

<b>Document Title</b>	Specification of Wireless Ethernet Transceiver Driver
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	799

<b>Document Status</b>	published
<b>Part of AUTOSAR Standard</b>	Classic Platform
<b>Part of Standard Release</b>	R23-11

<b>Document Change History</b>			
<b>Date</b>	<b>Release</b>	<b>Changed by</b>	<b>Description</b>
2023-11-23	R23-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• WEthTrcvId, WEthTrcvRadioid and WEthTrcvAntennaid no anymore manually configured</li> </ul>
2022-11-24	R22-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Figures are now at the begin of a chapter.</li> <li>• Fixed text alignment of some artifacts.</li> <li>• Fixed some incorrect trace references.</li> </ul>
2021-11-25	R21-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Header File Cleanup</li> <li>• Fixed configuration ranges</li> <li>• Editorial changes</li> </ul>
2020-11-30	R20-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Editorial changes</li> </ul>
2019-11-28	R19-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Partition handling released</li> <li>• Changed Document Status from Final to published</li> </ul>
2018-10-31	4.4.0	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Basic Software Multicore Distribution (DRAFT).</li> </ul>
2017-12-08	4.3.1	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• editorial changes</li> </ul>



△

2016-11-30	4.3.0	AUTOSAR Release Management	• Initial Release.
------------	-------	----------------------------------	--------------------

## Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

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

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

## Contents

1	Introduction and functional overview	6
2	Acronyms and Abbreviations	7
3	Related documentation	8
3.1	Input documents & related standards and norms	8
3.2	Related specification	8
4	Constraints and assumptions	9
4.1	Limitations	9
4.2	Applicability to car domains	9
5	Dependencies to other modules	10
6	Requirements Tracing	11
7	Functional specification	12
7.1	Wireless Ethernet BSW stack	12
7.1.1	Indexing scheme	12
7.1.2	Requirements	12
7.1.3	Transceiver Parameters	13
7.1.4	Key/Value Parameter Mapping	13
7.1.5	MainFunction	15
7.1.6	V2X Specific Transceiver Requirements	15
7.1.7	Wake-up support	15
7.2	Error Classification	16
7.2.1	Development Errors	16
7.2.2	Runtime Errors	16
7.2.3	Transient Faults	16
7.2.4	Production Errors	16
7.2.5	Extended Production Errors	16
7.3	Security Events	17
8	API specification	18
8.1	Imported types	18
8.2	Type definitions	18
8.2.1	WEthTrcv_ConfigType	19
8.2.2	WEthTrcv_SetRadioParamIdType	19
8.2.3	WEthTrcv_SetChanRxParamIdType	19
8.2.4	WEthTrcv_SetChanTxParamIdType	20
8.2.5	WEthTrcv_GetChanRxParamIdType	21
8.2.6	WEthTrcv_BandwidthType	21
8.2.7	WEthTrcv_TxPwrLvlType	22
8.2.8	WEthTrcv_RssiType	22
8.2.9	WEthTrcv_RadioModeType	22

8.3	Function definitions	23
8.3.1	WEthTrcv_Init	23
8.3.2	WEthTrcv_SetTransceiverMode	24
8.3.3	WEthTrcv_GetTransceiverMode	25
8.3.4	WEthTrcv_GetLinkState	26
8.3.5	WEthTrcv_SetRadioParams	27
8.3.6	WEthTrcv_SetChanRxParams	28
8.3.7	WEthTrcv_SetChanTxParams	29
8.3.8	WEthTrcv_GetChanRxParams	30
8.3.9	WEthTrcv_GetVersionInfo	31
8.4	Callback notifications	32
8.5	Scheduled functions	32
8.5.1	WEthTrcv_MainFunction	32
8.6	Expected interfaces	33
8.6.1	Mandatory interfaces	34
8.6.2	Optional interfaces	34
8.6.3	Configurable interfaces	34
8.7	Service Interfaces	34
9	Sequence diagrams	35
10	Configuration specification	36
10.1	How to read this chapter	36
10.2	Containers and configuration parameters	36
10.2.1	WEthTrcv	36
10.2.2	WEthTrcvConfigSet	38
10.2.3	WEthTrcvConfig	38
10.2.4	WEthTrcvDemEventParameterRefs	42
10.2.5	WEthTrcvRadioConfigSet	42
10.2.6	WEthTrcvRadioConfig	43
10.2.7	WEthTrcvRadioChannelTxQueueConfig	48
10.2.8	WEthTrcvAntennaConfigSet	49
10.2.9	WEthTrcvAntennaConfig	50
10.2.10	WEthTrcvGeneral	51
10.3	Published Information	54
A	Change history of AUTOSAR traceable items	55
A.1	Traceable item history of this document according to AUTOSAR Release R23-11	55
A.1.1	Added Specification Items in R23-11	55
A.1.2	Changed Specification Items in R23-11	55
A.1.3	Deleted Specification Items in R23-11	55
A.1.4	Added Constraints in R23-11	55
A.1.5	Changed Constraints in R23-11	55
A.1.6	Deleted Constraints in R23-11	55

# 1 Introduction and functional overview

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

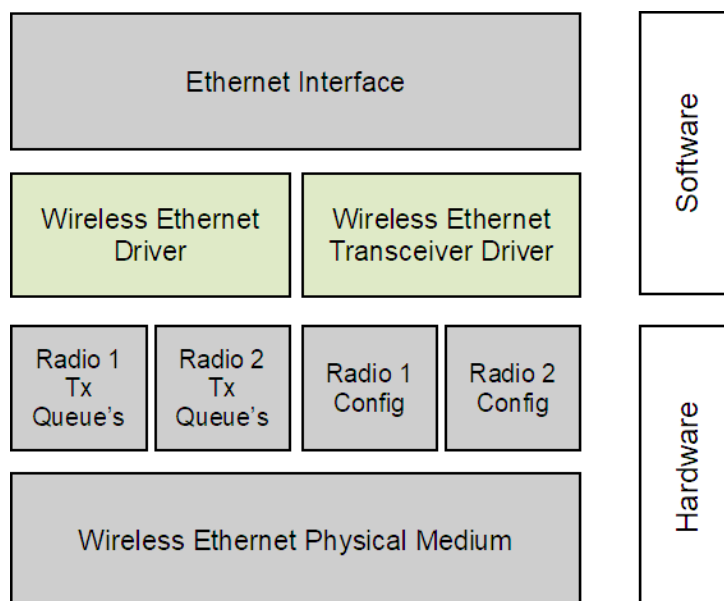
In the AUTOSAR Layered Software Architecture, the Wireless Ethernet Transceiver driver belongs to the Microcontroller Abstraction Layer, or more precisely, to the Communication Drivers.

This indicates the main task of the Wireless 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 interface provides functionality for initialization, configuration and data transmission. The configuration of the Wireless Ethernet Transceiver driver however is bus specific, since it takes into account the specific features of the communication controller.

A single Wireless Ethernet Transceiver driver module supports only one type of transceiver hardware. The Wireless Ethernet Transceiver driver's prefix requires a unique namespace. The Ethernet Interface can access different Wireless Ethernet controller types using different Wireless 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 Wireless Ethernet stack. One Ethernet Interface can access several transceivers using several Wireless Ethernet Transceiver drivers. Each transceiver may support multiple radio configurations.



**Figure 1.1: Wireless Ethernet module overview**

## 2 Acronyms and Abbreviations

The glossary below includes acronyms and abbreviations relevant to the Wireless Ethernet Transceiver Driver module that are not included in the [1, AUTOSAR glossary].

Abbreviation / Acronym:	Description:
AIFS	Arbitration Inter Frame Space
CBR	Channel Busy Ratio
CIT	Channel Idle Time
CW	Contention Window
DP	DCC Profile
EthIf	Ethernet Interface (AUTOSAR BSW module)
Eth	Ethernet Driver (AUTOSAR BSW module)
EthTrcv	Ethernet Transceiver Driver (AUTOSAR BSW module)
ISR	Interrupt Service Routine
MCG	Module Configuration Generator
WEth	Wireless Ethernet Driver (AUTOSAR BSW module)
WEthTrcv	Wireless Ethernet Transceiver (AUTOSAR BSW module)

**Table 2.1: Acronyms and abbreviations used in the scope of this Document**

## 3 Related documentation

### 3.1 Input documents & related standards and norms

- [1] Glossary  
AUTOSAR\_FO\_TR\_Glossary
- [2] General Specification of Basic Software Modules  
AUTOSAR\_CP\_SWS\_BSWGeneral
- [3] Specification of Ethernet Driver  
AUTOSAR\_CP\_SWS\_EthernetDriver
- [4] General Requirements on Basic Software Modules  
AUTOSAR\_CP\_SRS\_BSWGeneral
- [5] Requirements on Vehicle-2-X Communication  
AUTOSAR\_CP\_SRS\_V2XCommunication
- [6] Specification of Ethernet Transceiver Driver  
AUTOSAR\_CP\_SWS\_EthernetTransceiverDriver
- [7] Specification of Default Error Tracer  
AUTOSAR\_CP\_SWS\_DefaultErrorTracer
- [8] Specification of Ethernet Interface  
AUTOSAR\_CP\_SWS\_EthernetInterface

### 3.2 Related specification

AUTOSAR provides a General Specification on Basic Software modules [2, SWS BSW General], which is also valid for Wireless Ethernet Transceiver.

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

Furthermore, this document uses the Ethernet Transceiver Driver as a base for the requirements, APIs and configuration, because the wired and the wireless use case have many things (but not all) in common. The term "Ethernet Transceiver Driver" as used in this document describes the class of Ethernet drivers regardless of the used physical layer and means Wireless as well as Wired Ethernet Transceiver Drivers.



## **4 Constraints and assumptions**

### **4.1 Limitations**

The Microcontroller Abstraction Layer Multi-Core Distribution Concept is implemented as "draft" in this software specification. Refer to chapter [10](#) for more information.

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

The Wireless Ethernet Transceiver Driver is intended to be used for wireless access of customer hardware (Access Point mode) and to abstract the hardware for wireless access of Vehicle-2-X (V2X) applications / BSW Modules (using a meshed network).

## 5 Dependencies to other modules

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

Modules that use Wireless Ethernet Transceiver Driver module:

- Ethernet Interface (EthIf)

Modules used by the Wireless Ethernet Transceiver Driver module:

- Wireless Ethernet Controller Driver (WEth) to abstract the hardware access via an transceiver dependent interface
- Typically the V2X modem and/or the transceiver hardware is an external device that is accessed by an existing communication driver such as SPI.

## 6 Requirements Tracing

Requirement IDs within this document have an encoding to state where each requirement has its origin:

- SWS items starting with a leading 0 (SWS\_WEth\_0xxxx) are inherited from the [3, SWS Ethernet Driver].
- SWS items starting with a leading 1 (SWS\_WEth\_1xxxx) are module specific and not inherited.
- SWS items starting with a leading 2 (SWS\_WEth\_2xxxx) are inherited from C2C-CC Basic System Profile

The following tables reference the requirements specified in [4], [5] and links to the fulfillment of these. Please note that if column “Satisfied by” is empty for a specific requirement this means that this requirement is not fulfilled by this document.

Requirement	Description	Satisfied by
[SRS_BSW_00323]	All AUTOSAR Basic Software Modules shall check passed API parameters for validity	[SWS_WEthTrcv_00007]
[SRS_BSW_00327]	Error values naming convention	[SWS_WEthTrcv_00017]
[SRS_BSW_00339]	Reporting of production relevant error status	[SWS_WEthTrcv_00105]
[SRS_BSW_00413]	An index-based accessing of the instances of BSW modules shall be done	[SWS_WEthTrcv_00003] [SWS_WEthTrcv_10001]
[SRS_BSW_00432]	Modules should have separate main processing functions for read/receive and write/transmit data path	[SWS_WEthTrcv_10057]
[SRS_BSW_00487]	Errors for module initialization shall follow a naming rule	[SWS_WEthTrcv_10027] [SWS_WEthTrcv_10034] [SWS_WEthTrcv_10042] [SWS_WEthTrcv_10050]
[SRS_V2X_00010]	The implementation of the V2X system shall follow additional guidance given by C2C-CC requirements	[SWS_WEthTrcv_20226] [SWS_WEthTrcv_20244]
[SRS_V2X_00232]	The V2X system shall cooperate with tolling zone stations in vicinity	[SWS_WEthTrcv_10059]
[SRS_V2X_00245]	The V2X system shall support per-packet transmission power control	[SWS_WEthTrcv_20246]
[SRS_V2X_00322]	The V2X system shall provide services to avoid channel congestion of the shared media	[SWS_WEthTrcv_10057]
[SRS_V2X_00391]	The V2X system's access layer shall be ITS-G5 compliant	[SWS_WEthTrcv_10026] [SWS_WEthTrcv_10039] [SWS_WEthTrcv_10041] [SWS_WEthTrcv_10049] [SWS_WEthTrcv_10058] [SWS_WEthTrcv_10059] [SWS_WEthTrcv_10060] [SWS_WEthTrcv_10061] [SWS_WEthTrcv_10062] [SWS_WEthTrcv_10063] [SWS_WEthTrcv_10064] [SWS_WEthTrcv_10065] [SWS_WEthTrcv_10066]

**Table 6.1: RequirementsTracing**

## 7 Functional specification

The Wireless Ethernet Transceiver driver sets up the radio for wireless communications.

### 7.1 Wireless Ethernet BSW stack

As part of the AUTOSAR Layered Software Architecture (see Figure 1.1), the Wireless Ethernet BSW modules also form a layered software stack. The Ethernet Interface module accesses several transceivers using the Wireless Ethernet transceiver Driver layer, which can be made up of several Wireless Ethernet Transceiver Drivers modules.

#### 7.1.1 Indexing scheme

Users of the Wireless Ethernet Driver identify controller resources using an indexing scheme as described in the [6, SWS Ethernet Transceiver Driver].

**[SWS\_WEthTrcv\_00003]** [The Wireless Ethernet Transceiver Driver is using a zero-based index to abstract the access for upper software layers. The parameter WEthTrcvId within configuration corresponds to parameter TrcvId used in the API.] ([SRS\\_BSW\\_00413](#))

**[SWS\_WEthTrcv\_10001]** [The Wireless Ethernet Transceiver Driver is using a zero-based index to abstract the access to Radios for upper software layers. The parameter WEthTrcvRadioid within configuration corresponds to parameter Radioid used in the API.] ([SRS\\_BSW\\_00413](#))

#### 7.1.2 Requirements

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

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

**[SWS\_WEthTrcv\_00007]** [In case development error detection is enabled for the Wireless Ethernet Transceiver Driver module: The Wireless Ethernet Transceiver Driver module shall check API parameters for validity and report detected errors to the DET.] ([SRS\\_BSW\\_00323](#))

DET API functions are specified in the [7, SWS Default Error Tracer].

### 7.1.3 Transceiver Parameters

**[SWS\_WEthTrcv\_10026]** [The function WEthTrcv\_SetRadioParams shall set properties of type WEthTrcv\_SetRadioParamIdType to the access layer of a specific wireless radio indexed by Radioid.] ([SRS\\_V2X\\_00391](#))

**[SWS\_WEthTrcv\_10039]** [The function WEthTrcv\_SetChanRxParams shall set properties of type WEthTrcv\_SetChanRxParamIdType to a specific wireless channel within a wireless radio indexed by Radioid.] ([SRS\\_V2X\\_00391](#))

**[SWS\_WEthTrcv\_10041]** [The function WEthTrcv\_SetChanTxParams shall set of type WEthTrcv\_SetChanTxParamIdType to a specific wireless channel within a wireless radio indexed by Radioid.] ([SRS\\_V2X\\_00391](#))

**[SWS\_WEthTrcv\_10049]** [The function EthTrcv\_GetChanRxParams shall provide properties of type WEthTrcv\_GetChanRxParamIdType of a specific wireless channel within a wireless radio indexed by Radioid.] ([SRS\\_V2X\\_00391](#))

### 7.1.4 Key/Value Parameter Mapping

**[SWS\_WEthTrcv\_10066]** [For unique reference to transmission and reception parameters, unique enumeration IDs shall be used within this module.] ([SRS\\_V2X\\_00391](#))

**[SWS\_WEthTrcv\_10058]** [Functions using the type WEthTrcv\_SetRadioParamIdType shall use a generic list of uint32 values for the list of corresponding values.] ([SRS\\_V2X\\_00391](#))

**[SWS\_WEthTrcv\_10059]** [

ParamId	ParamValue Type
WETHTRCV_SETRADIOPID_SEL_TRCV_CHCFG	uint8
WETHTRCV_SETRADIOPID_SET_CHCFGID	uint8
WETHTRCV_SETRADIOPID_TOLLINGZONE_INFO	uint8

Functions using the WEthTrcv\_SetRadioParamIdType shall use the type mapping provided by the table above for the corresponding values.

] ([SRS\\_V2X\\_00232](#), [SRS\\_V2X\\_00391](#))

**[SWS\_WEthTrcv\_10060]** [Functions using the type WEthTrcv\_SetChanRxParamIdType shall use a generic list of uint32 values for the list of corresponding values.] ([SRS\\_V2X\\_00391](#))

**[SWS\_WEthTrcv\_10061]** [

ParamId	ParamValue Type
WETHTRCV_SETCHRXPID_BITRATE	uint8
WETHTRCV_SETCHRXPID_BANDWIDTH	WEthTrcv_BandwidthType
WETHTRCV_SETCHRXPID_FREQ	uint16
WETHTRCV_SETCHRXPID_CSPWRTRESH	WEthTrcv_RssiType
WETHTRCV_SETCHRXPID_RADIO_MODE	WEthTrcv_RadioModeType
WETHTRCV_SETCHRXPID_ANTENNA	uint8

Functions using the WEthTrcv\_SetChanRxParamIdType shall use the type mapping provided by the table above for the corresponding values.

]([SRS\\_V2X\\_00391](#))

**[SWS\_WEthTrcv\_10062]** [Functions using the type WEthTrcv\_SetChanTxParamIdType shall use a generic list of uint32 values for the list of corresponding values.]([SRS\\_V2X\\_00391](#))

**[SWS\_WEthTrcv\_10063]** [

ParamId	ParamValue Type
WETHTRCV_SETCHTXPID_BITRATE	uint8
WETHTRCV_SETCHTXPID_BANDWIDTH	WEthTrcv_BandwidthType
WETHTRCV_SETCHTXPID_TXPOWER	WEthTrcv_TxPwrLvlType
WETHTRCV_SETCHTXPID_DCC_CBR	uint8
WETHTRCV_SETCHTXPID_TXQSEL	uint8
WETHTRCV_SETCHTXPID_TXQCFG_AIFSN	uint8
WETHTRCV_SETCHTXPID_TXQCFG_CWMIN	uint8
WETHTRCV_SETCHTXPID_TXQCFG_CWMAX	uint16
WETHTRCV_SETCHTXPID_TXQCFG_TXOP	uint8
WETHTRCV_SETCHTXPID_RADIO_MODE	WEthTrcv_RadioModeType
WETHTRCV_SETCHTXPID_ANTENNA	uint8
WETHTRCV_SETCHTXPID_PACKET_INTERVAL	uint16
WETHTRCV_SETCHTXPID_DCC_STATE	uint8

Functions using the WEthTrcv\_SetChanTxParamIdType shall use the type mapping provided by the table above for the corresponding values.

]([SRS\\_V2X\\_00391](#))

**[SWS\_WEthTrcv\_10064]** [Functions using the type WEthTrcv\_GetChanRxParamIdType shall use a generic list of uint32 values for the list of corresponding values.]([SRS\\_V2X\\_00391](#))

**[SWS\_WEthTrcv\_10065]** [

ParamId	ParamValue Type
WETHTRCV_GETCHRXPID_CBR	uint8
WETHTRCV_GETCHRXPID_CIT	uint16

Functions using the WEthTrcv\_GetChanRxParamIdType shall use the type mapping provided by the table above for the corresponding values.

]([SRS\\_V2X\\_00391](#))

### 7.1.5 MainFunction

**[SWS\_WEthTrcv\_10057]** [The MainFunction is used for hardware / software implementation specific execution of cyclic tasks.

In case of V2X the MainFunction is used to get Information of the current wireless transceiver channel status (CBR) if the transceiver is not indirectly accessed via the wireless Ethernet driver.]([SRS\\_V2X\\_00322](#), [SRS\\_BSW\\_00432](#))

### 7.1.6 V2X Specific Transceiver Requirements

**[SWS\_WEthTrcv\_20226]** [RF output power of the WEthTrcv module shall be adjustable.]([SRS\\_V2X\\_00010](#))

**[SWS\_WEthTrcv\_20244]** [The WEthTrcv module shall abide by the following maximum message rates:

- For the relaxed state: the sum of all messages sent on DP1, DP2 and DP3 while in relaxed state shall not surpass  $R_{\max\_relaxed} = 16.7$  messages per second. Message bursts are allowed for DP0 with  $R_{Burst} = 20$  messages per second, with a maximum duration of  $T_{Burst} = 1$  seconds, and may only take place every  $T_{BurstPeriod} = 10$  seconds. Thus, adding DP0 messages, the maximum message rate amounts to  $R_{\max\_relaxed} = 36.7$  messages per second.

]([SRS\\_V2X\\_00010](#))

**[SWS\_WEthTrcv\_20246]** [The WEthTrcv module shall reduce its transmission power to  $P_{Toll} = 10$  dBm as soon as the protected communication zone is entered, and without changing any other DCC transmission parameters. DP0 messages are excluded from this restriction.]([SRS\\_V2X\\_00245](#))

### 7.1.7 Wake-up support

There is currently no efficient concept for technologies like Wake on Wireless LAN. Wireless Wake-up is therefore not supported.

## 7.2 Error Classification

Section "Error Handling" of the document [2] "General Specification of Basic Software Modules" describes the error handling of the Basic Software in detail. Above all, it constitutes a classification scheme consisting of five error types which may occur in BSW modules.

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

### 7.2.1 Development Errors

[SWS\_WEthTrcv\_00017] Definiton of development errors in module WEthTrcv [

<i>Type of error</i>	<i>Related error code</i>	<i>Error value</i>
Invalid transceiver index	WETHTRCV_E_INV_TRCV_ID	0x01
WEthTrcv module was not initialized	WETHTRCV_E_UNINIT	0x02
Invalid pointer in parameter list	WETHTRCV_E_PARAM_POINTER	0x03

]([SRS\\_BSW\\_00327](#))

### 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\_WEthTrcv\_00105]** [

Error Name:	WETHTRCV_E_ACCESS	
Short Description:	Wireless Ethernet Transceiver Access Failure.	
Long Description:	Monitors the access to the Wireless Ethernet Transceiver if a transceiver hardware is separate from the baseband modem hardware.	
Detection Criteria:	Fail	When access to the Wireless 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 Wireless Ethernet Transceiver succeeds the module shall report the extended production error with event status DEM_EVENT_STATUS_PREPASSED to DEM.
Secondary Parameters:	None.	
Time Required:	None.	
Monitor Frequency	None.	

] ([SRS\\_BSW\\_00339](#))

### 7.3 Security Events

The module does not report security events.

## 8 API specification

### 8.1 Imported types

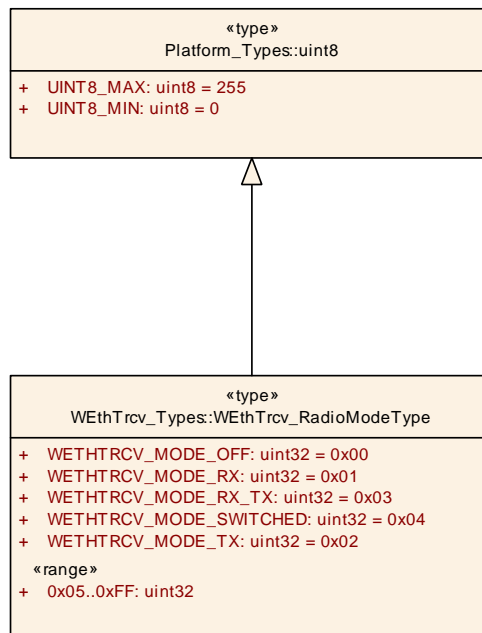
In this chapter all types included from the following files are listed.

**[SWS\_WEthTrcv\_00027] Definition of imported datatypes of module WEthTrcv [**

Module	Header File	Imported Type
Dem	Rte_Dem_Type.h	Dem_EventIdType
	Rte_Dem_Type.h	Dem_EventStatusType
Eth	Eth_GeneralTypes.h	Eth_ModeType
EthTrcv	Eth_GeneralTypes.h	EthTrcv_LinkStateType
Std	Std_Types.h	Std_ReturnType
	Std_Types.h	Std_VersionInfoType

]

### 8.2 Type definitions



**Figure 8.1**

### 8.2.1 WEthTrcv\_ConfigType

#### [SWS\_WEthTrcv\_00098] Definition of datatype WEthTrcv\_ConfigType [

<b>Name</b>	WEthTrcv_ConfigType
<b>Kind</b>	Structure
<b>Description</b>	Implementation specific structure of the post build configuration
<b>Available via</b>	WEthTrcv.h

]()

### 8.2.2 WEthTrcv\_SetRadioParamIdType

#### [SWS\_WEthTrcv\_10008] Definition of datatype WEthTrcv\_SetRadioParamIdType [

<b>Name</b>	WEthTrcv_SetRadioParamIdType		
<b>Kind</b>	Enumeration		
<b>Range</b>	WETHTRCV_SETRADIOPID_SEL_TRCV_CHCFG	0x01	Select which channel config within the transceiver should be set within multichannel context mode
	WETHTRCV_SETRADIOPID_SET_CHCFGID	0x02	Switch to another channel config in single channel context mode or in multichannel context mode (requires previous selection of channel config in transceiver)
	WETHTRCV_SETRADIOPID_TOLLINGZONE_INFO	0x03	Information of entering and leaving a Tolling Zone Area
<b>Description</b>	Wireless radio settings for the transceiver		
<b>Available via</b>	WEth_GeneralTypes.h		

]()

### 8.2.3 WEthTrcv\_SetChanRxParamIdType

#### [SWS\_WEthTrcv\_10009] Definition of datatype WEthTrcv\_SetChanRxParamIdType [

<b>Name</b>	WEthTrcv_SetChanRxParamIdType		
<b>Kind</b>	Enumeration		
<b>Range</b>	WETHTRCV_SETCHRXPID_BITRATE	0x00	Bitrate
	WETHTRCV_SETCHRXPID_BANDWIDTH	0x01	Bandwidth
	WETHTRCV_SETCHRXPID_FREQ	0x02	Center frequency of a channel





	WETHRCV_ SETCHRXPID_ CSPWRTRESH	0x03	Parameter for Rx busy detection
	WETHRCV_ SETCHRXPID_RADIO_ MODE	0x04	Param for Rx Radio Mode
	WETHRCV_ SETCHRXPID_ANTENNA	0x05	Rx Antenna Id
<b>Description</b>	Wireless channel settings for the receive side		
<b>Available via</b>	WEth_GeneralTypes.h		

]()

## 8.2.4 WEthTrcv\_SetChanTxParamIdType

[SWS\_WEthTrcv\_10011] Definition of datatype WEthTrcv\_SetChanTxParamIdType

<b>Name</b>	WEthTrcv_SetChanTxParamIdType		
<b>Kind</b>	Enumeration		
<b>Range</b>	WETHRCV_ SETCHTXPID_BITRATE	0x00	Bitrate
	WETHRCV_ SETCHTXPID_ BANDWIDTH	0x01	Bandwidth
	WETHRCV_ SETCHTXPID_TXPOWER	0x02	Transmission power
	WETHRCV_ SETCHTXPID_DCC_CBR	0x03	Param for Channel Busy Ratio for DCC
	WETHRCV_ SETCHTXPID_TXQSEL	0x04	Selection of the transmit queue for that the settings should be set
	WETHRCV_ SETCHTXPID_TXQCFG_ AIFSN	0x05	Arbitration inter-frame-spacing number (multiplier with value of 0 to 15)
	WETHRCV_ SETCHTXPID_TXQCFG_ CWMIN	0x06	Contention window min
	WETHRCV_ SETCHTXPID_TXQCFG_ CWMAX	0x07	Contention window max
	WETHRCV_ SETCHTXPID_TXQCFG_ TXOP	0x08	TXOP duration limit [μs] divided by 32
	WETHRCV_ SETCHTXPID_RADIO_ MODE	0x09	Param for Tx Radio Mode
	WETHRCV_ SETCHTXPID_ANTENNA	0x0A	Tx Antenna Id
	WETHRCV_ SETCHTXPID_PACKET_ INTERVAL	0x0C	Packet interval for transmission interspace



△

	WETHTRCV_ SETCHTXPID_DCC_STATE	0x0D	State of DCC state machine
<b>Description</b>	-		
<b>Available via</b>	WEth_GeneralTypes.h		

]()

## 8.2.5 WEthTrcv\_GetChanRxParamIdType

[SWS\_WEthTrcv\_10007] Definition of datatype WEthTrcv\_GetChanRxParamIdType [

<b>Name</b>	WEthTrcv_GetChanRxParamIdType		
<b>Kind</b>	Enumeration		
<b>Range</b>	WETHTRCV_ GETCHRXPID_CBR	0x00	Parameter Id for Channel Busy Ratio
	WETHTRCV_ GETCHRXPID_CIT	0x01	Parameter Id for Channel Idle Time
<b>Description</b>	Wireless channel properties of the receive side		
<b>Available via</b>	WEth_GeneralTypes.h		

]()

## 8.2.6 WEthTrcv\_BandwidthType

[SWS\_WEthTrcv\_10012] Definition of datatype WEthTrcv\_BandwidthType [

<b>Name</b>	WEthTrcv_BandwidthType		
<b>Kind</b>	Type		
<b>Derived from</b>	uint32		
<b>Range</b>	0x0000004..0xFFFFFFFF	-	Invalid
	WETHTRCV_BW_5MHz	0x00	Indicates 5 MHz
	WETHTRCV_BW_10MHz	0x01	Indicates 10 MHz
	WETHTRCV_BW_20MHz	0x02	Indicates 20 MHz
	WETHTRCV_BW_40MHz	0x03	Indicates 40 MHz
<b>Description</b>	Bandwidth of a radio channel		
<b>Available via</b>	WEth_GeneralTypes.h		

]()

## 8.2.7 WEthTrcv\_TxPwrLvIType

### [SWS\_WEthTrcv\_10014] Definition of datatype WEthTrcv\_TxPwrLvIType [

<b>Name</b>	WEthTrcv_TxPwrLvIType		
<b>Kind</b>	Type		
<b>Derived from</b>	uint16		
<b>Range</b>	0..399	–	Valid values of 0.5db with an offset of -100dBm
	400..65535	–	Invalid
<b>Description</b>	Power of frame, in 0.5 dBm units, raw value 0 equals -100 dBm		
<b>Available via</b>	WEth_GeneralTypes.h		

]()

## 8.2.8 WEthTrcv\_RssiType

### [SWS\_WEthTrcv\_10016] Definition of datatype WEthTrcv\_RssiType [

<b>Name</b>	WEthTrcv_RssiType		
<b>Kind</b>	Type		
<b>Derived from</b>	uint16		
<b>Range</b>	0..399	–	Valid values of 0.5db with an offset of -100dBm
	400..65535	–	Invalid
<b>Description</b>	Power of frame, in 0.5 dBm units, raw value 0 equals -100 dBm		
<b>Available via</b>	WEth_GeneralTypes.h		

]()

## 8.2.9 WEthTrcv\_RadioModeType

### [SWS\_WEthTrcv\_10018] Definition of datatype WEthTrcv\_RadioModeType [

<b>Name</b>	WEthTrcv_RadioModeType		
<b>Kind</b>	Type		
<b>Derived from</b>	uint8		
<b>Range</b>	0x05..0xFF	–	Invalid
	WETHTRCV_MODE_OFF	0x00	Radio is off
	WETHTRCV_MODE_RX	0x01	Receive is on
	WETHTRCV_MODE_TX	0x02	Transmit is on
	WETHTRCV_MODE_RX_TX	0x03	Receive and Transmit is on
	WETHTRCV_MODE_SWITCHED	0x04	Radio channel switching is on
<b>Description</b>	Radio operation mode with multiple radio channel configurations		
<b>Available via</b>	WEth_GeneralTypes.h		

]()

## 8.3 Function definitions

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

### 8.3.1 WEthTrcv\_Init

[SWS\_WEthTrcv\_00028] Definition of API function WEthTrcv\_Init [

<b>Service Name</b>	WEthTrcv_Init	
<b>Syntax</b>	<pre>void WEthTrcv_Init (     const WEthTrcv_ConfigType* CfgPtr )</pre>	
<b>Service ID [hex]</b>	0x01	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CfgPtr	Points to the implementation specific structure
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	None	
<b>Description</b>	Initializes the Wireless Ethernet Transceiver Driver	
<b>Available via</b>	WEthTrcv.h	

]()

[SWS\_WEthTrcv\_10022] [The function shall store the access to the configuration structure for subsequent API calls.]()

[SWS\_WEthTrcv\_00035] [The function shall configure all transceiver configuration parameters (e.g. baud rate, duplex mode, automatic negotiation, ...).]()

[SWS\_WEthTrcv\_00030] [The function shall change the state of the component from WETHTRCV\_STATE\_UNINIT to WETHTRCV\_STATE\_INIT.]()

[SWS\_WEthTrcv\_00040] [The function shall check the access to the Wireless Ethernet Transceiver. If the check fails, the function shall raise the production error WETHTRCV\_E\_ACCESS and return E\_NOT\_OK, otherwise pass the production error WETHTRCV\_E\_ACCESS and return E\_OK.]()

[SWS\_WEthTrcv\_00032] [Caveat: The API has to be called during initialization.]()

### 8.3.2 WEthTrcv\_SetTransceiverMode

#### [SWS\_WEthTrcv\_00042] Definition of API function WEthTrcv\_SetTransceiver Mode

<b>Service Name</b>	WEthTrcv_SetTransceiverMode	
<b>Syntax</b>	Std_ReturnType WEthTrcv_SetTransceiverMode ( uint8 TrcvId, Eth_ModeType TrcvMode )	
<b>Service ID [hex]</b>	0x03	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	TrcvId	Index of the transceiver within the context of the Ethernet Transceiver Driver
	TrcvMode	ETH_MODE_DOWN: disable the wireless Ethernet transceiver ETH_MODE_ACTIVE: enable the wireless Ethernet transceiver
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	E_OK: Service accepted E_NOT_OK: Service denied
<b>Description</b>	Enables / disables the indexed transceiver	
<b>Available via</b>	WEthTrcv.h	

]()

**[SWS\_WEthTrcv\_10023]** [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 WEthTrcv\_MainFunction.]()

**[SWS\_WEthTrcv\_00044]** [If development error detection is enabled: The function shall check that the service WEthTrcv\_Init was previously called.

If the check fails, the function shall raise the development error WETHTRCV\_E\_UNINIT otherwise (if DET is disabled) return E\_NOT\_OK.]()

**[SWS\_WEthTrcv\_00045]** [If development error detection is enabled: The function shall check the parameter TrcvIdx for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_INV\_TRCV\_IDX otherwise (if DET is disabled) return E\_NOT\_OK.]()

**[SWS\_WEthTrcv\_00046]** [The function shall be pre compile time configurable On/Off by the configuration parameter: WEthTrcvSetTransceiverModeApi.]()

**[SWS\_WEthTrcv\_00107]** [If the transceiver is already in the requested mode E\_OK shall be returned and no development error shall be raised.]()

**[SWS\_WEthTrcv\_00104]** [The function shall check the access to the Wireless Ethernet transceiver. If the check fails, the function shall raise the production error WETHTRCV\_E\_ACCESS and return E\_NOT\_OK, otherwise pass the production error WETHTRCV\_E\_ACCESS and return E\_OK.]()



**[SWS\_WEthTrcv\_00047]** [Caveat: The function requires previous transceiver initialization (EthTrcv\_Init).] ()

### 8.3.3 WEthTrcv\_GetTransceiverMode

**[SWS\_WEthTrcv\_00048]** Definition of API function WEthTrcv\_GetTransceiverMode [

<b>Service Name</b>	WEthTrcv_GetTransceiverMode	
<b>Syntax</b>	Std_ReturnType WEthTrcv_GetTransceiverMode ( uint8 TrcvId, Eth_ModeType* TrcvModePtr )	
<b>Service ID [hex]</b>	0x04	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	TrcvId	Index of the transceiver within the context of the Wireless Ethernet Transceiver Driver
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	TrcvModePtr	ETH_MODE_DOWN: the wireless Ethernet transceiver is disabled ETH_MODE_ACTIVE: the wireless Ethernet transceiver is enabled
<b>Return value</b>	Std_ReturnType	E_OK: success E_NOT_OK: wireless Ethernet transceiver could not be initialized
<b>Description</b>	Obtains the state of the indexed transceiver	
<b>Available via</b>	WEthTrcv.h	

] ()

**[SWS\_WEthTrcv\_10024]** [The function shall read the current transceiver mode.] ()

**[SWS\_WEthTrcv\_00050]** [If development error detection is enabled: The function shall check that the service EthTrcv\_Init was previously called.

If the check fails, the function shall raise the development error WETHTRCV\_E\_UNINIT otherwise (if DET is disabled) return E\_NOT\_OK.] ()

**[SWS\_WEthTrcv\_00051]** [If development error detection is enabled: The function shall check the parameter TrcvIdx for being valid.

If the check fails, the function shall raise the development error WETHTRCV\_E\_INV\_TRCV\_IDX otherwise (if DET is disabled) return E\_NOT\_OK.] ()

**[SWS\_WEthTrcv\_00052]** [If development error detection is enabled: The function shall check the parameter TrcvModePtr for being valid.

If the check fails, the function shall raise the development error WETHTRCV\_E\_PARAM\_POINTER otherwise (if DET is disabled) return E\_NOT\_OK.] ()

**[SWS\_WEthTrcv\_00053]** [The function shall be pre compile time configurable On/Off by the configuration parameter: WEthTrcvGetTransceiverModeApi.] ()

**[SWS\_WEthTrcv\_00054]** [Caveat: The function requires previous transceiver initialization (WEthTrcv\_Init).] ()

### 8.3.4 WEthTrcv\_GetLinkState

**[SWS\_WEthTrcv\_00061]** Definition of API function WEthTrcv\_GetLinkState [

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

] ()

**[SWS\_WEthTrcv\_10073]** [The function shall read the current transceiver link state.] ()

**[SWS\_WEthTrcv\_00063]** [If development error detection is enabled: The function shall check that the service WEthTrcv\_Init was previously called.

If the check fails, the function shall raise the development error WETHTRCV\_E\_UNINIT otherwise (if DET is disabled) return E\_NOT\_OK.] ()

**[SWS\_WEthTrcv\_00064]** [If development error detection is enabled: The function shall check the parameter TrcvIdx for being valid.

If the check fails, the function shall raise the development error WETHTRCV\_E\_INV\_TRCV\_IDX otherwise (if DET is disabled) return E\_NOT\_OK.] ()

**[SWS\_WEthTrcv\_00065]** [If development error detection is enabled: The function shall check the parameter LinkStatePtr for being valid.

If the check fails, the function shall raise the development error WETHTRCV\_E\_PARAM\_POINTER otherwise (if DET is disabled) return E\_NOT\_OK.] ()

**[SWS\_WEthTrcv\_00066]** [The function shall be pre compile time configurable On/Off by the configuration parameter: WEthTrcvGetLinkStateApi.] ()

[SWS\_WEthTrcv\_00067] [Caveat: The function requires previous transceiver initialization (WEthTrcv\_Init).]()

### 8.3.5 WEthTrcv\_SetRadioParams

[SWS\_WEthTrcv\_10025] Definition of API function WEthTrcv\_SetRadioParams [

<b>Service Name</b>	WEthTrcv_SetRadioParams	
<b>Syntax</b>	<pre>Std_ReturnType WEthTrcv_SetRadioParams (     uint8 TrcvId,     const WEthTrcv_SetRadioParamIdType* ParamIds,     const uint32* ParamValue,     uint8 NumParams )</pre>	
<b>Service ID [hex]</b>	0x30	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	TrcvId	Index of the transceiver
	ParamIds	IDs of the Parameters to set
	ParamValue	Values of the Parameters to set
	NumParams	Number of Parameters to set
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	E_OK: success E_NOT_OK: failed writing parameters
<b>Description</b>	Set values related to a transceiver's wireless radio. For example, this could be the selection of the radio settings (channel, ...).	
<b>Available via</b>	WEthTrcv.h	

]()

[SWS\_WEthTrcv\_10067] [The function shall use the type mapping from SWS\_WEthTrcv\_10059 for the ParamIds and ParamValues parameters.]()

[SWS\_WEthTrcv\_10027] [If development error detection is enabled: The function shall check that the service WEthTrcv\_Init was previously called. If the check fails, the function shall raise the development error WETHTRCV\_E\_UNINIT.]([SRS\\_BSW\\_00487](#))

[SWS\_WEthTrcv\_10028] [If development error detection is enabled: The function shall check the parameter TrcvId for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_INV\_TRCV\_ID otherwise (if DET is disabled) return E\_NOT\_OK.]()

[SWS\_WEthTrcv\_10029] [If development error detection is enabled: The function shall check the parameter Radioid for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_INV\_PARAM otherwise (if DET is disabled) return E\_NOT\_OK.]()

**[SWS\_WEthTrcv\_10030]** [If development error detection is enabled: The function shall check the parameter ParamIds for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_PARAM\_POINTER.]()

**[SWS\_WEthTrcv\_10031]** [If development error detection is enabled: The function shall check the parameter ParamValues for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_PARAM\_POINTER.]()

### 8.3.6 WEthTrcv\_SetChanRxParams

**[SWS\_WEthTrcv\_10033] Definition of API function WEthTrcv\_SetChanRxParams**

[

<b>Service Name</b>	WEthTrcv_SetChanRxParams	
<b>Syntax</b>	<pre>Std_ReturnType WEthTrcv_SetChanRxParams (     uint8 TrcvId,     uint8 RadioId,     const WEthTrcv_SetChanRxParamIdType* ParamIds,     const uint32* ParamValues,     uint8 NumParams )</pre>	
<b>Service ID [hex]</b>	0x31	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	TrcvId	Index of the transceiver
	RadioId	Index of the Transceiver's Radio (including channel)
	ParamIds	IDs of the Parameters to set
	ParamValues	Values of the Parameters to set
	NumParams	Number of Parameters to set
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	E_OK: success E_NOT_OK: failed writing parameters
<b>Description</b>	Set values related to the receive direction of a transceiver's wireless channel. For example, this could be a channel parameter like the frequency.	
<b>Available via</b>	WEthTrcv.h	

]()

**[SWS\_WEthTrcv\_10068]** [The function shall use the type mapping from SWS\_WEthTrcv\_10061 for the ParamIds and ParamValues parameters.]()

**[SWS\_WEthTrcv\_10034]** [If development error detection is enabled: The function shall check that the service WEthTrcv\_Init was previously called. If the check fails, the function shall raise the development error WETHTRCV\_E\_UNINIT.] ([SRS\\_BSW\\_00487](#))

**[SWS\_WEthTrcv\_10035]** [If development error detection is enabled: The function shall check the parameter TrcvId for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_INV\_TRCV\_ID otherwise (if DET is disabled) return E\_NOT\_OK.]()

**[SWS\_WEthTrcv\_10036]** [If development error detection is enabled: The function shall check the parameter RadioId for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_INV\_PARAM otherwise (if DET is disabled) return E\_NOT\_OK.]()

**[SWS\_WEthTrcv\_10037]** [If development error detection is enabled: The function shall check the parameter ParamIds for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_PARAM\_POINTER.]()

**[SWS\_WEthTrcv\_10038]** [If development error detection is enabled: The function shall check the parameter ParamValues for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_PARAM\_POINTER.]()

### 8.3.7 WEthTrcv\_SetChanTxParams

**[SWS\_WEthTrcv\_10040]** Definition of API function WEthTrcv\_SetChanTxParams

[

<b>Service Name</b>	WEthTrcv_SetChanTxParams	
<b>Syntax</b>	<pre>Std_ReturnType WEthTrcv_SetChanTxParams (     uint8 TrcvId,     uint8 RadioId,     const WEthTrcv_SetChanTxParamIdType* TxParamIds,     const uint32* ParamValues,     uint8 NumParams )</pre>	
<b>Service ID [hex]</b>	0x32	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	TrcvId	Index of the transceiver
	RadioId	Index of the Transceiver's Radio (including channel)
	TxParamIds	IDs of the Parameters to set
	ParamValues	Values of the Parameters to set
	NumParams	Number of Parameters to set
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	E_OK: success E_NOT_OK: failed writing parameters
<b>Description</b>	Set values related to the transmit direction of a transceiver's wireless channel. For example, this could be the bitrate of a channel.	
<b>Available via</b>	WEthTrcv.h	

]()

**[SWS\_WEthTrcv\_10069]** [The function shall use the type mapping from SWS\_WEthTrcv\_10063 for the TxParamIds and ParamValues parameters.]()

**[SWS\_WEthTrcv\_10042]** [If development error detection is enabled: The function shall check that the service WEthTrcv\_Init was previously called. If the check fails, the function shall raise the development error WETHTRCV\_E\_UNINIT.]([SRS\\_BSW\\_00487](#))

**[SWS\_WEthTrcv\_10043]** [If development error detection is enabled: The function shall check the parameter TrcvId for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_INV\_TRCV\_ID otherwise (if DET is disabled) return E\_NOT\_OK.]()

**[SWS\_WEthTrcv\_10044]** [If development error detection is enabled: The function shall check the parameter RadioId for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_INV\_PARAM otherwise (if DET is disabled) return E\_NOT\_OK.]()

**[SWS\_WEthTrcv\_10045]** [If development error detection is enabled: The function shall check the parameter TxParamIds for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_PARAM\_POINTER.]()

**[SWS\_WEthTrcv\_10046]** [If development error detection is enabled: The function shall check the parameter ParamValues for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_PARAM\_POINTER.]()

### 8.3.8 WEthTrcv\_GetChanRxParams

**[SWS\_WEthTrcv\_10048]** Definition of API function WEthTrcv\_GetChanRxParams

[

<b>Service Name</b>	WEthTrcv_GetChanRxParams	
<b>Syntax</b>	<pre>Std_ReturnType WEthTrcv_GetChanRxParams (     uint8* TrcvId,     uint8 RadioId,     const WEthTrcv_GetChanRxParamIdType* ParamIds,     uint32* ParamValues,     uint8 NumParams )</pre>	
<b>Service ID [hex]</b>	0x33	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	TrcvId	Index of the transceiver
	RadioId	Index of the Transceiver's Radio ( including channel)
	ParamIds	IDs of the Parameters to read
	NumParams	Number of Parameters to read
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	ParamValues	Values of the requested Parameters
<b>Return value</b>	Std_ReturnType	E_OK: success E_NOT_OK: failed reading parameters
<b>Description</b>	Read values related to the receive direction of the transceiver. For example, this could be a Channel Busy Ratio (CBR) or the average Channel Idle Time (CIT).	
<b>Available via</b>	WEthTrcv.h	

]()

**[SWS\_WEthTrcv\_10070]** [The function shall use the type mapping from SWS\_WEthTrcv\_10065 for the ParamIds and ParamValues parameters.]()

**[SWS\_WEthTrcv\_10050]** [If development error detection is enabled: The function shall check that the service WEthTrcv\_Init was previously called. If the check fails, the function shall raise the development error WETHTRCV\_E\_UNINIT.] ([SRS\\_BSW\\_00487](#))

**[SWS\_WEthTrcv\_10051]** [If development error detection is enabled: The function shall check the parameter TrcvId for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_INV\_TRCV\_ID otherwise (if DET is disabled) return E\_NOT\_OK.]()

**[SWS\_WEthTrcv\_10052]** [If development error detection is enabled: The function shall check the parameter Radioid for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_INV\_PARAM otherwise (if DET is disabled) return E\_NOT\_OK.]()

**[SWS\_WEthTrcv\_10053]** [If development error detection is enabled: The function shall check the parameter ParamIds for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_PARAM\_POINTER.]()

**[SWS\_WEthTrcv\_10054]** [If development error detection is enabled: The function shall check the parameter ParamValues for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_PARAM\_POINTER.]()

### 8.3.9 WEthTrcv\_GetVersionInfo

**[SWS\_WEthTrcv\_00082]** Definition of API function WEthTrcv\_GetVersionInfo [

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

]()

**[SWS\_WEthTrcv\_00093]** [If development error detection is enabled: The function shall check the parameter VersionInfoPtr for being valid. If the check fails, the function shall raise the development error WETHTRCV\_E\_PARAM\_POINTER.]()

## 8.4 Callback notifications

The Wireless Ethernet Transceiver Driver does not provide any callback functions.

## 8.5 Scheduled functions

These functions are directly called by Basic Software Scheduler. The following functions shall have no return value and no parameter. All functions shall be non reentrant.

### 8.5.1 WEthTrcv\_MainFunction

[SWS\_WEthTrcv\_00106] Definition of scheduled function WEthTrcv\_MainFunction [

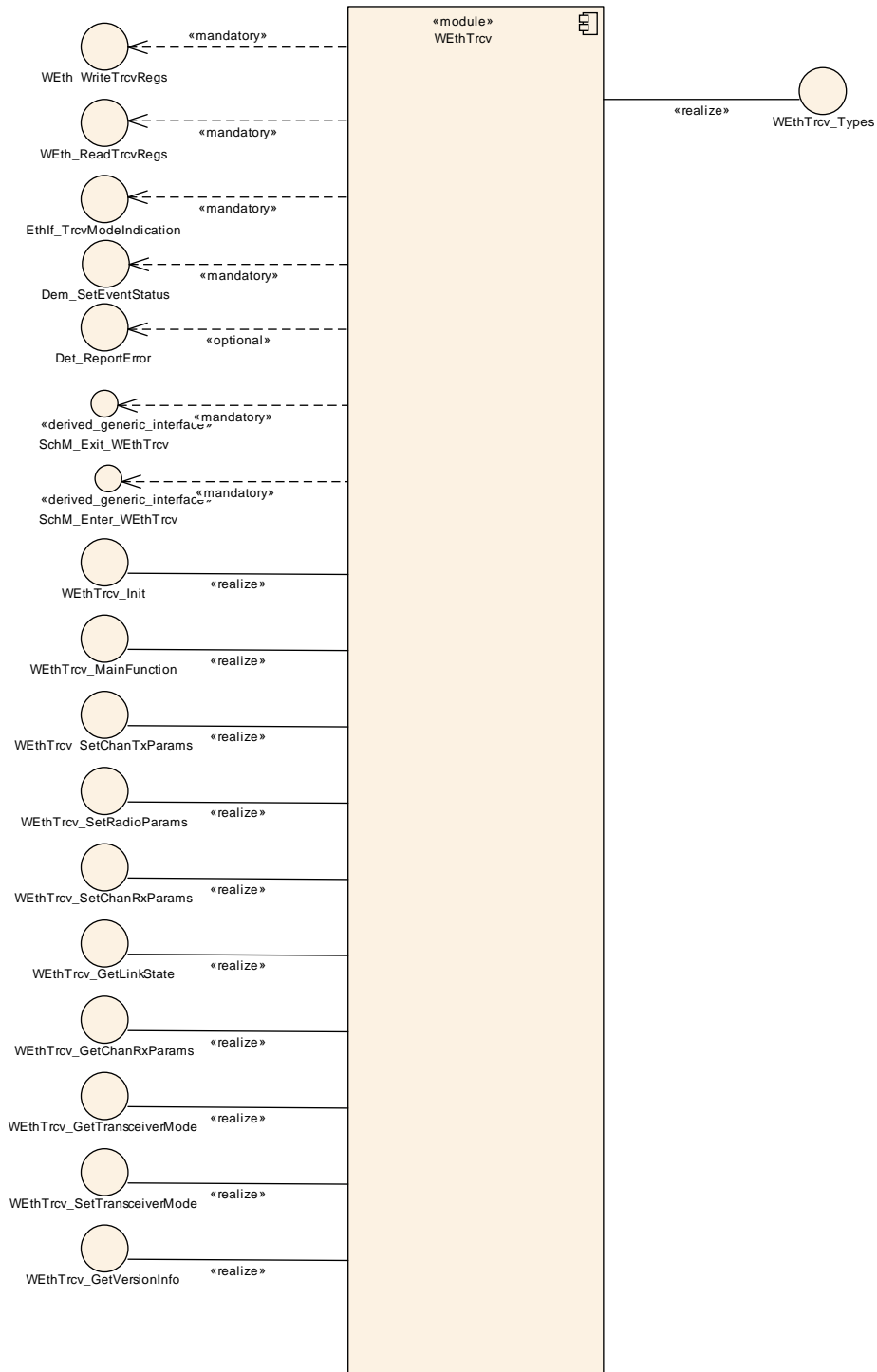
<b>Service Name</b>	WEthTrcv_MainFunction
<b>Syntax</b>	void WEthTrcv_MainFunction ( void )
<b>Service ID [hex]</b>	0x0c
<b>Description</b>	Used for polling state changes. Calls EthIf_TrcvModeIndication when the transceiver mode changed.
<b>Available via</b>	SchM_WEthTrcv.h

]()



## 8.6 Expected interfaces

In this chapter all external interfaces required from other modules are listed.



**Figure 8.2**

### 8.6.1 Mandatory interfaces

Note: This section defines all interfaces, which are required to fulfill the core functionality of the module.

#### [SWS\_WEthTrcv\_00085] Definition of mandatory interfaces in module WEthTrcv

[

<i>API Function</i>	<i>Header File</i>	<i>Description</i>
Dem_SetEventStatus	Dem.h	Called by SW-Cs or BSW modules to report monitor status information to the Dem. BSW modules calling Dem_SetEventStatus can safely ignore the return value. This API will be available only if ((Dem/Dem ConfigSet/DemEventParameter/DemEvent ReportingType) == STANDARD_REPORTING)
EthIf_TrcvModeIndication	EthIf.h	Called asynchronously when a mode change has been read out. If the function is triggered by previous call of EthTrcv_SetTransceiverMode it can directly be called within the trigger function.
SchM_Enter_WEthTrcv	SchM_<Mip>.h	Invokes the SchM_Enter function to enter a module local exclusive area.
SchM_Exit_WEthTrcv	SchM_<Mip>.h	Invokes the SchM_Exit function to exit an exclusive area.
WEth_ReadTrcvRegs	WEth.h	Reads a transceiver register
WEth_WriteTrcvRegs	WEth.h	Configures a transceivers registers or triggers a function offered by the receiver

]()

### 8.6.2 Optional interfaces

This section defines all interfaces, which are required to fulfill an optional functionality of the module.

#### [SWS\_WEthTrcv\_00120] Definition of optional interfaces in module WEthTrcv

[

<i>API Function</i>	<i>Header File</i>	<i>Description</i>
Det_ReportError	Det.h	Service to report development errors.

]()

### 8.6.3 Configurable interfaces

The Wireless Ethernet Transceiver Driver does not use configurable interfaces.

## 8.7 Service Interfaces

The Wireless Ethernet Transceiver Driver does not provide any Service Interface.

## 9 Sequence diagrams

The Wireless Ethernet Transceiver driver will interact with Ethernet Interface in the same way as the Ethernet Transceiver driver, see sequence diagrams in [8, SWS Ethernet Interface]. Note: There is no Link State Change event in Wireless Ethernet Transceiver driver.

## 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 Wireless Ethernet Transceiver Driver.

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

### 10.1 How to read this chapter

For details refer to the chapter 10.1 “Introduction to configuration specification” in SWS\_BSWGeneral.

### 10.2 Containers and configuration parameters

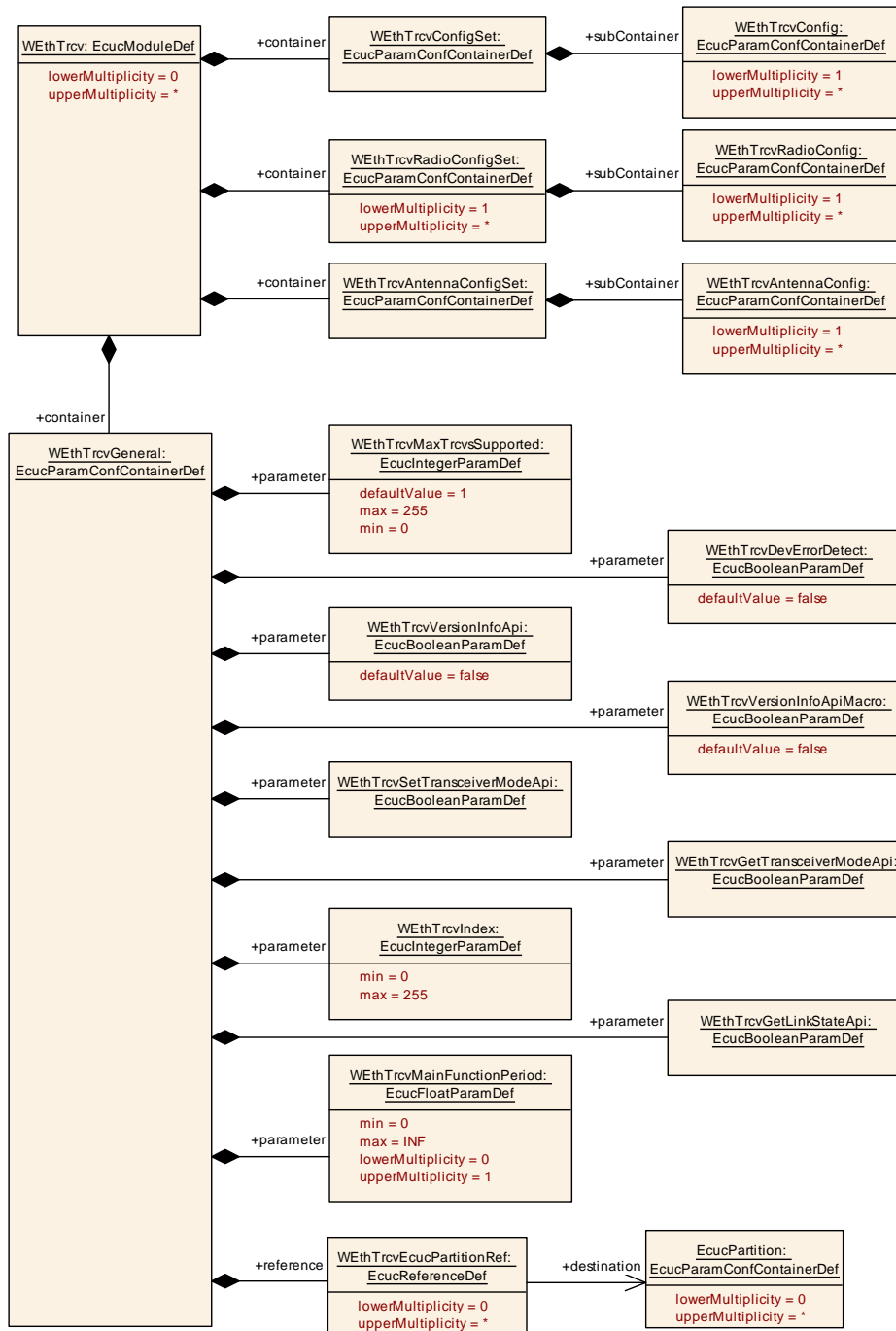
The following chapters summarize all configuration parameters. The detailed meanings of the parameters describe Chapter 7 and Chapter 8.

**[SWS\_WEthTrcv\_00094]** [The Wireless Ethernet Transceiver Driver module shall reject configurations with partition mappings, which are not supported by the implementation.] ()

#### 10.2.1 WEthTrcv

<b>SWS Item</b>	[ECUC_WEthTrcv_10023]
<b>Module Name</b>	WEthTrcv
<b>Description</b>	Configuration of Ethernet Transceiver Driver module
<b>Post-Build Variant Support</b>	true
<b>Supported Config Variants</b>	VARIANT-LINK-TIME, VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">WEthTrcvAntennaConfigSet</a>	1	This container contains the antenna configurations.
<a href="#">WEthTrcvConfigSet</a>	1	This container contains the configuration parameters and sub containers of the AUTOSAR WEthTrcv module.
<a href="#">WEthTrcvGeneral</a>	1	General configuration of Wireless Ethernet Transceiver Driver module
<a href="#">WEthTrcvRadioConfigSet</a>	1..*	This container contains the radio configurations.



**Figure 10.1: WEthTrcv**

## 10.2.2 WEthTrcvConfigSet

<b>SWS Item</b>	[ECUC_WEthTrcv_00016]
<b>Container Name</b>	WEthTrcvConfigSet
<b>Parent Container</b>	<a href="#">WEthTrcv</a>
<b>Description</b>	This container contains the configuration parameters and sub containers of the AUTOSAR WEthTrcv module.
<b>Configuration Parameters</b>	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">WEthTrcvConfig</a>	1..*	Configuration of the individual transceiver

## 10.2.3 WEthTrcvConfig

<b>SWS Item</b>	[ECUC_WEthTrcv_00012]
<b>Container Name</b>	WEthTrcvConfig
<b>Parent Container</b>	<a href="#">WEthTrcvConfigSet</a>
<b>Description</b>	Configuration of the individual transceiver
<b>Configuration Parameters</b>	

<b>SWS Item</b>	[ECUC_WEthTrcv_00015]		
<b>Parameter Name</b>	WEthTrcvBusId		
<b>Parent Container</b>	<a href="#">WEthTrcvConfig</a>		
<b>Description</b>	Specifies the hardware id used for lower level bus interface access (e.g. MII/SPI) to the transceiver's hardware module. For example the MII index if MII would have been used.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default value</b>	-		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	[ECUC_WEthTrcv_00013]		
<b>Parameter Name</b>	WEthTrcvId		
<b>Parent Container</b>	<a href="#">WEthTrcvConfig</a>		
<b>Description</b>	Specifies the instance ID of the configured transceiver.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>	0 .. 255		
<b>Default value</b>	-		
<b>Post-Build Variant Value</b>	false		





<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: ECU withAuto = true		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_00024]</b>		
<b>Parameter Name</b>	WEthTrcvPhysLayerType		
<b>Parent Container</b>	<a href="#">WEthTrcvConfig</a>		
<b>Description</b>	Specifies the physical layer type of the Wireless Ethernet transceiver link.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	TRCV_PHYS_LAYER_TYPE_80211_P	802.11p physical layer	
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10025]</b>		
<b>Parameter Name</b>	WEthTrcvConfigEcucPartitionRef		
<b>Parent Container</b>	<a href="#">WEthTrcvConfig</a>		
<b>Description</b>	Maps one Wireless Ethernet transceiver to zero or one ECUC partitions. The ECUC partition referenced is a subset of the ECUC partitions where the Wireless Ethernet transceiver driver is mapped to.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to EcucPartition		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: ECU		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10022]</b>		
<b>Parameter Name</b>	WEthTrcvCtrlRef		
<b>Parent Container</b>	<a href="#">WEthTrcvConfig</a>		
<b>Description</b>	Specifies a reference to the wireless ethernet controller used for lower layer bus interface access to the transceiver.		



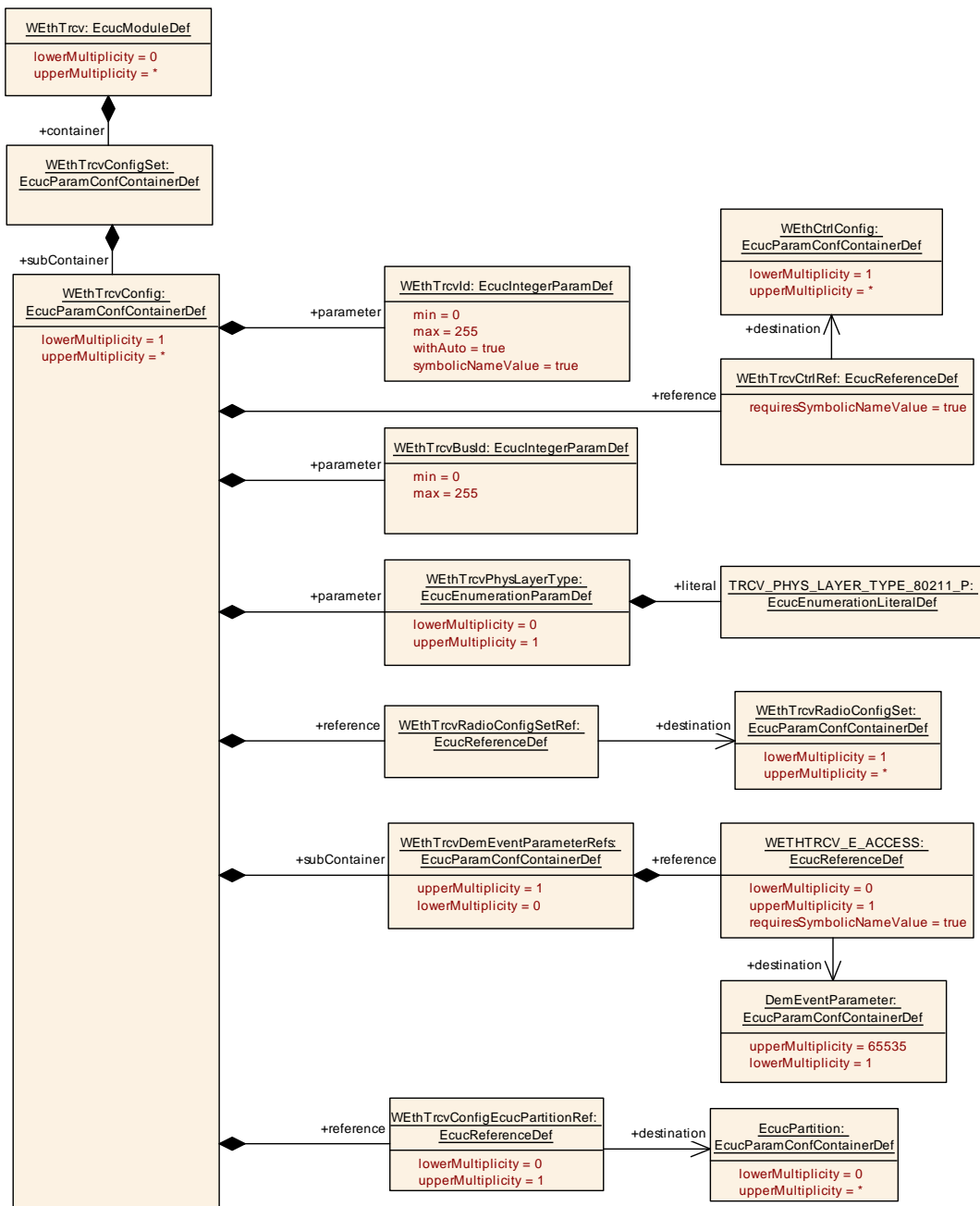


<b>Multiplicity</b>	1		
<b>Type</b>	Symbolic name reference to WEthCtrlConfig		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	[ECUC_WEthTrcv_10001]		
<b>Parameter Name</b>	WEthTrcvRadioConfigSetRef		
<b>Parent Container</b>	<a href="#">WEthTrcvConfig</a>		
<b>Description</b>	Reference to a WEthTrcvRadioConfigSet.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to <a href="#">WEthTrcvRadioConfigSet</a>		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>			

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">WEthTrcvDemEventParameterRefs</a>	0..1	Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.





**Figure 10.2: WethTrcvConfig**

**[SWS\_WEthTrcv\_CONSTR\_00097]** [The ECUC partitions referenced by WETHTrcvConfigEcucPartitionRef shall be a subset of the ECUC partitions referenced by WETHTrcvEcucPartitionRef.] ()

**[SWS\_WEthTrcv\_CONSTR\_00098]** [If WETHTrcvConfigEcucPartitionRef references one or more ECUC partitions, WETHTrcvConfigEcucPartitionRef shall have a multiplicity of one and reference one of these ECUC partitions as well.] ()

## 10.2.4 WEthTrcvDemEventParameterRefs

<b>SWS Item</b>	[ECUC_WEthTrcv_00017]
<b>Container Name</b>	WEthTrcvDemEventParameterRefs
<b>Parent Container</b>	<a href="#">WEthTrcvConfig</a>
<b>Description</b>	Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The Event Id is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.
<b>Configuration Parameters</b>	

<b>SWS Item</b>	[ECUC_WEthTrcv_00018]		
<b>Parameter Name</b>	WETHTRCV_E_ACCESS		
<b>Parent Container</b>	<a href="#">WEthTrcvDemEventParameterRefs</a>		
<b>Description</b>	Reference to the DemEventParameter which shall be issued when the error "Transceiver access failed" has occurred.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Symbolic name reference to DemEventParameter		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>No Included Containers</b>
-------------------------------

## 10.2.5 WEthTrcvRadioConfigSet

<b>SWS Item</b>	[ECUC_WEthTrcv_10002]
<b>Container Name</b>	WEthTrcvRadioConfigSet
<b>Parent Container</b>	<a href="#">WEthTrcv</a>
<b>Description</b>	This container contains the radio configurations.
<b>Configuration Parameters</b>	

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
<a href="#">WEthTrcvRadioConfig</a>	1..*	Configuration of the individual radio (PHY + MAC).

## 10.2.6 WEthTrcvRadioConfig

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10003]</b>		
<b>Container Name</b>	WEthTrcvRadioConfig		
<b>Parent Container</b>	<a href="#">WEthTrcvRadioConfigSet</a>		
<b>Description</b>	Configuration of the individual radio (PHY + MAC).		
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10007]</b>		
<b>Parameter Name</b>	WEthTrcvRadioChannelBandwidth		
<b>Parent Container</b>	<a href="#">WEthTrcvRadioConfig</a>		
<b>Description</b>	Specifies the bandwidth of the physical channel.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	BW_10MHZ	–	
	BW_20MHZ	–	
	BW_40MHZ	–	
	BW_5MHZ	–	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10012]</b>		
<b>Parameter Name</b>	WEthTrcvRadioChannelCsPowerThreshold		
<b>Parent Container</b>	<a href="#">WEthTrcvRadioConfig</a>		
<b>Description</b>	Specifies the threshold for carrier sense (CS) power of the physical channel [dBm].		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	[-100 .. 100]		
<b>Default value</b>	–		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10006]</b>		
<b>Parameter Name</b>	WEthTrcvRadioChannelFreq		
<b>Parent Container</b>	<a href="#">WEthTrcvRadioConfig</a>		
<b>Description</b>	Specifies the frequency of the physical channel [Hz].		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 18446744073709551615		
<b>Default value</b>	–		
<b>Post-Build Variant Value</b>	false		





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

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10011]</b>		
<b>Parameter Name</b>	WEthTrcvRadioChannelMaxTxPower		
<b>Parent Container</b>	<a href="#">WEthTrcvRadioConfig</a>		
<b>Description</b>	Specifies the transmit power of the physical channel [dBm].		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	[-100 .. 100]		
<b>Default value</b>	–		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10010]</b>		
<b>Parameter Name</b>	WEthTrcvRadioChannelTxDataRate		
<b>Parent Container</b>	<a href="#">WEthTrcvRadioConfig</a>		
<b>Description</b>	Specifies the transmit data rate of the physical channel. [bit/s]		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	]0 .. INF[		
<b>Default value</b>	–		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10004]</b>		
<b>Parameter Name</b>	WEthTrcvRadioId		
<b>Parent Container</b>	<a href="#">WEthTrcvRadioConfig</a>		
<b>Description</b>	Specifies the instance ID of the configured radio.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>	0 .. 255		
<b>Default value</b>	–		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	





<b>Scope / Dependency</b>	scope: ECU withAuto = true
---------------------------	-------------------------------

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10005]</b>		
<b>Parameter Name</b>	WEthTrcvRadioMode		
<b>Parent Container</b>	<a href="#">WEthTrcvRadioConfig</a>		
<b>Description</b>	Specifies the mode of the radio within a WEthTrcvRadioConfig. Inside of a WEthTrcvRadioConfigSet different modes for the respective WEthTrcvRadioConfigs are possible. The WEthTrcvRadioConfigSet can be selected at runtime.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	OFF	–	
	RX_ON	–	
	RX_TX_ON	–	
	TX_ON	–	
	USED_FOR_CHANNEL_SWITCHING	–	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10009]</b>		
<b>Parameter Name</b>	WEthTrcvRadioChannelRxAntenna		
<b>Parent Container</b>	<a href="#">WEthTrcvRadioConfig</a>		
<b>Description</b>	Specifies the antenna used for reception of packets of the physical channel.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to <a href="#">WEthTrcvAntennaConfig</a>		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10008]</b>		
<b>Parameter Name</b>	WEthTrcvRadioChannelTxAntenna		
<b>Parent Container</b>	<a href="#">WEthTrcvRadioConfig</a>		
<b>Description</b>	Specifies the antenna used for transmission of packets to the physical channel.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to <a href="#">WEthTrcvAntennaConfig</a>		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">WEthTrcvRadioChannelTxQueue Config</a>	1..*	Configuration of the individual EDCA transmit queue of a channel.



## 10.2.7 WEthTrcvRadioChannelTxQueueConfig

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10013]</b>		
<b>Container Name</b>	WEthTrcvRadioChannelTxQueueConfig		
<b>Parent Container</b>	<a href="#">WEthTrcvRadioConfig</a>		
<b>Description</b>	Configuration of the individual EDCA transmit queue of a channel.		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10015]</b>		
<b>Parameter Name</b>	WEthTrcvRadioChannelTxQueueAifsn		
<b>Parent Container</b>	<a href="#">WEthTrcvRadioChannelTxQueueConfig</a>		
<b>Description</b>	Specifies the arbitration inter frame space number (AIFSN) of the queue.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 15		
<b>Default value</b>	–		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10017]</b>		
<b>Parameter Name</b>	WEthTrcvRadioChannelTxQueueCwMax		
<b>Parent Container</b>	<a href="#">WEthTrcvRadioChannelTxQueueConfig</a>		
<b>Description</b>	Specifies the maximum size of the contention windows (CW) of the queue.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default value</b>	–		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10016]</b>		
<b>Parameter Name</b>	WEthTrcvRadioChannelTxQueueCwMin		
<b>Parent Container</b>	<a href="#">WEthTrcvRadioChannelTxQueueConfig</a>		
<b>Description</b>	Specifies the minimum size of the contention windows (CW) of the queue.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		







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

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10014]</b>		
<b>Parameter Name</b>	WEthTrcvRadioChannelTxQueueId		
<b>Parent Container</b>	<a href="#">WEthTrcvRadioChannelTxQueueConfig</a>		
<b>Description</b>	Specifies the ID (equals priority) of the queue.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default value</b>	-		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10018]</b>		
<b>Parameter Name</b>	WEthTrcvRadioChannelTxQueueTxOpDurationLimit		
<b>Parent Container</b>	<a href="#">WEthTrcvRadioChannelTxQueueConfig</a>		
<b>Description</b>	Specifies the transmit operation duration limit of the queue in [s].		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	[3.2E-5 .. 0.00816]		
<b>Default value</b>	-		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Scope / Dependency</b>	scope: local		

No Included Containers

## 10.2.8 WEthTrcvAntennaConfigSet

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10019]</b>
<b>Container Name</b>	WEthTrcvAntennaConfigSet
<b>Parent Container</b>	<a href="#">WEthTrcv</a>
<b>Description</b>	This container contains the antenna configurations.
<b>Configuration Parameters</b>	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">WEthTrcvAntennaConfig</a>	1..*	Configuration of the individual antenna.

### 10.2.9 WEthTrcvAntennaConfig

SWS Item	[ECUC_WEthTrcv_10020]
Container Name	WEthTrcvAntennaConfig
Parent Container	<a href="#">WEthTrcvAntennaConfigSet</a>
Description	Configuration of the individual antenna.
Configuration Parameters	

SWS Item	[ECUC_WEthTrcv_10021]		
Parameter Name	WEthTrcvAntennald		
Parent Container	<a href="#">WEthTrcvAntennaConfig</a>		
Description	Specifies the instance ID of the configured antenna.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 255		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU withAuto = true		

No Included Containers

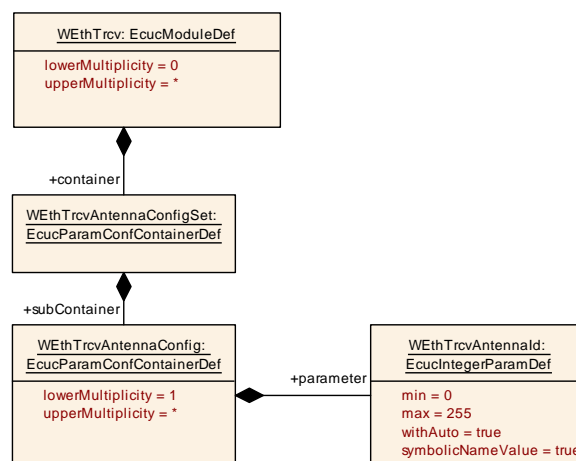


Figure 10.4: WEthTrcvAntennaConfig

### 10.2.10 WEthTrcvGeneral

<b>SWS Item</b>	[ECUC_WEthTrcv_00001]
<b>Container Name</b>	WEthTrcvGeneral
<b>Parent Container</b>	<a href="#">WEthTrcv</a>
<b>Description</b>	General configuration of Wireless Ethernet Transceiver Driver module
<b>Configuration Parameters</b>	

<b>SWS Item</b>	[ECUC_WEthTrcv_00003]		
<b>Parameter Name</b>	WEthTrcvDevErrorDetect		
<b>Parent Container</b>	<a href="#">WEthTrcvGeneral</a>		
<b>Description</b>	Switches the Default Error Tracer (Det) detection and notification ON or OFF. <ul style="list-style-type: none"> <li>• true: detection and notification is enabled.</li> <li>• false: detection and notification is disabled.</li> </ul>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	[ECUC_WEthTrcv_00009]		
<b>Parameter Name</b>	WEthTrcvGetLinkStateApi		
<b>Parent Container</b>	<a href="#">WEthTrcvGeneral</a>		
<b>Description</b>	Enables / Disables WEthTrcv_GetLinkState API		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	–		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	[ECUC_WEthTrcv_00007]		
<b>Parameter Name</b>	WEthTrcvGetTransceiverModeApi		
<b>Parent Container</b>	<a href="#">WEthTrcvGeneral</a>		
<b>Description</b>	Enables / Disables WEthTrcv_GetTransceiverMode API		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	–		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_00020]</b>		
<b>Parameter Name</b>	WEthTrcvIndex		
<b>Parent Container</b>	<a href="#">WEthTrcvGeneral</a>		
<b>Description</b>	Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default value</b>	-		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_00032]</b>		
<b>Parameter Name</b>	WEthTrcvMainFunctionPeriod		
<b>Parent Container</b>	<a href="#">WEthTrcvGeneral</a>		
<b>Description</b>	Specifies the period of main function WEthTrcv_MainFunction in seconds.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	]0 .. INF[		
<b>Default value</b>	-		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_00002]</b>		
<b>Parameter Name</b>	WEthTrcvMaxTrcvSupported		
<b>Parent Container</b>	<a href="#">WEthTrcvGeneral</a>		
<b>Description</b>	-		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default value</b>	1		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_00006]</b>		
<b>Parameter Name</b>	WEthTrcvSetTransceiverModeApi		
<b>Parent Container</b>	<a href="#">WEthTrcvGeneral</a>		
<b>Description</b>	Enables / Disables WEthTrcv_SetTransceiverMode API		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	-		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_00004]</b>		
<b>Parameter Name</b>	WEthTrcvVersionInfoApi		
<b>Parent Container</b>	<a href="#">WEthTrcvGeneral</a>		
<b>Description</b>	Enables / Disables version info API		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_00005]</b>		
<b>Parameter Name</b>	WEthTrcvVersionInfoApiMacro		
<b>Parent Container</b>	<a href="#">WEthTrcvGeneral</a>		
<b>Description</b>	Enables / Disables version info API macro implementation		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>[ECUC_WEthTrcv_10024]</b>		
<b>Parameter Name</b>	WEthTrcvEcucPartitionRef		
<b>Parent Container</b>	<a href="#">WEthTrcvGeneral</a>		
<b>Description</b>	Maps the Wireless Ethernet transceiver driver to zero or multiple ECUC partitions to make the modules API available in this partition. The Wireless Ethernet transceiver driver will operate as an independent instance in each of the partitions.		
<b>Multiplicity</b>	0..*		
<b>Type</b>	Reference to EcucPartition		
<b>Post-Build Variant Multiplicity</b>	true		





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

No Included Containers

**[SWS\_WEthTrcv\_CONSTR\_00095]** [The module will operate as an independent instance in each of the partitions, means the called API will only target the partition it is called in.]()

**[SWS\_WEthTrcv\_CONSTR\_00096]** [WEthTrcvConfig and WEthCtrlConfig of one communication channel shall all reference the same ECUC partition.]()

### 10.3 Published Information

For details refer to the chapter 10.3 “Published Information” in SWS\_BSWGeneral.

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

## **A Change history of AUTOSAR traceable items**

### **A.1 Traceable item history of this document according to AUTOSAR Release R23-11**

#### **A.1.1 Added Specification Items in R23-11**

none

#### **A.1.2 Changed Specification Items in R23-11**

none

#### **A.1.3 Deleted Specification Items in R23-11**

none

#### **A.1.4 Added Constraints in R23-11**

none

#### **A.1.5 Changed Constraints in R23-11**

none

#### **A.1.6 Deleted Constraints in R23-11**

none