

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

<b>Document Status</b>	Final
<b>Part of AUTOSAR Standard</b>	Classic Platform
<b>Part of Standard Release</b>	4.3.1

<b>Document Change History</b>			
<b>Date</b>	<b>Release</b>	<b>Changed by</b>	<b>Change Description</b>
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	<ul style="list-style-type: none"><li>• Initial Release</li></ul>

## Disclaimer

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

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

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

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

## Table of Contents

1	Introduction and functional overview .....	5
2	Acronyms and abbreviations .....	6
3	Related documentation .....	7
3.1	Input documents .....	7
3.2	Related standards and norms .....	8
3.3	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
5.1	File structure.....	10
6	Requirements traceability.....	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.....	12
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 .....	15
7.2.1	Development Errors .....	15
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
8	API specification.....	17
8.1	Imported types.....	17
8.2	Type definitions .....	17
8.2.1	WEthTrcv_ConfigType .....	17
8.2.2	WEthTrcv_SetRadioParamIdType .....	17
8.2.3	WEthTrcv_SetChanRxParamIdType .....	18
8.2.4	WEthTrcv_SetChanTxParamIdType .....	18
8.2.5	WEthTrcv_GetChanRxParamIdType .....	19
8.2.6	WEthTrcv_BandwidthType .....	19
8.2.7	WEthTrcv_TxPwrLvlType .....	19
8.2.8	WEthTrcv_RssiType .....	19
8.2.9	WEthTrcv_RadioModeType.....	19
8.3	Function definitions.....	20
8.3.1	WEthTrcv_Init .....	20

8.3.2	WEthTrcv_SetTransceiverMode .....	20
8.3.3	WEthTrcv_GetTransceiverMode .....	21
8.3.4	WEthTrcv_GetLinkState .....	21
8.3.5	WEthTrcv_SetRadioParams .....	22
8.3.6	WEthTrcv_SetChanRxParams .....	23
8.3.7	WEthTrcv_SetChanTxParams .....	24
8.3.8	WEthTrcv_GetChanRxParams .....	25
8.3.9	WEthTrcv_GetVersionInfo .....	26
8.4	Call-back notifications .....	27
8.5	Interrupt service routines .....	27
8.6	Scheduled functions .....	27
8.6.1	WEthTrcv_MainFunction .....	27
8.7	Expected Interfaces .....	27
8.7.1	Mandatory Interfaces .....	28
8.7.2	Optional Interfaces .....	28
8.7.3	Configurable interfaces .....	28
9	Sequence diagrams .....	29
10	Configuration specification .....	30
10.1	Containers and configuration parameters .....	30
10.1.1	WEthTrcv .....	30
10.1.2	WEthTrcvConfigSet .....	30
10.1.3	WEthTrcvConfig .....	30
10.1.4	WEthTrcvDemEventParameterRefs .....	32
10.1.5	WEthTrcvRadioConfigSet .....	33
10.1.6	WEthTrcvRadioConfig .....	33
10.1.7	WEthTrcvRadioChannelTxQueueConfig .....	36
10.1.8	WEthTrcvAntennaConfigSet .....	37
10.1.9	WEthTrcvAntennaConfig .....	38
10.1.10	WEthTrcvGeneral .....	38
10.2	Published Information .....	41
11	Not applicable requirements .....	42

# 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.

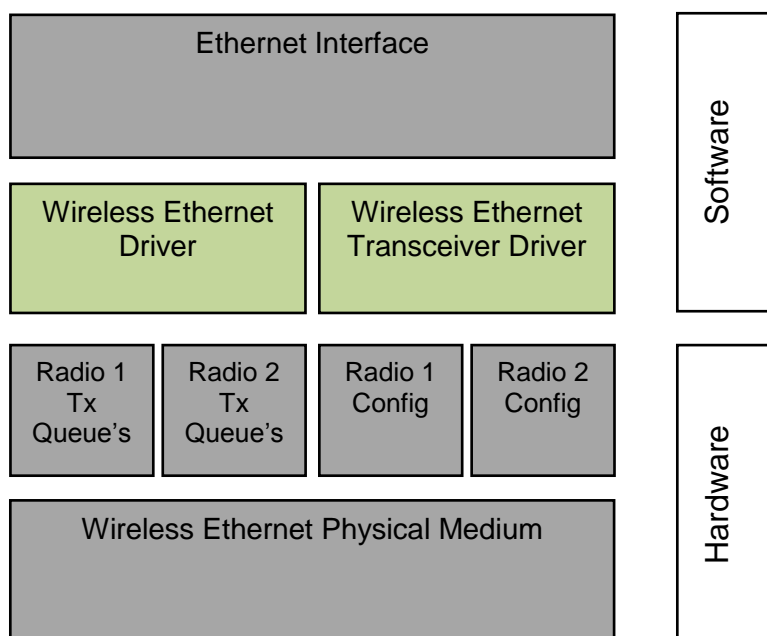


Figure 1.1: Wireless Ethernet module overview

## 2 Acronyms and abbreviations

<b>Abbreviation / Acronym:</b>	<b>Description:</b>
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)

## 3 Related documentation

### 3.1 Input documents

- [1] AUTOSAR Layered Software Architecture  
AUTOSAR\_EXP\_LayeredSoftwareArchitecture.pdf
- [2] AUTOSAR General Requirements on Basic Software Modules  
AUTOSAR\_SRS\_BSWGeneral.pdf
- [3] AUTOSAR General Specification for Basic Software Modules  
AUTOSAR\_SWS\_BSWGeneral.pdf
- [4] Specification of Communication  
AUTOSAR\_SWS\_COM.pdf
- [5] Specification of Ethernet Interface  
AUTOSAR\_SWS\_EthernetInterface.pdf
- [6] Specification of Wireless Ethernet Driver  
AUTOSAR\_SWS\_WirelessEthernetDriver.pdf
- [7] Specification of Ethernet Transceiver Driver  
AUTOSAR\_SWS\_EthernetTransceiverDriver.pdf
- [8] BSW Scheduler Specification  
AUTOSAR\_SWS\_Scheduler.pdf
- [9] Specification of ECU Configuration  
AUTOSAR\_TPS\_ECUConfiguration.pdf
- [10] Specification of Memory Mapping  
AUTOSAR\_SWS\_MemoryMapping.pdf
- [11] Specification of Standard Types  
AUTOSAR\_SWS\_StandardTypes.pdf
- [12] Specification of Default Error Tracer  
AUTOSAR\_SWS\_DefaultErrorTracer.pdf
- [13] Specification of Diagnostics Event Manager  
AUTOSAR\_SWS\_DiagnosticEventManager.pdf
- [14] Requirements on Vehicle-2-X communication  
AUTOSAR\_SRS\_V2XCommunication.pdf

### 3.2 Related standards and norms

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

[16] IEEE 802.11-2012

### 3.3 Related specification

AUTOSAR provides a General Specification on Basic Software (SWS BSW General) [3] which is also valid for Wireless Ethernet Transceiver.

Thus, the specification SWS BSW General [3] 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

No limitations.

### 4.2 Applicability to car domains

The Wireless Ethernet Driver is intended to be used for wireless access of customer hardware (Access Point mode) and 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) for transceiver access via an transceiver dependent interface
- Typically the transceiver hardware is an external device that is accessed by an existing communication driver such as SPI.

### 5.1 File structure

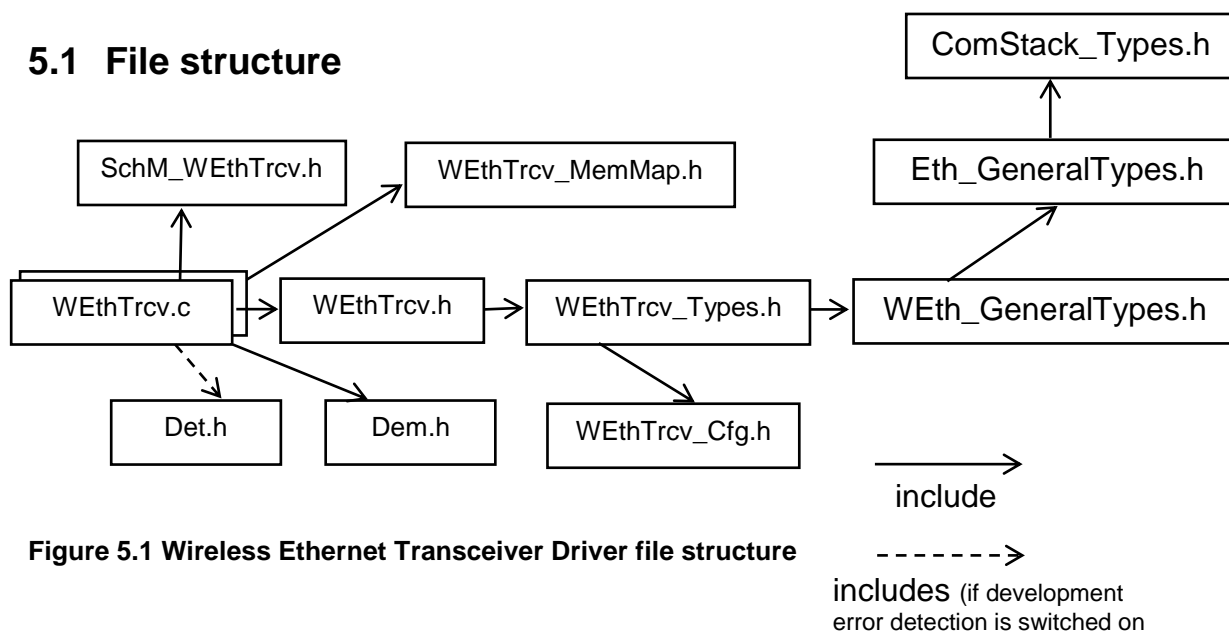


Figure 5.1 Wireless Ethernet Transceiver Driver file structure

#### [SWS\_WEthTrcv\_10072] [

WEthTrcv.h shall include EthTrcv\_GeneralTypes.h for the inclusion of general EthTrcv type declarations. ]()

#### [SWS\_WEthTrcv\_10006] [

WEthTrcv.h shall include WEth\_GeneralTypes.h for the inclusion of general WEth and WEthTrcv type declarations. ]()

#### [SWS\_WEthTrcv\_00096] [

The types specified in SWS\_WirelessEthernetTransceiverDriver shall be declared in WEth\_GeneralTypes.h. ]()

## 6 Requirements traceability

**Note:**

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 SWS Ethernet Driver [7].
- 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

Requirement	Description	Satisfied by
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_00245	The V2X system shall support per-packet transmission power control	SWS_WEthTrcv_20246
SRS_V2X_00451	The V2X system's access layer shall be compliant to the ETSI Harmonized Channel Specifications	SWS_WEthTrcv_10071

## 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 Ethernet Transceiver Driver, [7].

##### [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. ]()

##### [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. ]()

#### 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. ]()

DET API functions are specified in [12].

#### 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`. ]()

**[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`. ]()

**[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`. ]()

**[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`. ]()

#### 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. ]()

**[SWS\_WEthTrcv\_10058]** [

Functions using the type `WEthTrcv_SetRadioParamIdType` shall use a generic list of `uint32` values for the list of corresponding values. ]()

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

Functions using the `WEthTrcv_SetRadioParamIdType` shall use the following type mapping for the corresponding values:

<i>ParamId</i>	<i>ParamValue Type</i>
<code>WETHTRCV_SETRADIOPID_SEL_TRCV_CHCFG</code>	<code>uint8</code>
<code>WETHTRCV_SETRADIOPID_SET_CHCFGID</code>	<code>uint8</code>
<code>WETHTRCV_SETRADIOPID_TOLLINGZONE_INFO</code>	<code>uint8</code>

]()

**[SWS\_WEthTrcv\_10060]** [

Functions using the type `WEthTrcv_SetChanRxParamIdType` shall use a generic list of `uint32` values for the list of corresponding values. ]()

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

Functions using the `WEthTrcv_SetChanRxParamIdType` shall use the following type mapping for the corresponding values:

<b>ParamId</b>	<b>ParamValue Type</b>
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

()

**[SWS\_WEthTrcv\_10062]** [

Functions using the type WEthTrcv\_SetChanTxParamIdType shall use a generic list of uint32 values for the list of corresponding values.

()

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

Functions using the WEthTrcv\_SetChanTxParamIdType shall use the following type mapping for the corresponding values:

<b>ParamId</b>	<b>ParamValue Type</b>
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

()

**[SWS\_WEthTrcv\_10064]** [

Functions using the type WEthTrcv\_GetChanRxParamIdType shall use a generic list of uint32 values for the list of corresponding values.

()

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

Functions using the WEthTrcv\_GetChanRxParamIdType shall use the following type mapping for the corresponding values:

<b>ParamId</b>	<b>ParamValue Type</b>
WETHTRCV_GETCHRXPID_CBR	uint8
WETHTRCV_GETCHRXPID_CIT	uint16

()

### 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 trigger queue transmission via WEth\_TriggerPriorityQueueTransmit and to get Information of the current channel status (CBR). ]()

### 7.1.6 V2X Specific Transceiver Requirements

**[SWS\_WEthTrcv\_10071] [**

The following requirements are only valid for WEth Transceivers used within the V2X Communication Stack [14].

] (SRS\_V2X\_00451)

**[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

### 7.2.1 Development Errors

**[SWS\_WEthTrcv\_00017] [**

<i>Type of error</i>	<i>Related error code</i>	<i>Value [hex]</i>
Invalid transceiver index	WETHTRCV_E_INV_TRCV_ID	0x01
WEthTrcv module was not initialized	WETHTRCV_E_NOT_INITIALIZED	0x02
Invalid pointer in parameter list	WETHTRCV_E_PARAM_POINTER	0x03

]()

### 7.2.2 Runtime Errors

There are no runtime errors.

### 7.2.3 Transient Faults

There are no transient faults.

### 7.2.4 Production Errors

There are no production errors.

### 7.2.5 Extended Production Errors

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

#### [SWS\_WEthTrcv\_00105] [

<b>Error Name:</b>	WETHTRCV_E_ACCESS	
<b>Short Description:</b>	Wireless Ethernet Transceiver Access Failure.	
<b>Long Description:</b>	Monitors the access to the Wireless Ethernet Transceiver.	
<b>Detection Criteria:</b>	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.
<b>Secondary Parameters:</b>	None.	
<b>Time Required:</b>	None.	
<b>Monitor Frequency</b>	None.	

]()



## 8 API specification

### 8.1 Imported types

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

#### [SWS\_WEthTrcv\_00027] [

<i>Module</i>	<i>Imported Type</i>
Dem	Dem_EventIdType
	Dem_EventStatusType
Eth_GeneralTypes	EthTrcv_LinkStateType
	EthTrcv_ModeType
Std_Types	Std_ReturnType
	Std_VersionInfoType
WEth_GeneralTypes	WEthTrcv_ConfigType
	WEthTrcv_GetChanRxParamIdType
	WEthTrcv_SetChanRxParamIdType
	WEthTrcv_SetChanTxParamIdType
	WEthTrcv_SetRadioParamIdType

] ()

### 8.2 Type definitions

#### 8.2.1 WEthTrcv\_ConfigType

##### [SWS\_WEthTrcv\_00098] [

<b>Name:</b>	WEthTrcv_ConfigType
<b>Type:</b>	Structure
<b>Range:</b>	Implementation specific.
<b>Description:</b>	Implementation specific structure of the post build configuration

] ()

#### 8.2.2 WEthTrcv\_SetRadioParamIdType

##### [SWS\_WEthTrcv\_10008] [

<b>Name:</b>	WEthTrcv_SetRadioParamIdType		
<b>Type:</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		

] ()

### 8.2.3 WEthTrcv\_SetChanRxParamIdType

[SWS\_WEthTrcv\_10009] [

<b>Name:</b>	WEthTrcv_SetChanRxParamIdType		
<b>Type:</b>	Enumeration		
<b>Range:</b>	WETHTRCV_SETCHRXPID_BITRATE	0x00	Bitrate
	WETHTRCV_SETCHRXPID_BANDWIDTH	0x01	Bandwidth
	WETHTRCV_SETCHRXPID_FREQ	0x02	Center frequency of a channel
	WETHTRCV_SETCHRXPID_CSPWRTRESH	0x03	Parameter for Rx busy detection
	WETHTRCV_SETCHRXPID_RADIO_MODE	0x04	Param for Rx Radio Mode
	WETHTRCV_SETCHRXPID_ANTENNA	0x05	Rx Antenna Id
<b>Description:</b>	Wireless channel settings for the receive side		

] ()

### 8.2.4 WEthTrcv\_SetChanTxParamIdType

[SWS\_WEthTrcv\_10011] [

<b>Name:</b>	WEthTrcv_SetChanTxParamIdType		
<b>Type:</b>	Enumeration		
<b>Range:</b>	WETHTRCV_SETCHTXPID_BITRATE	0x00	Bitrate
	WETHTRCV_SETCHTXPID_BANDWIDTH	0x01	Bandwidth
	WETHTRCV_SETCHTXPID_TXPOWER	0x02	Transmission power
	WETHTRCV_SETCHTXPID_DCC_CBR	0x03	Param for Channel Busy Ratio for DCC
	WETHTRCV_SETCHTXPID_TXQSEL	0x04	Selection of the transmit queue for that the settings should be set
	WETHTRCV_SETCHTXPID_TXQCFG_AIFSN	0x05	Arbitration inter-frame-spacing number (multiplier with value of 0 to 15)
	WETHTRCV_SETCHTXPID_TXQCFG_CWMIN	0x06	Contention window min
	WETHTRCV_SETCHTXPID_TXQCFG_CWMAX	0x07	Contention window max
	WETHTRCV_SETCHTXPID_TXQCFG_TXOP	0x08	TXOP duration limit [ $\mu$ s] divided by 32
	WETHTRCV_SETCHTXPID_RADIO_MODE	0x09	Param for Tx Radio Mode
	WETHTRCV_SETCHTXPID_ANTENNA	0x0A	Tx Antenna Id
	WETHTRCV_SETCHTXPID_PACKET_INTERVAL	0x0C	Packet interval for transmission interspace
	WETHTRCV_SETCHTXPID_DCC_STATE	0x0D	State of DCC state machine
	<b>Description:</b>	--	

] ()

### 8.2.5 WEthTrcv\_GetChanRxParamIdType

[SWS\_WEthTrcv\_10007] [

<b>Name:</b>	WEthTrcv_GetChanRxParamIdType		
<b>Type:</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		

] ()

### 8.2.6 WEthTrcv\_BandwidthType

[SWS\_WEthTrcv\_10012] [

<b>Name:</b>	WEthTrcv_BandwidthType		
<b>Type:</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		

] ()

### 8.2.7 WEthTrcv\_TxPwrLvIType

[SWS\_WEthTrcv\_10014] [

<b>Name:</b>	WEthTrcv_TxPwrLvIType		
<b>Type:</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		

] ()

### 8.2.8 WEthTrcv\_RssiType

[SWS\_WEthTrcv\_10016] [

<b>Name:</b>	WEthTrcv_RssiType		
<b>Type:</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		

] ()

### 8.2.9 WEthTrcv\_RadioModeType

[SWS\_WEthTrcv\_10018] [

<b>Name:</b>	WEthTrcv_RadioModeType		
<b>Type:</b>	uint32		
<b>Range:</b>	0x00000005..0xFFFFFFFF	--	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		

] ()

## 8.3 Function definitions

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

### 8.3.1 WEthTrcv\_Init

[SWS\_WEthTrcv\_00028] [

<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	

] ()

[SWS\_WEthTrcv\_10022] [

The function shall behave as EthTrcv\_Init in [7], **SWS\_EthTrcv\_00028**. Instead of ETHTRCV\_E\_XXX the corresponding development error WETHTRCV\_E\_XXX shall be used. **SWS\_EthTrcv\_00115** does not apply. ]()

### 8.3.2 WEthTrcv\_SetTransceiverMode

[SWS\_WEthTrcv\_00042] [

<b>Service name:</b>	WEthTrcv_SetTransceiverMode	
<b>Syntax:</b>	<pre>Std_ReturnType WEthTrcv_SetTransceiverMode(     uint8 TrcvId,     EthTrcv_ModeType CtrlMode )</pre>	
<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
	CtrlMode	ETHTRCV_MODE_DOWN: disable the transceiver ETHTRCV_MODE_ACTIVE: enable the transceiver
<b>Parameters (inout):</b>	None	

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

] ()

**[SWS\_WEthTrcv\_10023]** [

The function shall behave as EthTrcv\_SetTransceiverMode in [7], **SWS\_EthTrcv\_00042**. Instead of EthTrcv\_XXX, the corresponding WEthTrcv\_XXX functions shall be used. Instead of ETHTRCV\_E\_YYY the corresponding development error WETHTRCV\_E\_YYY shall be used. Instead of EthTrcvSetTransceiverModeApi, WEthTrcvSetTransceiverModeApi shall be used. **SWS\_EthTrcv\_00117** and **SWS\_EthTrcv\_00118** do not apply. ]()

### 8.3.3 WEthTrcv\_GetTransceiverMode

**[SWS\_WEthTrcv\_00048]** [

<b>Service name:</b>	WEthTrcv_GetTransceiverMode	
<b>Syntax:</b>	Std_ReturnType WEthTrcv_GetTransceiverMode ( uint8 TrcvId, EthTrcv_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	ETHTRCV_MODE_DOWN: the transceiver is disabled ETHTRCV_MODE_ACTIVE: the transceiver is enable
<b>Return value:</b>	Std_ReturnType	E_OK: success E_NOT_OK: transceiver could not be initialized
<b>Description:</b>	Obtains the state of the indexed transceiver	

] ()

**[SWS\_WEthTrcv\_10024]** [

The function shall behave as EthTrcv\_GetTransceiverMode in [7], **SWS\_EthTrcv\_00048**. Instead of EthTrcv\_Init, the WEthTrcv\_Init function shall be used. Instead of ETHTRCV\_E\_XXX the corresponding development error WETHTRCV\_E\_XXX shall be used. Instead of EthTrcvGetTransceiverModeApi, WEthTrcvGetTransceiverModeApi shall be used. ]()

### 8.3.4 WEthTrcv\_GetLinkState

**[SWS\_WEthTrcv\_00061]** [

<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	

] ()

### [SWS\_WEthTrcv\_10073] [

The function shall behave as EthTrcv\_GetLinkState in [7], **SWS\_EthTrcv\_00061**. Instead of EthTrcv\_Init, the WEthTrcv\_Init function shall be used. Instead of ETHTRCV\_E\_XXX the corresponding development error WETHTRCV\_E\_XXX shall be used. Instead of EthTrcvGetLinkStateApi, WEthTrcvGetLinkStateApi shall be used. ]()

## 8.3.5 WEthTrcv\_SetRadioParams

### [SWS\_WEthTrcv\_10025] [

<b>Service name:</b>	WEthTrcv_SetRadioParams	
<b>Syntax:</b>	Std_ReturnType WEthTrcv_SetRadioParams ( uint8 TrcvId, const WEthTrcv_SetRadioParamIdType* ParamIds, const uint32* ParamValue, uint8 NumParams )	
<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, ...).	

] ()

### [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\_NOT\_INITIALIZED. ]()

**[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]** [

<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.	

] ()

**[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\_NOT\_INITIALIZED. ]()

**[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]** [

<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</b>	None	



<b>(inout):</b>	
<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.

] ()

**[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\_NOT\_INITIALIZED. ]()

**[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]** [

<b>Service name:</b>	WEthTrcv_GetChanRxParams
<b>Syntax:</b>	Std_ReturnType WEthTrcv_GetChanRxParams ( uint8* TrcvId, uint8 RadioId, const WEthTrcv_GetChanRxParamIdType* ParamIds, uint32* ParamValues, uint8 NumParams )
<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
	RadiId	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).	

] ()

**[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\_NOT\_INITIALIZED. ]()

**[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 RadiId 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]** [

<b>Service name:</b>	WEthTrcv_GetVersionInfo
<b>Syntax:</b>	void WEthTrcv_GetVersionInfo( Std_VersionInfoType* VersionInfoPtr )

<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	

] ()

#### [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 Call-back notifications

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

## 8.5 Interrupt service routines

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

## 8.6 Scheduled functions

### 8.6.1 WEthTrcv\_MainFunction

#### [SWS\_WEthTrcv\_00106] [

<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.	

] ()

## 8.7 Expected Interfaces

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

### 8.7.1 Mandatory Interfaces

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

#### [SWS\_WEthTrcv\_00085] [

<i>API function</i>	<i>Description</i>
Dem_SetEventStatus	Called by SW-Cs or BSW modules to report monitor status information to the Dem. BSW modules calling Dem_SetEventStatus can safely ignore the return value.
EthIf_TrcvModeIndication	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	Invokes the SchM_Enter function to enter a module local exclusive area.
SchM_Exit_WEthTrcv	Invokes the SchM_Exit function to exit an exclusive area.
WEth_ReadTrcvRegs	Reads a transceiver register
WEth_WriteTrcvRegs	Configures a transceivers registers or triggers a function offered by the receiver

] ()

### 8.7.2 Optional Interfaces

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

#### [SWS\_WEthTrcv\_00120] [

<i>API function</i>	<i>Description</i>
Det_ReportError	Service to report development errors.

] ()

### 8.7.3 Configurable interfaces

The Wireless Ethernet Transceiver Driver does not use configurable interfaces.

## 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 [5]. Note: there is no Link State Change event in Wireless Ethernet Transceiver driver.

## 10 Configuration specification

Chapter 10.1 specifies the structure (containers) and the parameters of the module Wireless Ethernet Transceiver Driver.

Chapter 10.2 specifies additionally published information of the module Wireless Ethernet Transceiver Driver.

### 10.1 Containers and configuration parameters

The following chapters summarize all configuration parameters.

#### 10.1.1 WEthTrcv

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

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

#### 10.1.2 WEthTrcvConfigSet

<b>SWS Item</b>	<b>ECUC_WEthTrcv_00016 :</b>
<b>Container Name</b>	WEthTrcvConfigSet
<b>Description</b>	This container contains the configuration parameters and sub containers of the AUTOSAR WEthTrcv module.
<b>Configuration Parameters</b>	

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

#### 10.1.3 WEthTrcvConfig

<b>SWS Item</b>	<b>ECUC_WEthTrcv_00012 :</b>
<b>Container Name</b>	WEthTrcvConfig
<b>Description</b>	Configuration of the individual transceiver

**Configuration Parameters**

<b>SWS Item</b>	<b>ECUC_WEthTrcv_00015 :</b>		
<b>Name</b>	WEthTrcvBusId		
<b>Parent Container</b>	WEthTrcvConfig		
<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>	<b>ECUC_WEthTrcv_00013 :</b>		
<b>Name</b>	WEthTrcvId		
<b>Parent Container</b>	WEthTrcvConfig		
<b>Description</b>	Specifies the instance ID of the configured transceiver.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>	0 .. 255		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: ECU		

<b>SWS Item</b>	<b>ECUC_WEthTrcv_00024 :</b>		
<b>Name</b>	WEthTrcvPhysLayerType		
<b>Parent Container</b>	WEthTrcvConfig		
<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_10022 :</b>		
<b>Name</b>	WEthTrcvCtrlRef		
<b>Parent Container</b>	WEthTrcvConfig		
<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>	<b>ECUC_WEthTrcv_10001 :</b>		
<b>Name</b>	WEthTrcvRadioConfigSetRef		
<b>Parent Container</b>	WEthTrcvConfig		
<b>Description</b>	Reference to a WEthTrcvRadioConfigSet.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to [ WEthTrcvRadioConfigSet ]		
<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>			

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

#### 10.1.4 WEthTrcvDemEventParameterRefs

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

<b>SWS Item</b>	<b>ECUC_WEthTrcv_00018 :</b>
<b>Name</b>	WETHTRCV_E_ACCESS
<b>Parent Container</b>	WEthTrcvDemEventParameterRefs
<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		

**No Included Containers**

### 10.1.5 WEthTrcvRadioConfigSet

<b>SWS Item</b>	<b>ECUC_WEthTrcv_10002 :</b>
<b>Container Name</b>	WEthTrcvRadioConfigSet
<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>
WEthTrcvRadioConfig	1..*	Configuration of the individual radio (PHY + MAC).

### 10.1.6 WEthTrcvRadioConfig

<b>SWS Item</b>	<b>ECUC_WEthTrcv_10003 :</b>
<b>Container Name</b>	WEthTrcvRadioConfig
<b>Description</b>	Configuration of the individual radio (PHY + MAC).
<b>Configuration Parameters</b>	

<b>SWS Item</b>	<b>ECUC_WEthTrcv_10007 :</b>		
<b>Name</b>	WEthTrcvRadioChannelBandwidth		
<b>Parent Container</b>	WEthTrcvRadioConfig		
<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>Name</b>	WEthTrcvRadioChannelCsPowerThreshold		
<b>Parent Container</b>	WEthTrcvRadioConfig		

<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>Name</b>	WEthTrcvRadioChannelFreq		
<b>Parent Container</b>	WEthTrcvRadioConfig		
<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>Name</b>	WEthTrcvRadioChannelMaxTxPower		
<b>Parent Container</b>	WEthTrcvRadioConfig		
<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>Name</b>	WEthTrcvRadioChannelTxDataRate		
<b>Parent Container</b>	WEthTrcvRadioConfig		
<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>Name</b>	WEthTrcvRadioId		
<b>Parent Container</b>	WEthTrcvRadioConfig		
<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		

<b>SWS Item</b>	<b>ECUC_WEthTrcv_10005 :</b>		
<b>Name</b>	WEthTrcvRadioMode		
<b>Parent Container</b>	WEthTrcvRadioConfig		
<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>Name</b>	WEthTrcvRadioChannelRxAntenna		
<b>Parent Container</b>	WEthTrcvRadioConfig		
<b>Description</b>	Specifies the antenna used for reception of packets of the physical channel.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to [ WEthTrcvAntennaConfig ]		
<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>Name</b>	WEthTrcvRadioChannelTxAntenna		
<b>Parent Container</b>	WEthTrcvRadioConfig		
<b>Description</b>	Specifies the antenna used for transmission of packets to the physical channel.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to [ WEthTrcvAntennaConfig ]		
<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>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
WEthTrcvRadioChannelTxQueueConfig	1..*	Configuration of the individual EDCA transmit queue of a channel.

### 10.1.7 WEthTrcvRadioChannelTxQueueConfig

<b>SWS Item</b>	<b>ECUC_WEthTrcv_10013 :</b>		
<b>Container Name</b>	WEthTrcvRadioChannelTxQueueConfig		
<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>Name</b>	WEthTrcvRadioChannelTxQueueAifsn		
<b>Parent Container</b>	WEthTrcvRadioChannelTxQueueConfig		
<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>Name</b>	WEthTrcvRadioChannelTxQueueCwMax		
<b>Parent Container</b>	WEthTrcvRadioChannelTxQueueConfig		
<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>Name</b>	WEthTrcvRadioChannelTxQueueCwMin		
<b>Parent Container</b>	WEthTrcvRadioChannelTxQueueConfig		

<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>Name</b>	WEthTrcvRadioChannelTxQueueId		
<b>Parent Container</b>	WEthTrcvRadioChannelTxQueueConfig		
<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>Name</b>	WEthTrcvRadioChannelTxQueueTxOpDurationLimit		
<b>Parent Container</b>	WEthTrcvRadioChannelTxQueueConfig		
<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**

### 10.1.8 WEthTrcvAntennaConfigSet

<b>SWS Item</b>	<b>ECUC_WEthTrcv_10019 :</b>		
<b>Container Name</b>	WEthTrcvAntennaConfigSet		
<b>Description</b>	This container contains the antenna configurations.		
<b>Configuration Parameters</b>			

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

### 10.1.9 WEthTrcvAntennaConfig

<b>SWS Item</b>	<b>ECUC_WEthTrcv_10020 :</b>		
<b>Container Name</b>	WEthTrcvAntennaConfig		
<b>Description</b>	Configuration of the individual antenna.		
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>ECUC_WEthTrcv_10021 :</b>		
<b>Name</b>	WEthTrcvAntennaId		
<b>Parent Container</b>	WEthTrcvAntennaConfig		
<b>Description</b>	Specifies the instance ID of the configured antenna.		
<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		

**No Included Containers**

### 10.1.10 WEthTrcvGeneral

<b>SWS Item</b>	<b>ECUC_WEthTrcv_00001 :</b>		
<b>Container Name</b>	WEthTrcvGeneral		
<b>Description</b>	General configuration of Wireless Ethernet Transceiver Driver module		
<b>Configuration Parameters</b>			

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

<b>SWS Item</b>	<b>ECUC_WEthTrcv_00009 :</b>		
<b>Name</b>	WEthTrcvGetLinkStateApi		

<b>Parent Container</b>	WEthTrcvGeneral		
<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>	<b>ECUC_WEthTrcv_00007 :</b>		
<b>Name</b>	WEthTrcvGetTransceiverModeApi		
<b>Parent Container</b>	WEthTrcvGeneral		
<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>Name</b>	WEthTrcvIndex		
<b>Parent Container</b>	WEthTrcvGeneral		
<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>Name</b>	WEthTrcvMainFunctionPeriod		
<b>Parent Container</b>	WEthTrcvGeneral		
<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>Name</b>	WEthTrcvMaxTrcvSupported		
<b>Parent Container</b>	WEthTrcvGeneral		
<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>Name</b>	WEthTrcvSetTransceiverModeApi		
<b>Parent Container</b>	WEthTrcvGeneral		
<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>Name</b>	WEthTrcvVersionInfoApi		
<b>Parent Container</b>	WEthTrcvGeneral		
<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>Name</b>	WEthTrcvVersionInfoApiMacro		
<b>Parent Container</b>	WEthTrcvGeneral		
<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		



*No Included Containers*

## 10.2 Published Information

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

## 11 Not applicable requirements