	-	SWS_EthTrcv_00006
-	-	SWS_EthTrcv_00007
-	-	SWS_EthTrcv_00008
-	-	SWS_EthTrcv_00009
-	-	SWS_EthTrcv_00010
-	-	SWS_EthTrcv_00011
-	-	SWS_EthTrcv_00012
-	-	SWS_EthTrcv_00013
-	-	SWS_EthTrcv_00014
-	-	SWS_EthTrcv_00015
-	-	SWS_EthTrcv_00017
-	-	SWS_EthTrcv_00027
-	-	SWS_EthTrcv_00028
-	-	SWS_EthTrcv_00029
-	-	SWS_EthTrcv_00030
-	-	SWS_EthTrcv_00032
-	-	SWS_EthTrcv_00035
-	-	SWS_EthTrcv_00040
-	-	SWS_EthTrcv_00042
-	-	SWS_EthTrcv_00043
-	-	SWS_EthTrcv_00044
-	-	SWS_EthTrcv_00045
-	-	SWS_EthTrcv_00046
-	-	SWS_EthTrcv_00047
-	-	SWS_EthTrcv_00048
-	-	SWS_EthTrcv_00049
-	-	SWS_EthTrcv_00050
-	-	SWS_EthTrcv_00051
-	-	SWS_EthTrcv_00052
-	-	SWS_EthTrcv_00053
-	-	SWS_EthTrcv_00054
-	-	SWS_EthTrcv_00055

-	-	SWS_EthTrcv_00056
-	-	SWS_EthTrcv_00057
-	-	SWS_EthTrcv_00058
-	-	SWS_EthTrcv_00059
-	-	SWS_EthTrcv_00060
-	-	SWS_EthTrcv_00061
-	-	SWS_EthTrcv_00062
-	-	SWS_EthTrcv_00063
-	-	SWS_EthTrcv_00064
-	-	SWS_EthTrcv_00065
-	-	SWS_EthTrcv_00066
-	-	SWS_EthTrcv_00067
-	-	SWS_EthTrcv_00068
-	-	SWS_EthTrcv_00069
-	-	SWS_EthTrcv_00070
-	-	SWS_EthTrcv_00071
-	-	SWS_EthTrcv_00072
-	-	SWS_EthTrcv_00073
-	-	SWS_EthTrcv_00074
-	-	SWS_EthTrcv_00075
-	-	SWS_EthTrcv_00076
-	-	SWS_EthTrcv_00077
-	-	SWS_EthTrcv_00078
-	-	SWS_EthTrcv_00079
-	-	SWS_EthTrcv_00080
-	-	SWS_EthTrcv_00081
-	-	SWS_EthTrcv_00082
-	-	SWS_EthTrcv_00085
-	-	SWS_EthTrcv_00086
-	-	SWS_EthTrcv_00088
-	-	SWS_EthTrcv_00089
-	-	SWS_EthTrcv_00090
-	-	SWS_EthTrcv_00093
-	-	SWS_EthTrcv_00094
-	-	SWS_EthTrcv_00095
-	-	SWS_EthTrcv_00096
-	-	SWS_EthTrcv_00098
-	-	SWS_EthTrcv_00099
-	-	SWS_EthTrcv_00100

-	-	SWS_EthTrcv_00101
-	-	SWS_EthTrcv_00102
-	-	SWS_EthTrcv_00103
-	-	SWS_EthTrcv_00104
-	-	SWS_EthTrcv_00105
-	-	SWS_EthTrcv_00106
-	-	SWS_EthTrcv_00107
-	-	SWS_EthTrcv_00108
-	-	SWS_EthTrcv_00109
-	-	SWS_EthTrcv_00110
-	-	SWS_EthTrcv_00111
-	-	SWS_EthTrcv_00112
-	-	SWS_EthTrcv_00113
-	-	SWS_EthTrcv_00114
-	-	SWS_EthTrcv_00115
-	-	SWS_EthTrcv_00117
-	-	SWS_EthTrcv_00119
-	-	SWS_EthTrcv_00120
-	-	SWS_EthTrcv_00121
-	-	SWS_EthTrcv_00122
-	-	SWS_EthTrcv_00123
-	-	SWS_EthTrcv_00125
-	-	SWS_EthTrcv_00126
-	-	SWS_EthTrcv_00127
-	-	SWS_EthTrcv_00128
-	-	SWS_EthTrcv_00129
-	-	SWS_EthTrcv_00130
-	-	SWS_EthTrcv_00131
-	-	SWS_EthTrcv_00132
-	-	SWS_EthTrcv_00133
-	-	SWS_EthTrcv_00134
-	-	SWS_EthTrcv_00136
-	-	SWS_EthTrcv_00137
-	-	SWS_EthTrcv_00138
-	-	SWS_EthTrcv_00140
-	-	SWS_EthTrcv_00141
-	-	SWS_EthTrcv_00142
-	-	SWS_EthTrcv_00144
-	-	SWS_EthTrcv_00145

-	-	SWS_EthTrcv_00146
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

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.

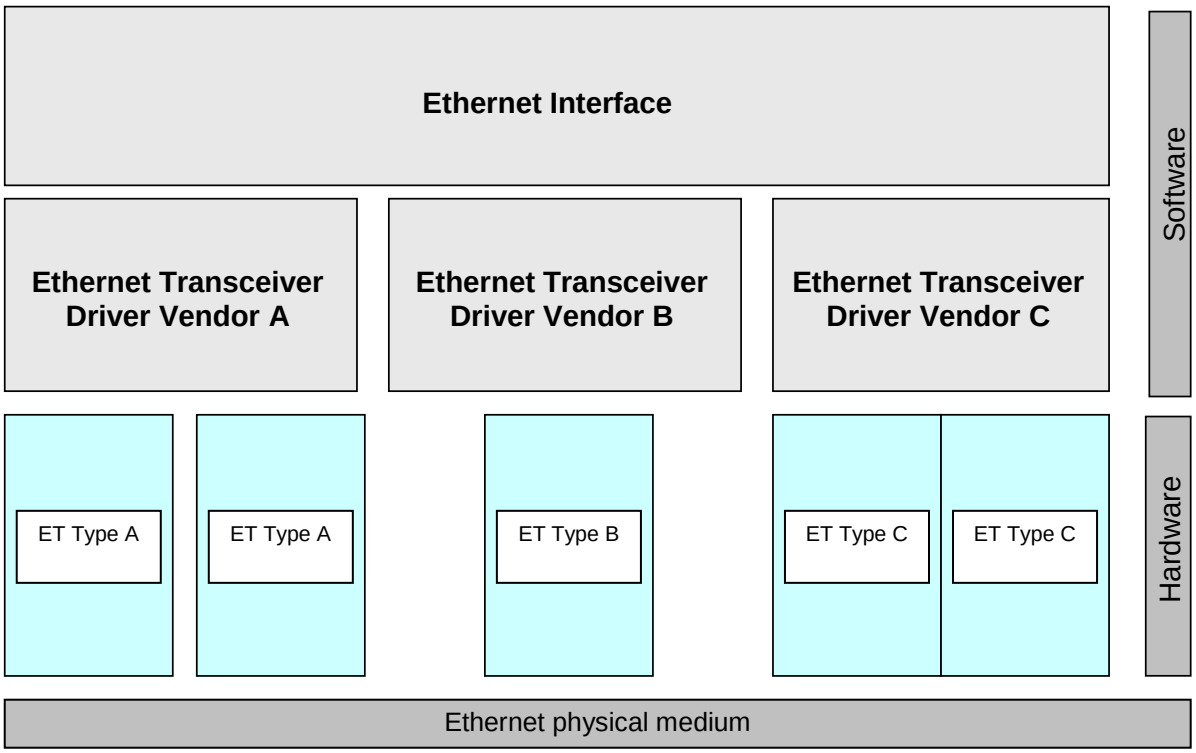

Figure 7.1: Basic Structure of the Ethernet BSW stack

7.1.1 Indexing scheme

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

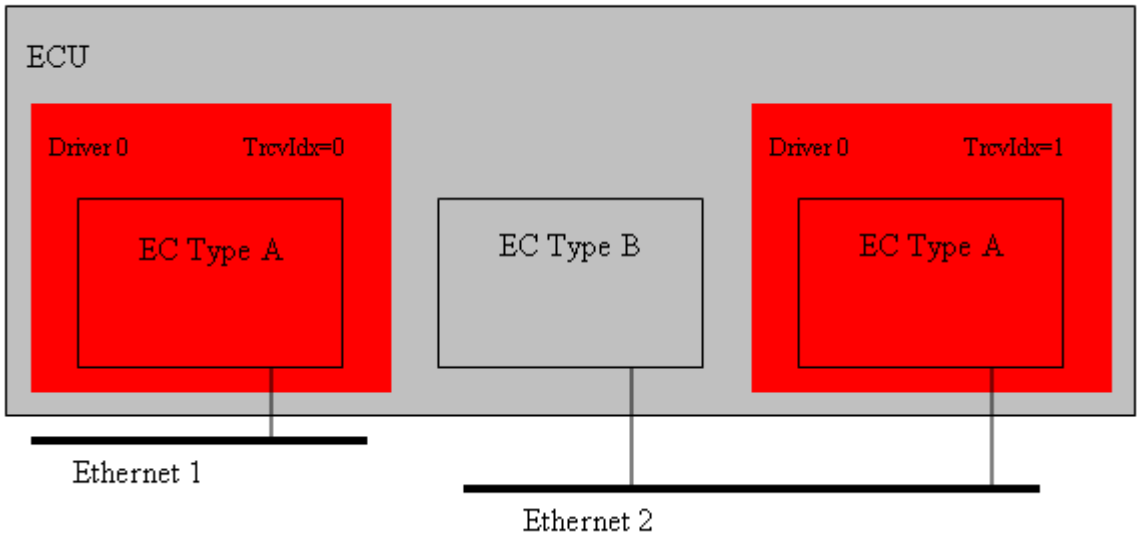

Figure 7.2: Ethernet Transceiver Driver indexing scheme

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

DET API functions are specified in [16].

[SWS_EthTrcv_00008] [
The Ethernet Transceiver Driver module implementation shall conform to the HIS subset of the MISRA C Standard (see document [18]).]()

[SWS_EthTrcv_00009] [
The Ethernet Transceiver Driver module shall implement the API functions specified by the Ethernet Transceiver Driver SWS as real C-code functions and shall not implement the API as macros for object code deliveries.]()

[SWS_EthTrcv_00010] [
None of the Ethernet Transceiver Driver module header files shall define global variables.]()

7.1.3 Configuration description

[SWS_EthTrcv_00011] [
The Ethernet Transceiver Driver module shall provide an XML file that contains the data, which is required for the SW identification (it shall contain the vendor identification, module ID and software version information), configuration and integration process. This file should describe vendor specific configuration parameters as well as it should contain recommended configuration parameter values.]()

[SWS_EthTrcv_00012] [
The MCG shall read the ECU configuration description of the Ethernet Driver module(s). Ethernet Driver related configuration data is contained in the Ethernet Driver module configuration description.]()

[SWS_EthTrcv_00013] [
The MCG shall ensure the consistency of the generated configuration data.]()

[SWS_EthTrcv_00014] [
The configuration of the Ethernet Transceiver Driver module shall be calculated at ECU configuration time. None of the communication parameters shall be calculated at runtime.]()

[SWS_EthTrcv_00015] [
The start address of post-build time configuration data shall be passed during module initialization (see chapter 8.3.1).]()

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

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

7.1.4 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.]()

7.2 Error classification

7.2.1 Default Errors

[SWS_EthTrcv_00017] [

Type or error	*Relevance*	*Related error code*	*Value [hex]*
Invalid transceiver index	Default error	ETHTRCV_E_INV_TRCV_IDX	0x01
EthTrcv module was not initialized	Default error	ETHTRCV_E_NOT_INITIALIZED	0x02
Invalid pointer in	Default error	ETHTRCV_E_PARAM_POINTER	0x03

parameter list			

]()

7.2.2 Runtime Errors

There are no runtime errors.

7.2.3 Transient Faults

There are no transient faults.

7.2.4 Production Errors

There are no production errors.

7.2.5 Extended Production Errors

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

[SWS_EthTrcv_00105] [

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

]()

8 API specification

8.1 Imported types

This chapter lists all types included from the following files:

[SWS_EthTrcv_00027] [

Module	*Imported Type*
ComStack_Types	BufReq_ReturnType
Dem	Dem_EventIdType
	Dem_EventStatusType
EcuM	EcuM_WakeupSourceType
Eth_GeneralTypes	EthTrcv_BaudRateType
	EthTrcv_ConfigType
	EthTrcv_DuplexModeType
	EthTrcv_LinkStateType
	EthTrcv_ModeType
	EthTrcv_WakeupModeType
	Eth_BufIdxType
	Eth_ConfigType
	Eth_FrameType
	Eth_ModeType
	Eth_RxStatusType
Icu	Icu_ChannelType
Std_Types	Std_ReturnType
	Std_VersionInfoType

]()

8.2 Type definitions

[SWS_EthTrcv_00095] [
EthTrcv.h shall include Eth_GeneralTypes.h for include of general EthTrcv type declarations.]()

[SWS_EthTrcv_00096] [
The types specified in SWS_EthernetTransceiverDriver shall be declared in Eth_GeneralTypes.h.]()

8.2.1 EthTrcv_ConfigType

[SWS_EthTrcv_00098] [

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

]()

8.2.2 EthTrcv_ModeType

[SWS_EthTrcv_00099] [

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

](

8.2.3 EthTrcv_LinkStateType

[SWS_EthTrcv_00100][

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

](

8.2.4 EthTrcv_StateType

[SWS_EthTrcv_00101] [

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

](

8.2.5 EthTrcv_BaudRateType

[SWS_EthTrcv_00102] [

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

](

8.2.6 EthTrcv_DuplexModeType

[SWS_EthTrcv_00103] [

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

](

8.2.7 EthTrcv_ WakeupModeType

[SWS_EthTrcv_00113] [

Name:	EthTrcv_WakeupModeType	
Type:	Enumeration	
Range:	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.
Description:	This type controls the transceiver wake up modes and/or clears the wake-up reason.	

]()

8.2.8 EthTrcv_ WakeupReasonType

[SWS_EthTrcv_00114] [

Name:	EthTrcv_WakeupReasonType	
Type:	Enumeration	
Range:	ETHTRCV_WUR_NONE	0x00: No wake up reason detected.
	ETHTRCV_WUR_GENERAL	0x01: General wake up detected, no distinct reason supported by hardware.
	ETHTRCV_WUR_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.
Description:	This type defines the transceiver wake up reasons.	

]()

8.3 Function definitions

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

8.3.1 EthTrcv_Init

[SWS_EthTrcv_00028] [

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

](

[SWS_EthTrcv_00029] [
The function shall store the access to the configuration structure for subsequent API calls.](

[SWS_EthTrcv_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] [

Service name:	EthTrcv_SetTransceiverMode	
Syntax:	`Std_ReturnType EthTrcv_SetTransceiverMode( uint8 TrcvIdx, EthTrcv_ModeType CtrlMode )`	
Service ID[hex]:	0x03	
Sync/Async:	Asynchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
	CtrlMode	ETHTRCV_MODE_DOWN: disable the transceiver ETHTRCV_MODE_ACTIVE: enable the transceiver
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: Service accepted E_NOT_OK: Service denied
Description:	Enables / disables the indexed transceiver	

⌋()

[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 default 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 default error ETHTRCV_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ⌋()

[SWS_EthTrcv_00045] ⌈
If default error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the default 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 default 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] ⌈

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

⌋()

[SWS_EthTrcv_00049] ⌈
The function shall read the current transceiver mode. ⌋()

[SWS_EthTrcv_00050] ⌈
If default 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 default error ETHTRCV_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ⌋()

[SWS_EthTrcv_00051] ⌈
If default error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the default error ETHTRCV_E_INV_TRCV_IDX otherwise (if DET is disabled) return E_NOT_OK. ⌋()

[SWS_EthTrcv_00052] ⌈
If default error detection is enabled: the function shall check the parameter TrcvModePtr for being valid. If the check fails, the function shall raise the default 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] ⌈

Service name:	EthTrcv_SetTransceiverWakeupMode	
Syntax:	`Std_ReturnType EthTrcv_SetTransceiverWakeupMode( uint8 TrcvIdx, EthTrcv_WakeupModeType TrcvWakeupMode )`	
Service ID[hex]:	0x0d	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	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
Parameters (inout):	None	
Parameters (out):	None	
Return value:	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.
Description:	Enables / disables the wake-up mode or clear the wake-up reason of the indexed transceiver	

](

[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 default 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 default error ETHTRCV_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK.]()

[SWS_EthTrcv_00123] [
If default 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 default 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 default 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] [

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

]()
[SWS_EthTrcv_00128] [
The function EthTrcv_GetTransceiverWakeupMode() shall read the current transceiver wake up mode and provide it into TrcvWakeupModePtr.]()

[SWS_EthTrcv_00129] [
If default 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 default error ETHTRCV_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK.]()

[SWS_EthTrcv_00130] [
If default 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 default error ETHTRCV_E_INV_TRCV_IDX otherwise (if DET is disabled) return E_NOT_OK.]()

[SWS_EthTrcv_00131] [
If default 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 default error ETHTRCV_E_INV_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] [

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

]()
[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 default 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 default error ETHTRCV_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK.]()

[SWS_EthTrcv_00138] [
If default 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 default 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] ⌈

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

⌋()
[SWS_EthTrcv_00056] ⌈
The function shall restart the automatic negotiation of the transmission parameters used by the indexed transceiver. ⌋()

[SWS_EthTrcv_00057] ⌈
If default 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 default error ETHTRCV_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ⌋()

[SWS_EthTrcv_00058] ⌈
If default error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the default 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_GetLinkState

[SWS_EthTrcv_00061] ⌈

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

⌋()
[SWS_EthTrcv_00062] ⌈
The function shall read the current transceiver link state. ⌋()

[SWS_EthTrcv_00063] ⌈
If default 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 default error ETHTRCV_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ⌋()

[SWS_EthTrcv_00064] ⌈
If default error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the default error ETHTRCV_E_INV_TRCV_IDX otherwise (if DET is disabled) return E_NOT_OK. ⌋()

[SWS_EthTrcv_00065] ⌈
If default error detection is enabled: the function shall check the parameter LinkStatePtr for being valid. If the check fails, the function shall raise the default 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). ⌋()

8.3.9 EthTrcv_GetBaudRate

[SWS_EthTrcv_00068] ⌈

Service name:	EthTrcv_GetBaudRate	
Syntax:	`Std_ReturnType EthTrcv_GetBaudRate( uint8 TrcvIdx, EthTrcv_BaudRateType* BaudRatePtr )`	
Service ID[hex]:	0x07	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
Parameters (inout):	None	
Parameters (out):	BaudRatePtr	ETHTRCV_BAUD_RATE_10MBIT: 10MBit connection ETHTRCV_BAUD_RATE_100MBIT: 100MBit connection ETHTRCV_BAUD_RATE_1000MBIT: 1000MBit connection
Return value:	Std_ReturnType	E_OK: success E_NOT_OK: transceiver could not be initialized
Description:	Obtains the baud rate of the indexed transceiver	

⌋()

[SWS_EthTrcv_00069] ⌈
The function shall read the current transceiver baud rate. ⌋()

[SWS_EthTrcv_00070] ⌈
If default 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 default error ETHTRCV_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ⌋()

[SWS_EthTrcv_00071] ⌈
If default error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the default error ETHTRCV_E_INV_TRCV_IDX otherwise (if DET is disabled) return E_NOT_OK. ⌋()

[SWS_EthTrcv_00072] ⌈
If default error detection is enabled: the function shall check the parameter BaudRatePtr for being valid. If the check fails, the function shall raise the default 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.]()

8.3.10 EthTrcv_GetDuplexMode

[SWS_EthTrcv_00075] [

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

]()
[SWS_EthTrcv_00076] [
The function shall read the current transceiver duplex mode.]()

[SWS_EthTrcv_00077] [
If default 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 default error ETHTRCV_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK.]()

[SWS_EthTrcv_00078] [
If default error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the default error ETHTRCV_E_INV_TRCV_IDX otherwise (if DET is disabled) return E_NOT_OK.]()

[SWS_EthTrcv_00079] [
If default error detection is enabled: the function shall check the parameter DuplexModePtr for being valid. If the check fails, the function shall raise the default 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. ⌋()

8.3.11 EthTrcv_GetVersionInfo

[SWS_EthTrcv_00082] ⌈

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

⌋()
[SWS_EthTrcv_00093] ⌈
If default error detection is enabled: the function shall check the parameter VersionInfoPtr for being valid. If the check fails, the function shall raise the default error ETHTRCV_E_PARAM_POINTER. ⌋()

8.4 Callback notifications

8.4.1 EthTrcv_ReadMiiIndication

[SWS_EthTrcv_00108] ⌈

Service name:	EthTrcv_ReadMiiIndication	
Syntax:	`void EthTrcv_ReadMiiIndication( uint8 CtrlIdx, uint8 TrcvIdx, uint8 RegIdx, uint8 RegVal )`	
Service ID[hex]:	0x09	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant for the same CtrlIdx, reentrant for different	
Parameters (in):	CtrlIdx	Index of the controller within the context of the Ethernet Driver
	TrcvIdx	Index of the transceiver on the MII
	RegIdx	Index of the transceiver register on the MII
	RegVal	Value contained in the indexed register
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Called when information has been read out via MII interface. Triggered by	

	previous Eth_ReadMii call. Can directly be called within Eth_ReadMii.

]()

8.4.2 EthTrcv_WriteMiiIndication

[SWS_EthTrcv_00109] [

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

]()

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] [

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

]()

[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.]()

[SWS_EthTrcv_00142] [
If default error detection is enabled: The function EthTrcv_MainFunction() shall check that the service EthTrcv_Init was previously called. If the check fails, the function shall raise the default error ETHTRCV_E_NOT_INITIALIZED.]()

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

]()

8.7.2 Optional Interfaces

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

[SWS_EthTrcv_00086] [

API function	*Description*
Det_ReportError	Service to report development errors.
EcuM_SetWakeupEvent	Sets the wakeup event.
Icu_DisableNotification	This function disables the notification of a channel.
Icu_EnableNotification	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] [

Service name:	<EthTrcvWakeUpCallout>	
Syntax:	`void <EthTrcvWakeUpCallout>( uint8 TrcvIdx )`	
Sync/Async:	--	
Reentrancy:	Dont care	
Parameters (in):	TrcvIdx	Index of the Ethernet Transceiver
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Indicates an wake-up request for the specified Ethernet Transceiver. Can be used to trigger integrator code that initiates a remote wake-up.	

]()

[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.

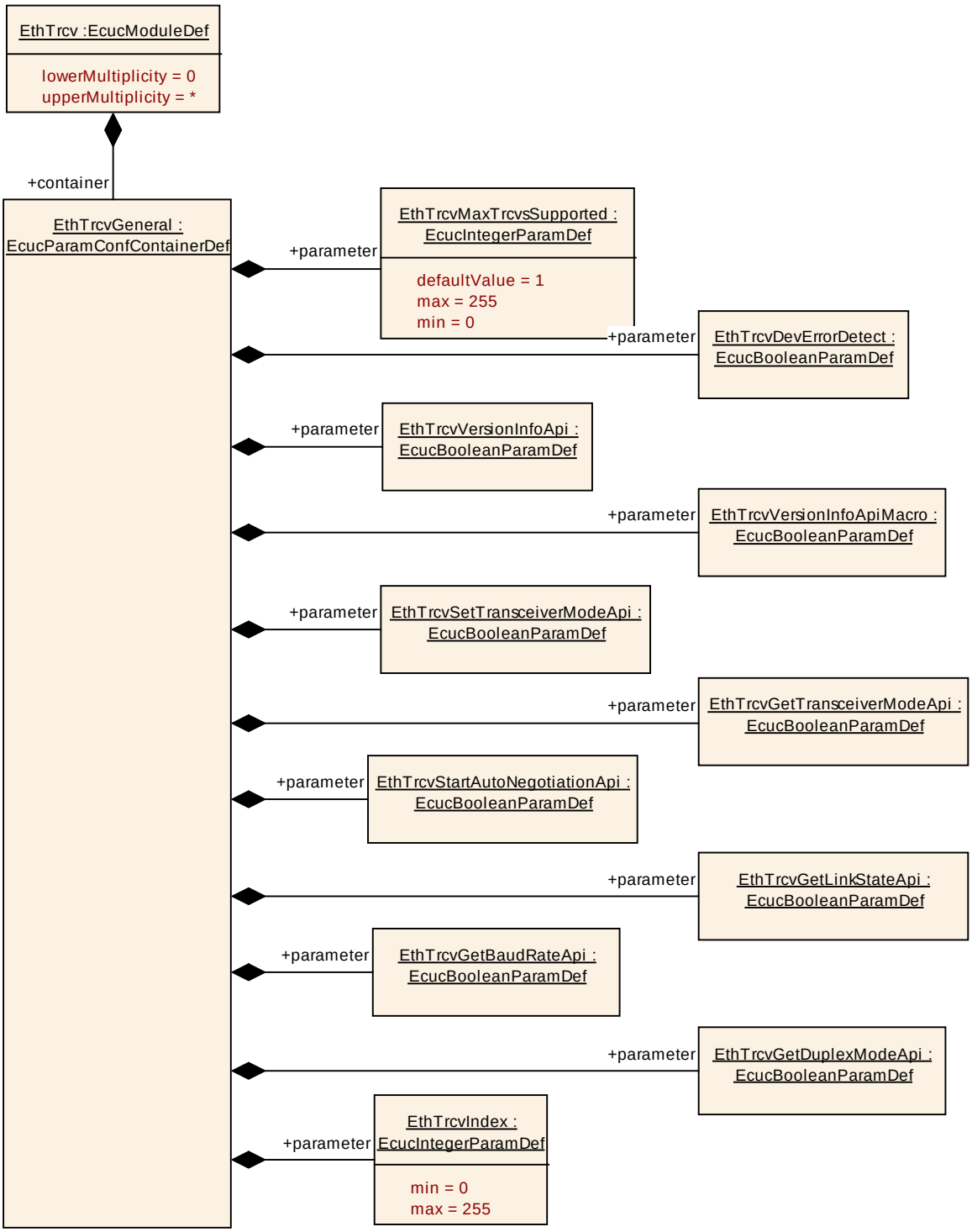

Figure 10.1: Ethernet Transceiver Driver configuration structure

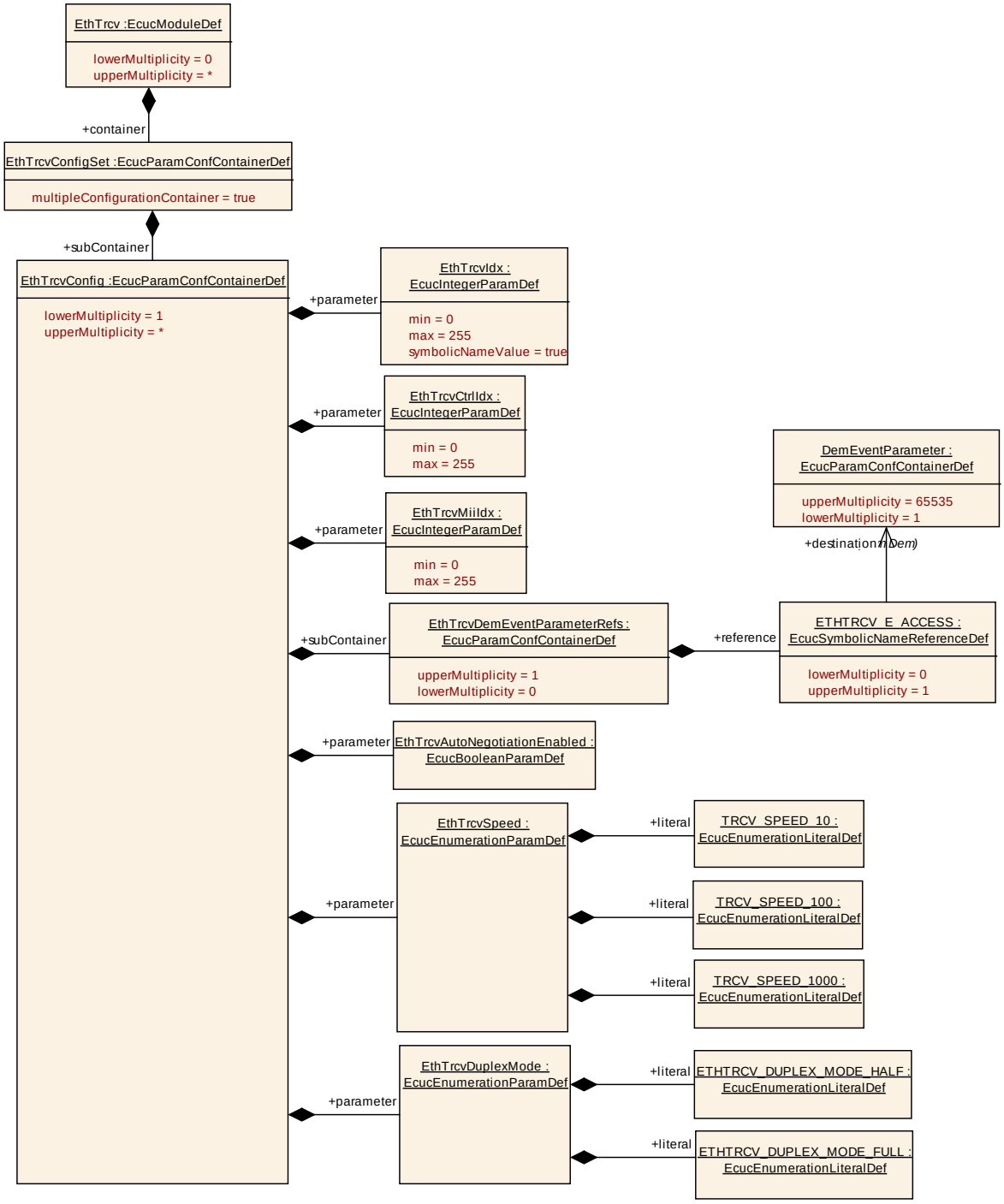

Figure 10.2: Ethernet Transceiver Driver Transceiver configuration structure

10.1.1 Variants

VARIANT-POST-BUILD: All configuration parameters in container ‘EthTrcvGeneral’ shall be configurable at pre-compile time.
Use case: Object code delivery, selectable configuration

VARIANT-LINK-TIME: All configuration parameters in container ‘EthTrcvGeneral’ shall be configurable at pre-compile time.
Use case: Object code delivery, single configuration

VARIANT-PRE-COMPILE: All configuration parameters shall be configurable at pre-compile time.
Use case: Execution time optimizations, fix configuration

10.1.2 EthTrcv

Module Name	*EthTrcv*
Module Description	Configuration of Ethernet Transceiver Driver module
Post-Build Variant Support	true

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

10.1.3 EthTrcvConfigSet

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

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

10.1.4 EthTrcvConfig

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

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

SWS Item	**ECUC_EthTrcv_00025 :**

Name	EthTrcvConnNeg		
Description	Specifies the connection negotiation of the Ethernet transceiver link.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	TRCV_CONN_NEG_AUTO	Automatic Negotiation	
	TRCV_CONN_NEG_MASTER	Master	
	TRCV_CONN_NEG_SLAVE	Slave	
Post-Build Variant Value	true		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_EthTrcv_00014 :**		
Name	EthTrcvCtrlIdx		
Description	Specifies the controller used for MII access to the transceiver		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

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

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

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

SWS Item	**ECUC_EthTrcv_00024 :**		
Name	EthTrcvPhysLayerType		
Description	Specifies the physical layer type of the Ethernet transceiver link.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	TRCV_PHYS_LAYER_TYPE_BASE_T		BaseT physcial layer (10BaseT, 1000BaseT, 1000BaseT)
	TRCV_PHYS_LAYER_TYPE_BROADR_REACH		BroadR-Reach physical layer
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_EthTrcv_00022 :**		
Name	EthTrcvSpeed		
Description	Specifies the speed of the Ethernet transceiver link in [MBit/s]. If AutoNegotiation is enabled this is the maximum speed advertised for Auto-Negotiation.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	TRCV_SPEED_10		10 MBit/s
	TRCV_SPEED_100		100 MBit/s
	TRCV_SPEED_1000		1000 MBit/s
Post-Build Variant Value	true		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: EthTrcvAutoNegotiationEnabled		

SWS Item	**ECUC_EthTrcv_00028 :**

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

SWS Item	**ECUC_EthTrcv_00026 :**		
Name	EthTrcvIcuChannelRef		
Description	Reference to the IcuChannel to enable/disable the interrupts for wakeups.		
Multiplicity	0..1		
Type	Symbolic name reference to [IcuChannel]		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

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

10.1.5 EthTrcvDemEventParameterRefs

SWS Item	**ECUC_EthTrcv_00017 :**
Container Name	EthTrcvDemEventParameterRefs
Description	Container for the references to DemEventParameter elements which shall

	be invoked using the API Dem_ReportErrorStatus API in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.
Configuration Parameters	

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

No Included Containers

10.1.6 EthTrcvWakeupMap

SWS Item	**ECUC_EthTrcv_00027 :**
Container Name	EthTrcvWakeupMap
Description	Container for the mapping of wake up reasons to wake up sources. At least one container is needed if EthTrcvWakeUpSupport is not ETHTRCV_WAKEUP_NOT_SUPPORTED.
Configuration Parameters	

SWS Item	**ECUC_EthTrcv_00033 :**	
Name	EthTrcvWakeupReason	
Description	This parameter defines the transceiver wake up reasons.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	ETHTRCV_WUR_BUS	0x02: Bus wake up detected. Available if supported by hardware.
	ETHTRCV_WUR_GENERAL	0x01: General wake up detected, no distinct reason supported by hardware.
	ETHTRCV_WUR_INTERNAL	0x03: Internal wake up detected. Available if supported by hardware.
	ETHTRCV_WUR_PIN	0x06: Pin wake up detected. Available if supported by hardware.
	ETHTRCV_WUR_POWER_ON	0x05: Power on wake up detected. Available if supported by hardware.
	ETHTRCV_WUR_RESET	0x04: Reset wake up detected. Available if supported by hardware.
	ETHTRCV_WUR_SYSERR	0x07: System error wake up detected. Available if supported by hardware.

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

SWS Item	**ECUC_EthTrcv_00029 :**		
Name	EthTrcvWakeupSourceRef		
Description	Configures the wake-up source defined in EcuM.		
Multiplicity	1		
Type	Symbolic name reference to [EcuMWakeupSource]		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.1.7 EthTrcvGeneral

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

SWS Item	**ECUC_EthTrcv_00003 :**		
Name	EthTrcvDevErrorDetect		
Description	Switches the Default Error Tracer (Det) detection and notification ON or OFF. • true: enabled (ON). • false: disabled (OFF).		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_EthTrcv_00010 :**		
Name	EthTrcvGetBaudRateApi		
Description	Enables / Disables EthTrcv_GetBaudRate API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	

	Post-build time	--	
Scope / Dependency	scope: local		

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

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

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

SWS Item	**ECUC_EthTrcv_00031 :**		
Name	EthTrcvGetTransceiverWakeupModeApi		
Description	Enables / Disables EthTrcv_GetTransceiverWakeupMode API		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local dependency: Only valid if EthTrcvWakeUpSupport is not		

	ETHTRCV_WAKEUP_NOT_SUPPORTED

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

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

SWS Item	**ECUC_EthTrcv_00002 :**		
Name	EthTrcvMaxTrcvsSupported		
Description	--		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	1		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_EthTrcv_00006 :**		
Name	EthTrcvSetTransceiverModeApi		
Description	Enables / Disables EthTrcv_SetTransceiverMode API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	

	Post-build time	--	
Scope / Dependency	scope: local		

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

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

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

SWS Item	**ECUC_EthTrcv_00030 :**		
Name	EthTrcvWakeUpSupport		
Description	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.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	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
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

.

11 Not applicable requirements

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