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

Document Version	1.2.0
Document Status	Final
Part of Release	4.0
Revision	3

Document Change History

Date	Version	Changed by	Change Description
09.12.2011	1.2.0	AUTOSAR Administration	• EthTrcv_GetVersionInfo revised
26.10.2010	1.1.0	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
30.11.2009	1.0.0	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.

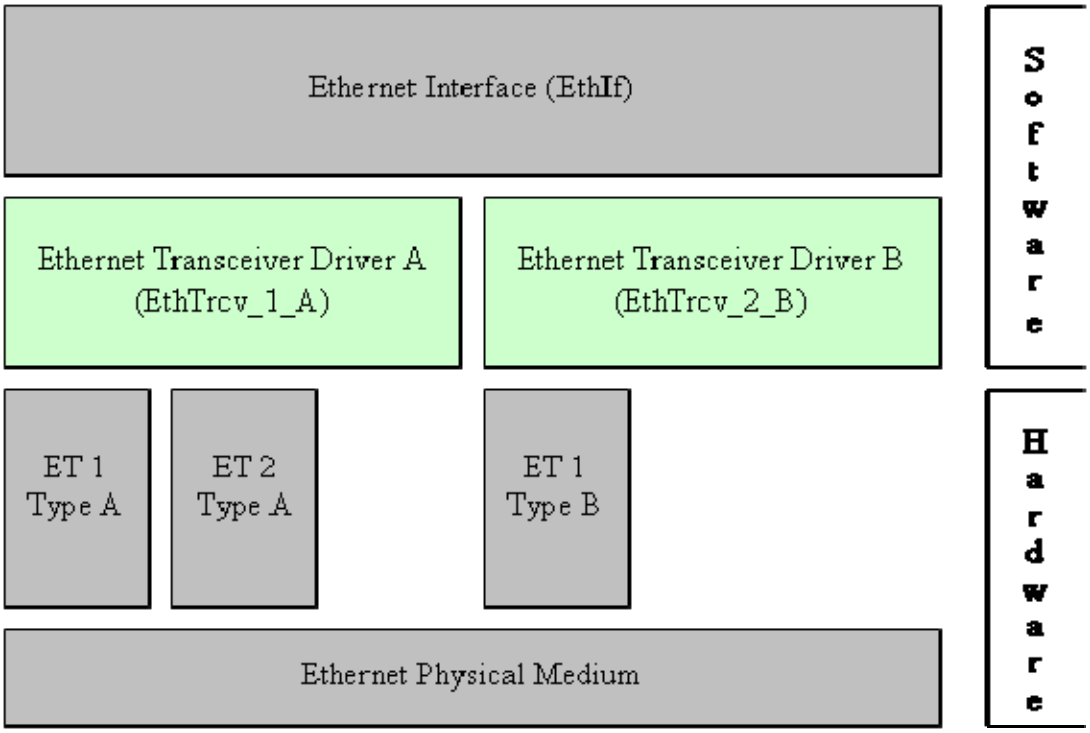

Figure 1.1: Ethernet stack module overview

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 Development Error Tracer
AUTOSAR_SWS_DevelopmentErrorTracer.pdf

[17] Specification of Diagnostics Event Manager
AUTOSAR_SWS_DiagnosticEventManager

[18] Specification of C Implementation Rules
AUTOSAR_TR_CImplementationRules.pdf

[19] Specification of ECU State Manager
AUTOSAR_SWS_ECUStateManager.pdf

3.2 Related standards and norms

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

[21] IEEE 802.3-2006

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:

- Development Error Tracer (DET) for reporting of development errors.
- Diagnostic Event Manager (DEM) for reporting of diagnostic-relevant events and states.
- BSW Scheduler mechanisms for data consistency and main function handling.
- 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 Code file structure

[ETHTRCV001] ⌈

This specification shall not completely define the code file structure. The code-file structure shall include the following files named:

- EthTrcv_Lcfg.c – for link time configurable parameters and
- EthTrcv_PBcfg.c – for post build time configurable parameters.

These files shall contain all link time and post-build time configurable parameters.⌋()

5.1.2 Header file structure

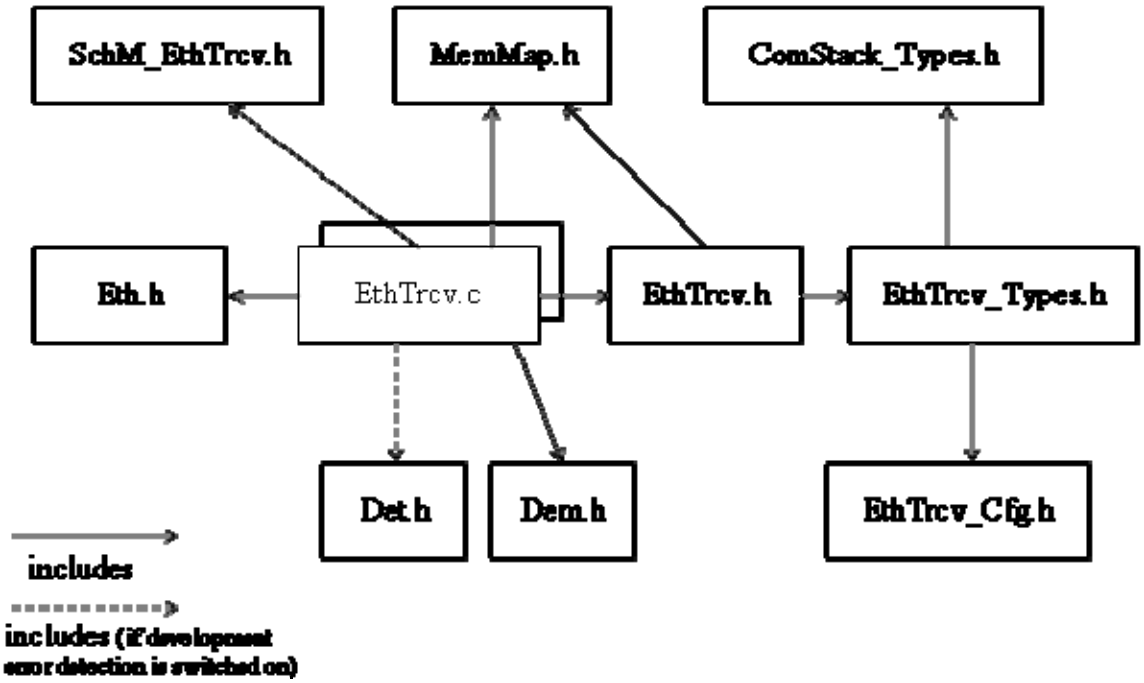

Figure 5.1: Ethernet Transceiver Driver file structure

[ETHTRCV002]「
The module shall include the Dem.h file. File Dem.h defines the APIs to report errors as well as the required Event Id symbols. This specification defines the name of the Event Id symbols provided by XML to the DEM configuration tool. The DEM configuration tool assigns ECU dependent values to the Event Id symbols and publishes the symbols.」()

6 Requirements traceability

Requirement	Description	Satisfied by
BSW00170	These requirements are not applicable to this specification.	ETHTRCV999

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.

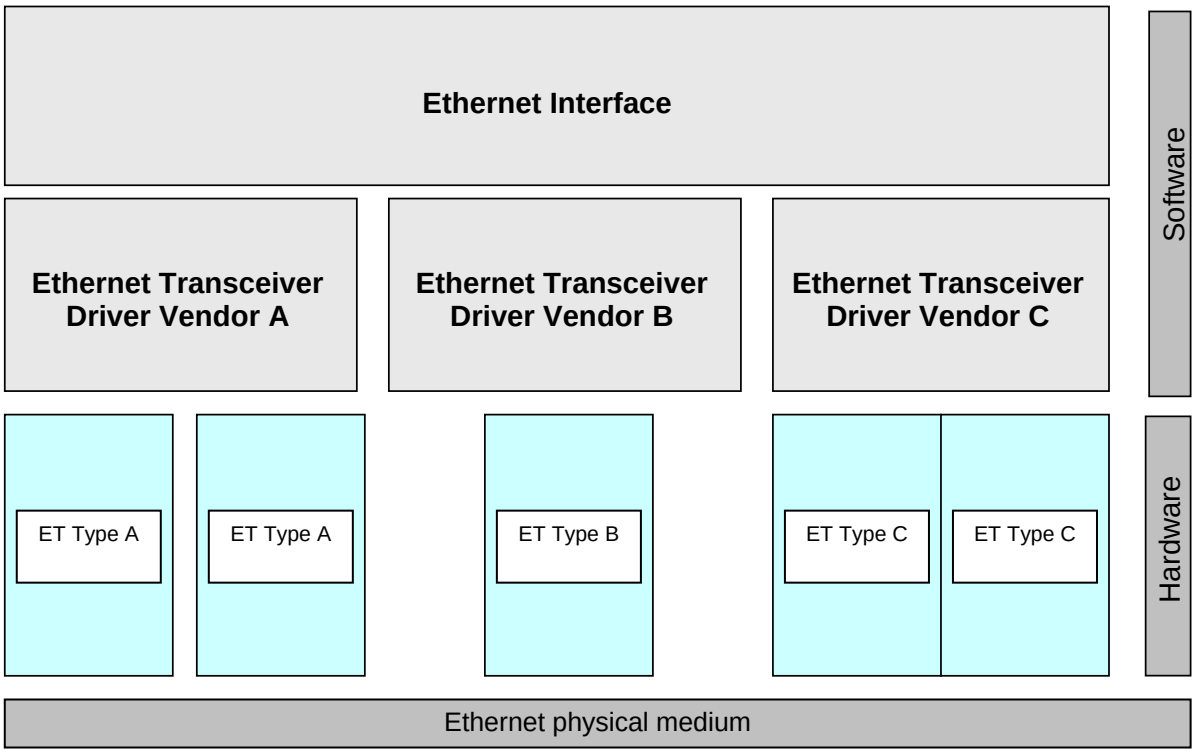

Figure 7.1: Basic Structure of the Ethernet BSW stack

7.1.1 Indexing scheme

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

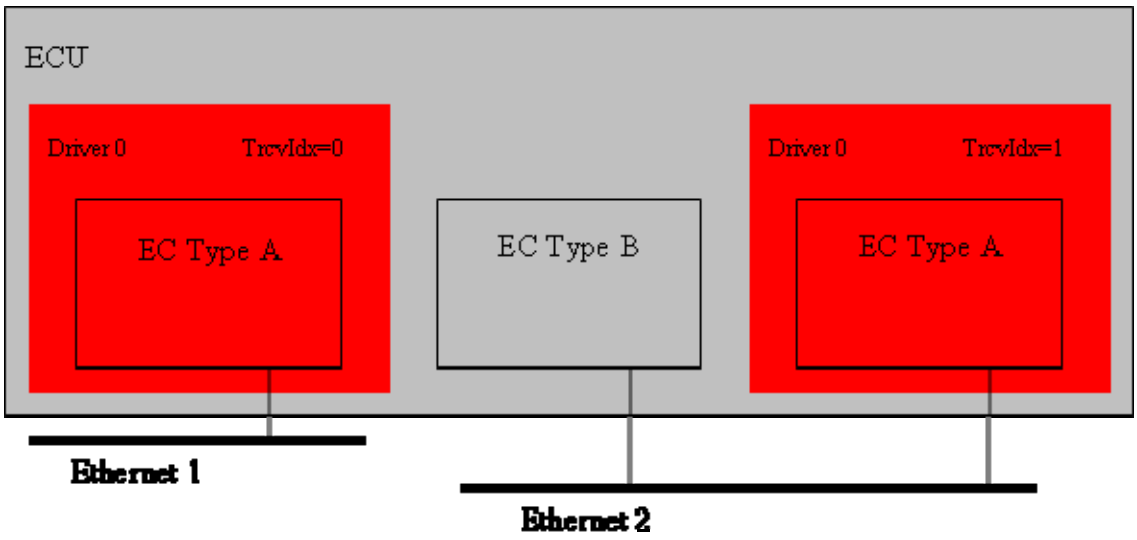

Figure 7.2: Ethernet Transceiver Driver indexing scheme

[ETHTRCV003]⌜
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.

[ETHTRCV004] ⌜
The Ethernet Transceiver Driver module shall support pre-compile time, link time and post-build time configuration.⌟()

[ETHTRCV005] ⌜
The header file *EthTrcv.h* shall include a software and specification version number.⌟
()

[ETHTRCV006] ⌜
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.⌟()

[ETHTRCV007] ⌈
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].

[ETHTRCV008] ⌈
The Ethernet Transceiver Driver module implementation shall conform to the HIS subset of the MISRA C Standard (see document [18]).⌋()

[ETHTRCV009] ⌈
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.⌋()

[ETHTRCV010] ⌈
None of the Ethernet Transceiver Driver module header files shall define global variables.⌋()

7.1.3 Configuration description

[ETHTRCV011] ⌈
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.⌋()

[ETHTRCV012] ⌈
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. ⌋()

[ETHTRCV013] ⌈
The MCG shall ensure the consistency of the generated configuration data.⌋()

[ETHTRCV014] ⌈

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. ⌋()

[ETHTRCV015] ⌈
The start address of post-build time configuration data shall be passed during module initialization (see chapter 8.3.1). ⌋()

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

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

7.2 Error classification

[ETHTRCV016] ⌈
The configuration of the Dem assigns values for production code Event Ids. The file Dem.h includes the file Dem_IntErrId.h. The file Dem_IntErrId.h publishes the values. ⌋()

[ETHTRCV017] ⌈

Development error values are of type uint8.⌋()

Type or error	***Relevance***	***Related error code***	***Value [hex]***
Invalid transceiver index	Development	ETHTRCV_E_INV_TRCV_IDX	0x01
EthTrcv module was not initialized	Development	ETHTRCV_E_NOT_INITIALIZED	0x02
Invalid pointer in parameter list	Development	ETHTRCV_E_INV_POINTER	0x03
Invalid configuration	Development	ETHTRCV_E_INV_CONFIG	0x04
Transceiver access failed	Production	ETHTRCV_E_ACCESS	Assigned by DEM

7.3 Error detection

[ETHTRCV018] ⌈
The detection of development errors is configurable (*ON* / *OFF*) at pre-compile time. The switch *EthTrcvDevErrorDetect* (see chapter 10) shall activate or deactivate the detection of all development errors.⌋()

[ETHTRCV019] ⌈
The *EthTrcvDevErrorDetect* switch enables API parameter checking. Chapter 7.2 and 8 contain the detailed description of the detected errors.⌋()

[ETHTRCV020] ⌈
Switching off the detection of production code errors shall not be possible.⌋()

7.4 Error notification

[ETHTRCV021] ⌈
The module shall report development errors to the *Det_ReportError* service of the Development Error Tracer (DET) if the pre-processor switch *EthTrcvDevErrorDetect* is set (see chapter 10).⌋()

[ETHTRCV022] ⌈
The module shall report production errors to the Diagnostic Event Manager.⌋()

7.5 Debugging

[ETHTRCV023] ⌈
Each variable that shall be accessible by AUTOSAR Debugging, shall be defined as global variable.⌋()

[ETHTRCV024] ⌈
All type definitions of variables, which shall be debugged, shall be accessible by the header file EthTrcv.h.⌋()

[ETHTRCV025] ⌈
The declaration of variables in the header file shall be such, that it is possible to calculate the size of the variables by C-“sizeof”.⌋()

[ETHTRCV026] ⌈
Variables available for debugging shall be described in the respective Basic Software Module Description.⌋()

7.6 Version checking

[ETHTRCV091] ⌈

The Ethernet Transceiver Driver module shall perform inter-module checks to avoid integration of incompatible files.

The imported include files shall be checked by preprocessing directives.⌋()

The Ethernet Transceiver Driver module shall verify the following version numbers:
- <MODULENAME>_AR_RELEASE_MAJOR_VERSION
- <MODULENAME>_AR_RELEASE_MINOR_VERSION

Where <MODULENAME> is the module abbreviation of the other (external) modules providing header files included by the Ethernet Transceiver Driver module.

If the values are not identical to the expected values, the Ethernet Transceiver Driver module shall report an error.

8 API specification

8.1 Imported types

This chapter lists all types included from the following files:

[ETHTRCV027]「

Module	*Imported Type*
ComStack_Types	BufReq_ReturnType
Dem	Dem_EventIdType
	Dem_EventStatusType
Eth	Eth_DataType
	Eth_FrameType
	Eth_ModeType
	Eth_ConfigType
Std_Types	Std_ReturnType
	Std_VersionInfoType

」()

8.2 Type definitions

8.2.1 EthTrcv_ConfigType

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

8.2.2 EthTrcv_ModeType

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

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

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

8.2.5 EthTrcv_BaudRateType

Name:	`EthTrcv_BaudRateType`	
Type:	`Enumeration`	
Range:	`ETHTRCV_BAUD_RATE_10MBIT`	0x00: 10MBIT Ethnernet connection
	`ETHTRCV_BAUD_RATE_100MBUT`	0x01: 100MBit 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

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

8.3 Function definitions

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

8.3.1 EthTrcv_Init

[ETHTRCV028]⌜

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

」()

[ETHTRCV029]「
The function shall store the access to the configuration structure for subsequent API calls.」()

[ETHTRCV030]「
The function shall change the state of the component from ETHTRCV_STATE_UNINIT to ETHTRCV_STATE_INIT.」()

[ETHTRCV031]「
If development error detection is enabled: the function shall check the parameter CfgPtr for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_POINTER.」()

[ETHTRCV032]「
Caveat: The API has to be called during initialization.」()

[ETHTRCV033]「
Configuration: The user shall pass the post-build configuration or a NULL_PTR as parameter depending on the configuration variant.」()

8.3.2 EthTrcv_TransceiverInit

[ETHTRCV034]「

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

」()

[ETHTRCV035] ⌈
The function shall:

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

[ETHTRCV036] ⌈
The function shall change the state of the component from ETHTRCV_STATE_INIT to ETHTRCV_STATE_ACTIVE.⌋()

[ETHTRCV037] ⌈
If development error detection is enabled: the function shall check that the service EthTrcv_Init was previously called. If the check fails, the function shall raise the development error ETHTRCV_E_NOT_INITIALIZED and return E_NOT_OK.⌋()

[ETHTRCV038] ⌈
If development error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_TRCV_IDX and return E_NOT_OK.⌋()

[ETHTRCV039] ⌈
If development error detection is enabled: the function shall check the parameter CfgIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_CONFIG and return E_NOT_OK.⌋()

[ETHTRCV040] ⌈
The function shall check the access to the Ethernet controller. If the check fails, the function shall raise the production error ETHTRCV_E_ACCESS and return E_NOT_OK.⌋()

[ETHTRCV041] ⌈
Caveat: The function requires previous initialization (EthTrcv_Init).⌋()

8.3.3 EthTrcv_SetTransceiverMode

[ETHTRCV042] ⌈

Service name:	EthTrcv_SetTransceiverMode
Syntax:	`Std_ReturnType EthTrcv_SetTransceiverMode( uint8 TrcvIdx, EthTrcv_ModeType CtrlMode )`
Service ID[hex]:	0x03
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant

Parameters (in):	TrcvIdx	Index of the transceiver within the context of the Ethernet Transceiver Driver
	CtrlMode	ETHTRCV_MODE_DOWN: disable the transceiver ETHTRCV_MODE_ACTIVE: enable the transceiver
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: success E_NOT_OK: transceiver could not be initialized
Description:	Enables / disables the indexed transceiver	

」()

[ETHTRCV043]「

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

[ETHTRCV044]「

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

[ETHTRCV045]「

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

[ETHTRCV046]「

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

[ETHTRCV094]「

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

[ETHTRCV047]「

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

8.3.4 EthTrcv_GetTransceiverMode

[ETHTRCV048]「

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	

」()

[ETHTRCV049]「
The function shall read the current transceiver mode.」()

[ETHTRCV050]「
If development error detection is enabled: the function shall check that the service EthTrcv_TransceiverInit was previously called. If the check fails, the function shall raise the development error ETHTRCV_E_NOT_INITIALIZED and return E_NOT_OK.」()

[ETHTRCV051]「
If development error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_TRCV_IDX and return E_NOT_OK.」()

[ETHTRCV052]「
If development error detection is enabled: the function shall check the parameter TrcvModePtr for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_POINTER and return E_NOT_OK.」()

[ETHTRCV053]「
The function shall be pre compile time configurable On/Off by the configuration parameter: EthTrcvGetTransceiverModeApi.」()

[ETHTRCV054]「
Caveat: The function requires previous transceiver initialization (EthTrcv_TransceiverInit).」()

8.3.5 EthTrcv_StartAutoNegotiation

[ETHTRCV055]「

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	

」()

[ETHTRCV056] 「
The function shall restart the automatic negotiation of the transmission parameters used by the indexed transceiver.」()

[ETHTRCV057] 「
If development error detection is enabled: the function shall check that the service EthTrcv_TransceiverInit was previously called. If the check fails, the function shall raise the development error ETHTRCV_E_NOT_INITIALIZED and return E_NOT_OK.」()

[ETHTRCV058] 「
If development error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_TRCV_IDX and return E_NOT_OK.」()

[ETHTRCV059] 「
The function shall be pre compile time configurable On/Off by the configuration parameter: EthTrcvStartAutoNegotiationApi.」()

[ETHTRCV060] 「
Caveat: The function requires previous transceiver initialization (EthTrcv_TransceiverInit).」()

[ETHTRCV088] 「
Caveat: The function is not required or called by an upper layer BSW software component.」()

8.3.6 EthTrcv_GetLinkState

[ETHTRCV061] ⌈

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

⌋()

[ETHTRCV062] ⌈

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

[ETHTRCV063] ⌈

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

[ETHTRCV064] ⌈

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

[ETHTRCV065] ⌈

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

[ETHTRCV066] ⌈

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

[ETHTRCV067] ⌈
Caveat: The function requires previous transceiver initialization (EthTrcv_TransceiverInit).⌋()

8.3.7 EthTrcv_GetBaudRate

[ETHTRCV068] ⌈

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
Return value:	Std_ReturnType	E_OK: success E_NOT_OK: transceiver could not be initialized
Description:	Obtains the baud rate of the indexed transceiver	

⌋()

[ETHTRCV069] ⌈

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

[ETHTRCV070] ⌈
If development error detection is enabled: the function shall check that the service EthTrcv_TransceiverInit was previously called. If the check fails, the function shall raise the development error ETHTRCV_E_NOT_INITIALIZED and return E_NOT_OK.⌋()

[ETHTRCV071] ⌈
If development error detection is enabled: the function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_TRCV_IDX and return E_NOT_OK.⌋()

[ETHTRCV072] ⌈
If development error detection is enabled: the function shall check the parameter BaudRatePtr for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_POINTER and return E_NOT_OK.⌋()

[ETHTRCV073] ⌈

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

[ETHTRCV074]「
Caveat: The function requires previous transceiver initialization (EthTrcv_TransceiverInit).」()

[ETHTRCV089]「
Caveat: The function is not required or called by an upper layer BSW software component.」()

8.3.8 EthTrcv_GetDuplexMode

[ETHTRCV075]「

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	

」()

[ETHTRCV076]「

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

[ETHTRCV077]「
If development error detection is enabled: the function shall check that the service EthTrcv_TransceiverInit was previously called. If the check fails, the function shall raise the development error ETHTRCV_E_NOT_INITIALIZED and return E_NOT_OK.」()

[ETHTRCV078]「

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

[ETHTRCV079]「
If development error detection is enabled: the function shall check the parameter DuplexModePtr for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_POINTER and return E_NOT_OK.」()

[ETHTRCV080]「
The function shall be pre compile time configurable On/Off by the configuration parameter: EthTrcvGetDuplexModeApi.」()

[ETHTRCV081]「
Caveat: The function requires previous transceiver initialization (EthTrcv_TransceiverInit).」()

[ETHTRCV090]「
Caveat: The function is not required or called by an upper layer BSW software component.」()

8.3.9 EthTrcv_GetVersionInfo

[ETHTRCV082]「

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	

」()

[ETHTRCV083]「
The function EthTrcv_ GetVersionInfo shall return the version information of this module. The version information includes:
- Two bytes for the vendor ID
- Two bytes for the module ID
- Three bytes version number. The numbering shall be vendor specific; it

consists of:
- The major, the minor and the patch version number of the module.
- The AUTOSAR specification version number shall not be included. The AUTOSAR specification version number is checked during compile time and therefore not required in this API.」()

[ETHTRCV084]「
The function EthTrcv_ GetVersionInfo shall be pre compile time configurable On/Off by the configuration parameter: EthTrcvVersionInfoApi using the keyword ETHTRCV_GET_VERSION_INFO.」()

[ETHTRCV093]「
If development error detection is enabled: the function shall check the parameter VersionInfoPtr for being valid. If the check fails, the function shall raise the development error ETHTRCV_E_INV_POINTER.」()

8.4 Callback notifications

The Ethernet Transceiver Driver does not provide any callback functions.

8.5 Interrupt service routines

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

8.6 Scheduled functions

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

8.7 Expected Interfaces

This chapter lists all interfaces required from other modules.

8.7.1 Mandatory Interfaces

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

[ETHTRCV085]「

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.
Eth_ControllerInit	Initializes the indexed controller

Eth_GetControllerMode	Obtains the state of the indexed controller
Eth_GetCounterState	Reads the value of a counter specified with its memory offset
Eth_GetPhysAddr	Obtains the physical source address used by the indexed controller
Eth_GetVersionInfo	Returns the version information of this module
Eth_Init	Initializes the Ethernet Driver
Eth_ProvideTxBuffer	Provides access to a transmit buffer of the specified controller
Eth_ReadMii	Reads a transceiver register
Eth_Receive	Triggers frame reception
Eth_SetControllerMode	Enables / disables the indexed controller
Eth_Transmit	Triggers transmission of a previously filled transmit buffer
Eth_TxConfirmation	Triggers frame transmission confirmation
Eth_WriteMii	Configures a transceiver register or triggers a function offered by the receiver
SchM_Enter_EthTrcv	Invokes the SchM_Enter function to enter a module local exclusive area.
SchM_Exit_EthTrcv	Invokes the SchM_Exit function to exit an exclusive area.

」()

8.7.2 Optional Interfaces

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

[ETHTRCV086] 「

API function	***Description***
Det_ReportError	Service to report development errors.

」()

8.7.3 Configurable interfaces

The Ethernet Transceiver Driver does not use configurable interfaces.

Terms and definitions:
Reentrant: interface is expected to be reentrant
Don't care: reentrancy of interface not relevant for this module (in general it is in this case not reentrant).

9 Sequence diagrams

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

10 Configuration specification

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

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

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

10.1 How to read this chapter

In addition to this section, it is highly recommended to read the documents:

- AUTOSAR Layered Software Architecture [2].
- AUTOSAR ECU Configuration Specification [13]. This document describes the AUTOSAR configuration methodology and the AUTOSAR configuration metamodel in detail.

The following is only a short survey of the topic and it will not replace the ECU Configuration Specification document.

10.1.1 Configuration and configuration parameters

Configuration parameters define the variability of the generic part(s) of an implementation of a module. This means that only generic or configurable module implementation can be adapted to the environment (software/hardware) in use during system and/or ECU configuration.

The configuration of parameters can be achieved at different times during the software process: before compile time, before link time or after build time. In the following, the term “configuration class” (of a parameter) shall be used in order to refer to a specific configuration point in time.

10.1.2 Variants

Variants describe sets of configuration parameters. E.g., variant 1: only pre-compile time configuration parameters; variant 2: mix of pre-compile and post-build time configuration parameters. In one variant, a parameter can only be of one configuration class.

10.1.3 Containers

Containers structure the set of configuration parameters. This means:

- *all* configuration parameters are kept in containers.
- (sub-) containers can reference (sub-) containers. It is possible to assign a multiplicity to these references. The multiplicity then defines the possible number of instances of the contained parameters.

10.1.4 Specification template for configuration parameters

The following tables consist of three sections:
- the general section
- the configuration parameter section
- the section of included/referenced containers

Pre-compile time - specifies whether the configuration parameter shall be of configuration class *Pre-compile time* or not

Label	***Description***
x	The configuration parameter shall be of configuration class *Pre-compile time*.
--	The configuration parameter shall never be of configuration class *Pre-compile time*.

Link time - specifies whether the configuration parameter shall be of configuration class *Link time* or not

Label	***Description***
x	The configuration parameter shall be of configuration class *Link time*.
--	The configuration parameter shall never be of configuration class *Link time*.

Post Build - specifies whether the configuration parameter shall be of configuration class *Post Build* or not

Label	***Description***
x	The configuration parameter shall be of configuration class *Post Build* and no specific implementation is required.
L	*Loadable* - the configuration parameter shall be of configuration class *Post Build* and only one configuration parameter set resides in the ECU.
M	*Multiple* - the configuration parameter shall be of configuration class *Post Build* and is selected out of a set of multiple parameters by passing a dedicated pointer to the init function of the module.
--	The configuration parameter shall never be of configuration class *Post Build*.

10.2 Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters describe Chapters 7 and Chapter 7.5.

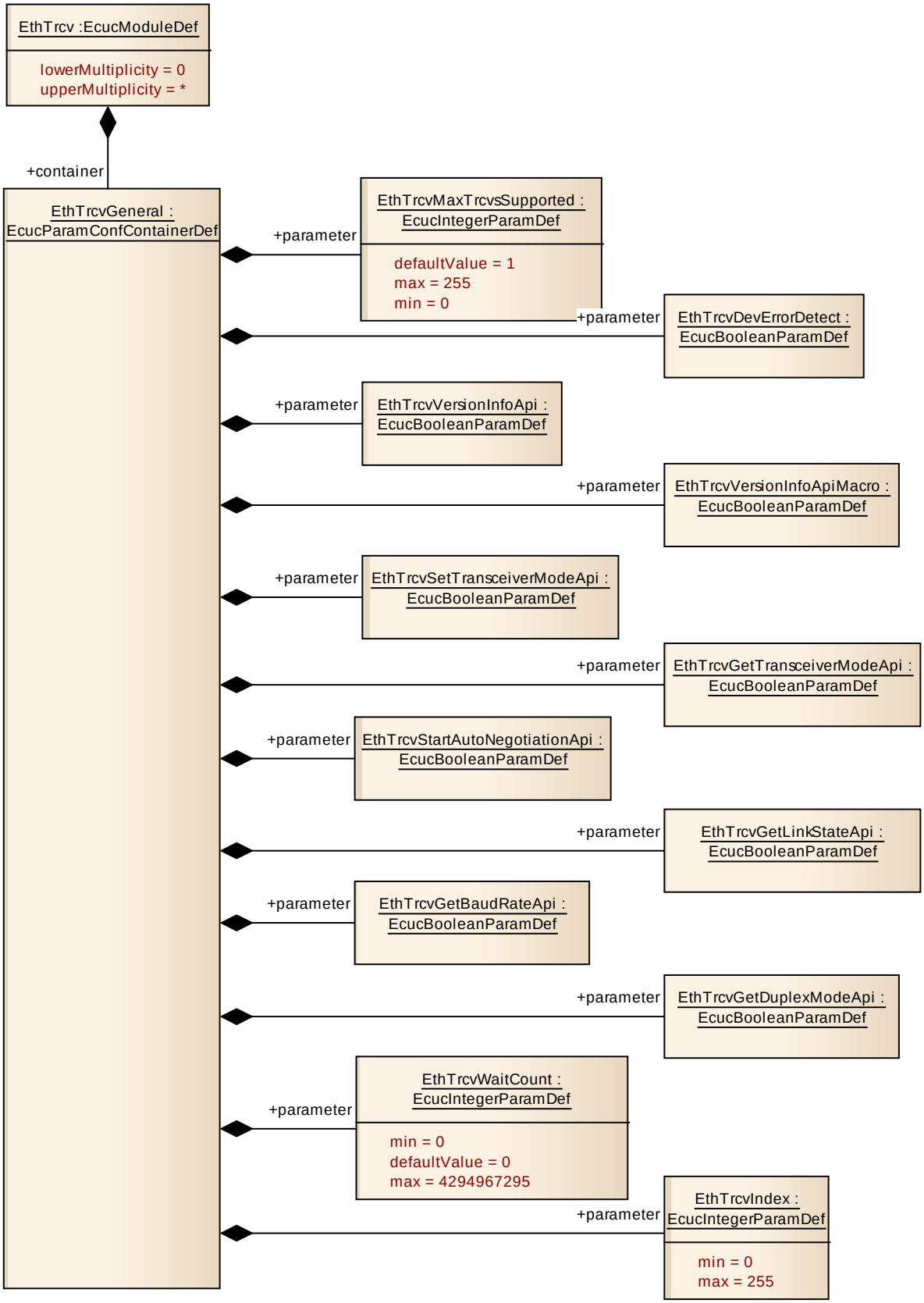

Figure 10.1: Ethernet Transceiver Driver configuration structure

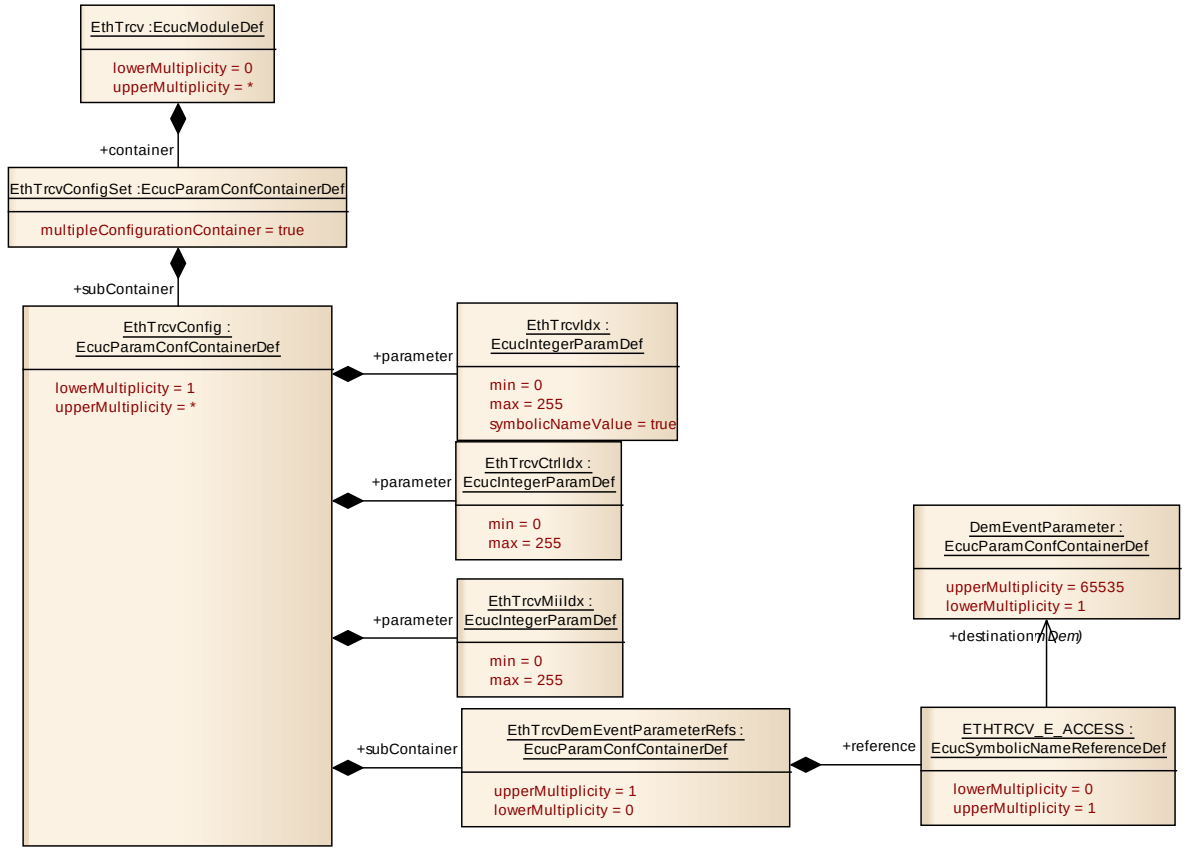

Figure 10.2: Ethernet Transceiver Driver Transceiver configuration structure

10.2.1 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

10.2.2 EthTrcv

Module Name	*EthTrcv*
Module Description	Configuration of Ethernet Transceiver Driver module

Included Containers		
Container Name	***Multiplicity***	***Scope / Dependency***
EthTrcvConfigSet	1	All underlying parameters may be part of a multiple

		configuration set.
EthTrcvGeneral	1	General configuration of Ethernet Transceiver Driver module

10.2.3 EthTrcvConfigSet

SWS Item	**ETHTRCV016_Conf :**
Container Name	EthTrcvConfigSet [Multi Config Container]
Description	All underlying parameters may be part of a multiple configuration set.
Configuration Parameters	

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

10.2.4 EthTrcvConfig

SWS Item	**ETHTRCV012_Conf :**
Container Name	EthTrcvConfig
Description	Configuration of the individual transceiver
Configuration Parameters	

SWS Item	**ETHTRCV014_Conf :**		
Name	EthTrcvCtrlIdx		
Description	Specifies the controller used for MII access to the transceiver		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: Module		

SWS Item	**ETHTRCV013_Conf :**		
Name	EthTrcvIdx		
Description	Specifies the instance ID of the configured transceiver.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 255		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: Module		

SWS Item	**ETHTRCV015_Conf :**		
Name	EthTrcvMiiIdx		
Description	Specifies the transceiver index used for MII access to the transceiver		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME

	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: Module		

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

10.2.5 EthTrcvDemEventParameterRefs

SWS Item	**ETHTRCV017_Conf :**
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	**ETHTRCV018_Conf :**		
Name	ETHTRCV_E_ACCESS		
Description	Reference to the DemEventParameter which shall be issued when the error "Transceiver access failed" has occured.		
Multiplicity	0..1		
Type	Reference to [DemEventParameter]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: Module		

No Included Containers

10.2.6 EthTrcvGeneral

SWS Item	**ETHTRCV001_Conf :**
Container Name	EthTrcvGeneral
Description	General configuration of Ethernet Transceiver Driver module
Configuration Parameters	

SWS Item	**ETHTRCV003_Conf :**		
Name	EthTrcvDevErrorDetect		
Description	Enables / Disables development error detection		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	

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

SWS Item	**ETHTRCV010_Conf :**		
Name	EthTrcvGetBaudRateApi		
Description	Enables / Disables EthTrcv_GetBaudRate API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: Module		

SWS Item	**ETHTRCV011_Conf :**		
Name	EthTrcvGetDuplexModeApi		
Description	Enables / Disables EthTrcv_GetDuplexMode API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: Module		

SWS Item	**ETHTRCV009_Conf :**		
Name	EthTrcvGetLinkStateApi		
Description	Enables / Disables EthTrcv_GetLinkState API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: Module		

SWS Item	**ETHTRCV007_Conf :**		
Name	EthTrcvGetTransceiverModeApi		
Description	Enables / Disables EthTrcv_GetTransceiverMode API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: Module		

SWS Item	**ETHTRCV020_Conf :**	
Name	EthTrcvIndex	
Description	Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	--	

ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: Module		

SWS Item	**ETHTRCV002_Conf :**		
Name	EthTrcvMaxTrcvsSupported		
Description	--		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	1		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: Module		

SWS Item	**ETHTRCV006_Conf :**		
Name	EthTrcvSetTransceiverModeApi		
Description	Enables / Disables EthTrcv_SetTransceiverMode API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: Module		

SWS Item	**ETHTRCV008_Conf :**		
Name	EthTrcvStartAutoNegotiationApi		
Description	Enables / Disables EthTrcv_StartAutoNegotiation API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: Module		

SWS Item	**ETHTRCV004_Conf :**		
Name	EthTrcvVersionInfoApi		
Description	Enables / Disables version info API		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: Module		

SWS Item	**ETHTRCV005_Conf :**
Name	EthTrcvVersionInfoApiMacro
Description	Enables / Disables version info API macro implementation
Multiplicity	1
Type	EcucBooleanParamDef

Default value	--		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: Module		

SWS Item	**ETHTRCV019_Conf :**		
Name	EthTrcvWaitCount		
Description	Wait count for transceiver state changes.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	0		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: Module		

No Included Containers

10.3 Published Information

[ETHTRCV087]「The standardized common published parameters as required by BSW00402 in the SRS General on Basic Software Modules [3] shall be published within the header file of this module and need to be provided in the BSW Module Description. The according module abbreviation can be found in the List of Basic Software Modules [6].」()

Additional module-specific published parameters are listed below if applicable.

11 Not applicable requirements

[ETHTRCV999]「 These requirements are not applicable to this specification. 」(BSW00170)