

Document Title	SWS_EthernetTransceiverDriver: Complete Change Documentation 4.3.0 - 4.3.1
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	695

Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.3.1

Table of Contents

1	SWS_EthernetTransceiverDriver	3
1.1	Specification Item ECUC_EthTrcv_00001	3
1.2	Specification Item ECUC_EthTrcv_00012	7
1.3	Specification Item ECUC_EthTrcv_00043	10
1.4	Specification Item ECUC_EthTrcv_00044	13
1.5	Specification Item ECUC_EthTrcv_00045	17
1.6	Specification Item ECUC_EthTrcv_00046	20
1.7	Specification Item ECUC_EthTrcv_00047	24
1.8	Specification Item ECUC_EthTrcv_00048	28
1.9	Specification Item ECUC_EthTrcv_00049	32
1.10	Specification Item SWS_EthTrcv_00017	36
1.11	Specification Item SWS_EthTrcv_00027	37
1.12	Specification Item SWS_EthTrcv_00044	39
1.13	Specification Item SWS_EthTrcv_00050	40
1.14	Specification Item SWS_EthTrcv_00057	41
1.15	Specification Item SWS_EthTrcv_00063	43
1.16	Specification Item SWS_EthTrcv_00070	44
1.17	Specification Item SWS_EthTrcv_00077	45
1.18	Specification Item SWS_EthTrcv_00085	47
1.19	Specification Item SWS_EthTrcv_00086	54
1.20	Specification Item SWS_EthTrcv_00122	56
1.21	Specification Item SWS_EthTrcv_00129	57
1.22	Specification Item SWS_EthTrcv_00131	59
1.23	Specification Item SWS_EthTrcv_00137	60
1.24	Specification Item SWS_EthTrcv_00142	62
1.25	Specification Item SWS_EthTrcv_91001	63

1 SWS_EthernetTransceiverDriver

1.1 Specification Item ECUC_EthTrcv_00001

Trace References:

none

Content:

Container Name	EthTrcvGeneralEthTrcvGeneral
Description	General configuration of Ethernet Transceiver Driver module
Configuration Parameters	

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
EthTrcvDevErrorDetect	ECUC_EthTrcv_00003
EthTrcvGetBaudRateApi	ECUC_EthTrcv_00010
EthTrcvGetCableDiagnosticsResultApi	ECUC_EthTrcv_00049
EthTrcvGetDuplexModeApi	ECUC_EthTrcv_00011
EthTrcvGetLinkStateApi	ECUC_EthTrcv_00009
EthTrcvGetPhyIdentifierApi	ECUC_EthTrcv_00046
EthTrcvGetPhySignalQualityApi	ECUC_EthTrcv_00045
EthTrcvGetTransceiverModeApi	ECUC_EthTrcv_00007
EthTrcvGetTransceiverWakeupModeApi	ECUC_EthTrcv_00031
EthTrcvIndex	ECUC_EthTrcv_00020
EthTrcvMainFunctionPeriod	ECUC_EthTrcv_00032
EthTrcvMaxTrcvsSupported	ECUC_EthTrcv_00002
EthTrcvSetPhyTestModeApi	ECUC_EthTrcv_00047
EthTrcvSetPhyTxModeApi	ECUC_EthTrcv_00048
EthTrcvSetTransceiverModeApi	ECUC_EthTrcv_00006
EthTrcvStartAutoNegotiationApi	ECUC_EthTrcv_00008
EthTrcvVersionInfoApi	ECUC_EthTrcv_00004
EthTrcvVersionInfoApiMacro	ECUC_EthTrcv_00005
EthTrcvWakeUpSupport	ECUC_EthTrcv_00030

Included containers:

No Included Containers

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77628: [EthSwt] Behaviour of certain APIs for ports without EthSwtPortTrcvRef (part I)

Problem description:

The following APIs have partly a description regarding the handling of ports without EthSwtPortTrcvRef:

EthSwt_GetPortSignalQuality
EthSwt_GetPortIdentifier
EthSwt_SetPortTestMode
EthTrcv_SetPhyTestMode
EthSwt_SetPortTxMode
EthSwt_GetPortCableDiagnosticsResult

The description should be adjusted and harmonized with related requirements.
–Last change on issue 77628 comment 2–

Agreed solution:

=== EthSwt ===

~[SWS_EthSwt_91014]EthSwt_GetPortSignalQuality

Description:

The function retrieves the signal quality of the link of the indexed Ethernet switch port.

~[SWS_EthSwt_00293]

The function EthSwt_GetPortSignalQuality() shall obtain the signal quality by calling the function EthTrcv_GetPhySignalQuality() of the referenced Ethernet Transceiver Driver. If the current signal quality is not available, the signal quality shall be set to 0xFF.

-[SWS_EthSwt_00298]as # 77349 introduces a general req

~[SWS_EthSwt_91015]EthSwt_GetPortIdentifier

Return value

E_NOT_OK: organizationally unique identifier of the Ethernet transceiver could not be obtained (i.e. OUI is not available)

Description:

This function retrieves the OUI (24 bit) of the indexed Ethernet switch port.

~[SWS_EthSwt_00299]

The function EthSwt_GetPortIdentifier() shall return the value of the organizationally unique identifier (OUI 24 bit) of the indexed Ethernet switch port that is connected to the indexed Ethernet switch. It shall set the 8 most significant bits of the OUI

to 0xFFxxxxxx. If the Ethernet switch port references an Ethernet transceiver, the function shall obtain the OUI by calling the function `EthTrcv_GetPhyIdentifier()` and set the 8 most significant bits of the OUI to 0x00xxxxxx.

SWS_EthSwt_00000] If neither the Ethernet switch port nor the Ethernet Transceiver Driver can provide an OUI the function `EthSwt_GetPortIdentifier()` shall return `E_NOT_OK`.

-[SWS_EthSwt_00304] as # 77349 introduces a general req

~[SWS_EthSwt_91029] `EthSwt_SetPortTestMode`

Description:

Activates a given test mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00328]

The function `EthSwt_SetPortTestMode` shall forward the call with the given test mode by calling the function `EthTrcv_SetPhyTestMode()` of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00333] as # 77349 introduces a general req

~[SWS_EthSwt_91023] `EthSwt_SetPortLoopbackMode`

Description:

Activates a given test loop-back mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00334]

The function `EthSwt_SetPortLoopbackMode()` shall forward the call with the given loop-back mode by calling the function `EthTrcv_SetPhyLoopbackMode()` of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00339] as # 77349 introduces a general req

~[SWS_EthSwt_91024] `EthSwt_SetPortTxMode`

Description:

Activates a given transmission mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00340]

The function `EthSwt_SetPortTxMode()` shall forward the call with the given transmission mode by calling the function `EthTrcv_SetPhyTxMode()` of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00345] as # 77349 introduces a general req

~[SWS_EthSwt_91025] `EthSwt_GetPortCableDiagnosticsResult`

Description:

Retrieves the cable diagnostics result of the indexed Ethernet switch port respec-

tively the referenced Ethernet Transceiver Driver.

~[SWS_EthSwt_00346]

The function EthSwt_GetPortCableDiagnosticsResult() shall obtain the cable diagnostics result by calling the function EthTrcv_GetCableDiagnosticsResult() of the referenced Ethernet Transceiver Driver. If no Ethernet transceiver is referenced by the Ethernet switch port and development error detection is not enabled, the cable diagnostic result shall be set to ETHTRCV_CABLEDIAG_OK.

-[SWS_EthSwt_00351] as # 77349 introduces a general req

=== EthTrcv ===

add certain parameter to enable/disable API functions:

+ SWS item ECUC_EthTrcv_xxxx1

Name EthTrcvGetPhySignalQualityApi

Description Enables / Disables EthTrcv_GetPhySignalQuality API

+ SWS item ECUC_EthTrcv_xxxx2

Name EthTrcvGetPhyIdentifierApi

Description Enables / Disables EthTrcv_GetPhyIdentifier API

+ SWS item ECUC_EthTrcv_xxxx3

Name EthTrcvSetPhyTestModeApi

Description Enables / Disables EthTrcv_SetPhyTestMode API

+ SWS item ECUC_EthTrcv_xxxx4

Name EthTrcvSetPhyTxModeApi

Description Enables / Disables EthTrcv_SetPhyTxMode API

+ SWS item ECUC_EthTrcv_xxxx5

Name EthTrcvGetCableDiagnosticsResultApi

Description Enables / Disables EthTrcv_GetCableDiagnosticsResult API

add the following specification to the configuration parameter above:

Multiplicity 1

Type EcucBooleanParamDef

Default value –

Post-Build Variant Value false

Value Configuration

Class Pre-compile time X

All Variants Link time –

Post-build time –

Scope / Dependency scope: local

–Last change on issue 77628 comment 31–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.2 Specification Item ECUC_EthTrcv_00012

Trace References:

none

Content:

Container Name	EthTrcvConfigEthTrcvConfig
Description	Configuration of the individual transceiver
Configuration Parameters	

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
EthTrcvAutoNegotiationEnabled	ECUC_EthTrcv_00021
EthTrcvConnNeg	ECUC_EthTrcv_00025
EthTrcvCtrlIdx	ECUC_EthTrcv_00014
EthTrcvDuplexMode	ECUC_EthTrcv_00023
EthTrcvIdx	ECUC_EthTrcv_00013
EthTrcvPhysLayerType	ECUC_EthTrcv_00024
EthTrcvPortMacLayerSpeed	ECUC_EthTrcv_00044
EthTrcvPortMacLayerSubType	ECUC_EthTrcv_00043
EthTrcvPortMacLayerType	ECUC_EthTrcv_00035
EthTrcvSpeed	ECUC_EthTrcv_00022
EthTrcvWakeUpCallout	ECUC_EthTrcv_00028
EthTrcvLcuChannelRef	ECUC_EthTrcv_00026

Included containers:

Included Containers		
Container Name	Multiplicity	Scope / Dependency

Included Containers		
Container Name	Multiplicity	Scope / Dependency
EthTrcvDemEventParameterRefs	0..1	Container for the references to Dem EventParameter elements which shall be invoked using the API Dem_SetEvent Status in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's Dem EventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.
EthTrcvMgmtInterface	0..1	The choice container allow to configure either the EthTrcv is accessed by a MII interface or Switch interface.
EthTrcvWakeupMap	0..7	Container for the mapping of wake up reasons to wake up sources. At least one container is needed if EthTrcvWakeUpSupport is not ETHTRCV_WAKEUP_NOT_SUPPORTED.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76343: Configuration parameters for Ethernet MAC layer types is incomplete

Problem description:

With https://www.autosar.org/bugzilla/show_bug.cgi?id=73074 the configuration of the MAC layer type was separated from the physical layer type. The configuration parameter EthSwtPortMacLayerType allows for the configuration of "families" of MAC layer types, namely xMII, xGMII, and xxGMII.

The concrete MAC layer type (e.g., SGMII vs. RGMII) however cannot be configure by this parameter. - Available switches (e.g., Broadcom's "Leo") however require the configuration of a *concrete* MAC layer type (and not only the configuration of a "family" of MAC layer types) for the individual port.

The same holds probably true for the Ethernet Driver and the Ethernet Transceiver driver.

One possibility to solve this is to introduce an additional config parameter named something like EthSwtPortMacLayerSubType with the following enum values "standard", "reduced", "reversed", "serial", "universal serial". - Thus for example the combination of EthSwtPortMacLayerType=xGMII and EthSwtPortMacLayerSubType="reversed" would yield RvGMII.

Agreed solution:

EthSwt:

add config parameter to container EthSwtPort named EthSwtPortMacLayerSub-

Type with the following enum values "STANDARD: standard media-independent interface", "REDUCED: Reduced media-independent interface", "REVERSED: reversed media-independent interface (to provide direct connection between two Ethernet MACs)", "SERIAL: low-power and low pin-count serial 8b/10b-coded media-independent interface", "UNIVERSAL SERIAL": Universal low-power and low pin-count serial 8b/10b-coded media-independent interface".

add config parameter to container EthSwtPort named EthSwtPortMacLayerSpeed with the following enum values "ETH_MAC_LAYER_SPEED_10M", "ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G", "ETH_MAC_LAYER_SPEED_10G".

All parameters shall have the same Multiplicity, Variant and config class as EthSwtPortMacLayerType.

add description for EthSwt_PortMacLayerSubTypes:
Defines the MAC layer subtype of a switch port

add description for EthSwt_PortMacLayerSpeed:
Defines the baud rate of the MAC layer

Eth:

add config parameter to container EthCtrlConfig named EthCtrlMacLayerSubType with the following enum values "STANDARD", "REDUCED", "REVERSED", "SERIAL", "UNIVERSAL SERIAL". - Thus for example the combination of EthCtrlMacLayerType=xGMII and EthCtrlMacLayerSubType="REVERSED" would yield RvGMII.

add config parameter to container EthCtrlConfig named EthCtrlMacLayerSpeed with the following enum values "ETH_MAC_LAYER_SPEED_10M", "ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G", "ETH_MAC_LAYER_SPEED_10G".

add description for Eth_PortMacLayerSubTypes:
Defines the MAC layer subtype of a switch port

add description for Eth_PortMacLayerSpeed:
Defines the baud rate of the MAC layer

EthTrcv:

add config parameter to container EthTrcvCtrlConfig named EthTrcvPortMacLayerSubType with the following enum values "STANDARD", "REDUCED", "REVERSED", "SERIAL", "UNIVERSAL SERIAL". - Thus for example the combination of EthTrcvPortMacLayerType=xGMII and EthTrcvPortMacLayerSubType="REVERSED" would

yield RvGMII.

add config parameter to container EthTrcvCtrlConfig named EthTrcvPortMacLayerSpeed with the following enum values "ETH_MAC_LAYER_SPEED_10M", "ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G", "ETH_MAC_LAYER_SPEED_10G".

add description for EthTrcv_PortMacLayerSubTypes:
Defines the MAC layer subtype of a switch port

add description for EthTrcv_PortMacLayerSpeed:
Defines the baud rate of the MAC layer

All parameters shall have the same Multiplicity, Variant and config class as EthSwtPortMacLayerType.

No upstream Mapping as this decision is made on ECU-Configuration.
–Last change on issue 76343 comment 30–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.3 Specification Item ECUC_EthTrcv_00043

Trace References:

none

Content:

Name	EthTrcvPortMacLayerSubTypeEthTrcvConfig.EthTrcvPortMacLayerSubType
Description	Defines the MAC layer subtype of a switch port
Multiplicity	0..1
Type	EcucEnumerationParamDef

Range	REDUCEDEthTrcvConfig.EthTrcvPortMacLayerSubType.REDUCED	—	
	REVERSEDEthTrcvConfig.EthTrcvPortMacLayerSubType.REVERSED	—	
	SERIALEthTrcvConfig.EthTrcvPortMacLayerSubType.SERIAL	—	
	STANDARDEthTrcvConfig.EthTrcvPortMacLayerSubType.STANDARD	—	
	UNIVERSAL_SERIALEthTrcvConfig.EthTrcvPortMacLayerSubType.UNIVERSAL_SERIAL	—	
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	—	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	—	
Scope / Dependency	scope: ECU		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76343: Configuration parameters for Ethernet MAC layer types is incomplete

Problem description:

With https://www.autosar.org/bugzilla/show_bug.cgi?id=73074 the configuration of the MAC layer type was separated from the physical layer type. The configuration parameter EthSwtPortMacLayerType allows for the configuration of "families" of MAC layer types, namely xMII, xGMII, and xxGMII.

The concrete MAC layer type (e.g., SGMII vs. RGMII) however cannot be configure by this parameter. - Available switches (e.g., Broadcom's "Leo") however require the configuration of a *concrete* MAC layer type (and not only the configu-

ration of a "family" of MAC layer types) for the individual port.

The same holds probably true for the Ethernet Driver and the Ethernet Transceiver driver.

One possibility to solve this is to introduce an additional config parameter named something like EthSwtPortMacLayerSubType with the following enum values "standard", "reduced", "reversed", "serial", "universal serial". - Thus for example the combination of EthSwtPortMacLayerType=xGMII and EthSwtPortMacLayerSubType="reversed" would yield RvGMII.

Agreed solution:

EthSwt:

add config parameter to container EthSwtPort named EthSwtPortMacLayerSubType with the following enum values "STANDARD: standard media-independent interface", "REDUCED: Reduced media-independent interface", "REVERSED: reversed media-independent interface (to provide direct connection between two Ethernet MACs)", "SERIAL: low-power and low pin-count serial 8b/10b-coded media-independent interface", "UNIVERSAL SERIAL: Universal low-power and low pin-count serial 8b/10b-coded media-independent interface".

add config parameter to container EthSwtPort named EthSwtPortMacLayerSpeed with the following enum values "ETH_MAC_LAYER_SPEED_10M", "ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G", "ETH_MAC_LAYER_SPEED_10G".

All parameters shall have the same Multiplicity, Variant and config class as EthSwtPortMacLayerType.

add description for EthSwt_PortMacLayerSubTypes:

Defines the MAC layer subtype of a switch port

add description for EthSwt_PortMacLayerSpeed:

Defines the baud rate of the MAC layer

Eth:

add config parameter to container EthCtrlConfig named EthCtrlMacLayerSubType with the following enum values "STANDARD", "REDUCED", "REVERSED", "SERIAL", "UNIVERSAL SERIAL". - Thus for example the combination of EthCtrlMacLayerType=xGMII and EthCtrlMacLayerSubType="REVERSED" would yield RvGMII.

add config parameter to container EthCtrlConfig named EthCtrlMacLayerSpeed with the following enum values "ETH_MAC_LAYER_SPEED_10M",

"ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G",
"ETH_MAC_LAYER_SPEED_10G".

add description for Eth_PortMacLayerSubTypes:
Defines the MAC layer subtype of a switch port

add description for Eth_PortMacLayerSpeed:
Defines the baud rate of the MAC layer

EthTrcv:

add config parameter to container EthTrcvCtrlConfig named EthTrcvPortMacLayerSubType with the following enum values "STANDARD", "REDUCED", "REVERSED", "SERIAL", "UNIVERSAL SERIAL". - Thus for example the combination of EthTrcvPortMacLayerType=xGMII and EthTrcvPortMacLayerSubType="REVERSED" would yield RvGMII.

add config parameter to container EthTrcvCtrlConfig named EthTrcvPortMacLayerSpeed with the following enum values "ETH_MAC_LAYER_SPEED_10M", "ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G", "ETH_MAC_LAYER_SPEED_10G".

add description for EthTrcv_PortMacLayerSubTypes:
Defines the MAC layer subtype of a switch port

add description for EthTrcv_PortMacLayerSpeed:
Defines the baud rate of the MAC layer

All parameters shall have the same Multiplicity, Variant and config class as EthSwtPortMacLayerType.

No upstream Mapping as this decision is made on ECU-Configuration.

–Last change on issue 76343 comment 30–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.4 Specification Item ECUC_EthTrcv_00044

Trace References:

none

Content:

Name	EthTrcvPortMacLayerSpeedEthTrcvConfig.EthTrcvPortMacLayerSpeedin container EthTrcv Config		
Description	Defines the baud rate of the MAC layer.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	ETH_MAC_LAYER_SPEED_100MEthTrcvConfig.EthTrcvPortMacLayerSpeed.ETH_MAC_LAYER_SPEED_100M		
	ETH_MAC_LAYER_SPEED_10GEthTrcvConfig.EthTrcvPortMacLayerSpeed.ETH_MAC_LAYER_SPEED_10G		
	ETH_MAC_LAYER_SPEED_10MEthTrcvConfig.EthTrcvPortMacLayerSpeed.ETH_MAC_LAYER_SPEED_10M		
	ETH_MAC_LAYER_SPEED_1GEthTrcvConfig.EthTrcvPortMacLayerSpeed.ETH_MAC_LAYER_SPEED_1G		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	—	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	—	
Scope / Dependency	scope: ECU		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76343: Configuration parameters for Ethernet MAC layer types is incomplete

Problem description:

With https://www.autosar.org/bugzilla/show_bug.cgi?id=73074 the configuration of the MAC layer type was separated from the physical layer type. The configuration parameter `EthSwtPortMacLayerType` allows for the configuration of "families" of MAC layer types, namely xMII, xGMII, and xxGMII.

The concrete MAC layer type (e.g., SGMII vs. RGMII) however cannot be configured by this parameter. - Available switches (e.g., Broadcom's "Leo") however require the configuration of a *concrete* MAC layer type (and not only the configuration of a "family" of MAC layer types) for the individual port.

The same holds probably true for the Ethernet Driver and the Ethernet Transceiver driver.

One possibility to solve this is to introduce an additional config parameter named something like `EthSwtPortMacLayerSubType` with the following enum values "standard", "reduced", "reversed", "serial", "universal serial". - Thus for example the combination of `EthSwtPortMacLayerType=xGMII` and `EthSwtPortMacLayerSubType="reversed"` would yield RvGMII.

Agreed solution:

EthSwt:

add config parameter to container `EthSwtPort` named `EthSwtPortMacLayerSubType` with the following enum values "STANDARD: standard media-independent interface", "REDUCED: Reduced media-independent interface", "REVERSED: reversed media-independent interface (to provide direct connection between two Ethernet MACs)", "SERIAL: low-power and low pin-count serial 8b/10b-coded media-independent interface", "UNIVERSAL SERIAL: Universal low-power and low pin-count serial 8b/10b-coded media-independent interface".

add config parameter to container `EthSwtPort` named `EthSwtPortMacLayerSpeed` with the following enum values "ETH_MAC_LAYER_SPEED_10M", "ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G", "ETH_MAC_LAYER_SPEED_10G".

All parameters shall have the same Multiplicity, Variant and config class as `EthSwtPortMacLayerType`.

add description for `EthSwt_PortMacLayerSubTypes`:
Defines the MAC layer subtype of a switch port

add description for `EthSwt_PortMacLayerSpeed`:
Defines the baud rate of the MAC layer

Eth:

add config parameter to container EthCtrlConfig named EthCtrlMacLayerSubType with the following enum values "STANDARD", "REDUCED", "REVERSED", "SERIAL", "UNIVERSAL SERIAL". - Thus for example the combination of EthCtrlMacLayerType=xGMII and EthCtrlMacLayerSubType="REVERSED" would yield RvGMII.

add config parameter to container EthCtrlConfig named EthCtrlMacLayerSpeed with the following enum values "ETH_MAC_LAYER_SPEED_10M", "ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G", "ETH_MAC_LAYER_SPEED_10G".

add description for Eth_PortMacLayerSubTypes:
Defines the MAC layer subtype of a switch port

add description for Eth_PortMacLayerSpeed:
Defines the baud rate of the MAC layer

EthTrcv:

add config parameter to container EthTrcvCtrlConfig named EthTrcvPortMacLayerSubType with the following enum values "STANDARD", "REDUCED", "REVERSED", "SERIAL", "UNIVERSAL SERIAL". - Thus for example the combination of EthTrcvPortMacLayerType=xGMII and EthTrcvPortMacLayerSubType="REVERSED" would yield RvGMII.

add config parameter to container EthTrcvCtrlConfig named EthTrcvPortMacLayerSpeed with the following enum values "ETH_MAC_LAYER_SPEED_10M", "ETH_MAC_LAYER_SPEED_100M", "ETH_MAC_LAYER_SPEED_1G", "ETH_MAC_LAYER_SPEED_10G".

add description for EthTrcv_PortMacLayerSubTypes:
Defines the MAC layer subtype of a switch port

add description for EthTrcv_PortMacLayerSpeed:
Defines the baud rate of the MAC layer

All parameters shall have the same Multiplicity, Variant and config class as EthSwtPortMacLayerType.

No upstream Mapping as this decision is made on ECU-Configuration.
–Last change on issue 76343 comment 30–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.5 Specification Item ECUC_EthTrcv_00045

Trace References:

none

Content:

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

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77628: [EthSwt] Behaviour of certain APIs for ports without EthSwtPortTrcvRef (part I)

Problem description:

The following APIs have partly a description regarding the handling of ports without EthSwtPortTrcvRef:

EthSwt_GetPortSignalQuality

EthSwt_GetPortIdentifier

EthSwt_SetPortTestMode

EthTrcv_SetPhyTestMode

EthSwt_SetPortTxMode

EthSwt_GetPortCableDiagnosticsResult

The description should be adjusted and harmonized with related requirements.
–Last change on issue 77628 comment 2–

Agreed solution:

=== EthSwt ===

~[SWS_EthSwt_91014]EthSwt_GetPortSignalQuality

Description:

The function retrieves the signal quality of the link of the indexed Ethernet switch port.

~[SWS_EthSwt_00293]

The function EthSwt_GetPortSignalQuality() shall obtain the signal quality by calling the function EthTrcv_GetPhySignalQuality() of the referenced Ethernet Transceiver Driver. If the current signal quality is not available, the signal quality shall be set to 0xFF.

-[SWS_EthSwt_00298]as # 77349 introduces a general req

~[SWS_EthSwt_91015]EthSwt_GetPortIdentifier

Return value

E_NOT_OK: organizationally unique identifier of the Ethernet transceiver could not be obtained (i.e. OUI is not available)

Description:

This function retrieves the OUI (24 bit) of the indexed Ethernet switch port.

~[SWS_EthSwt_00299]

The function EthSwt_GetPortIdentifier() shall return the value of the organizationally unique identifier (OUI 24 bit) of the indexed Ethernet switch port that is connected to the indexed Ethernet switch. It shall set the 8 most significant bits of the OUI to 0xFFxxxxxx. If the Ethernet switch port references an Ethernet transceiver, the function shall obtain the OUI by calling the function EthTrcv_GetPhyIdentifier() and set the 8 most significant bits of the OUI to 0x00xxxxxx.

SWS_EthSwt_xxxxx] If neither the Ethernet switch port nor the Ethernet Transceiver Driver can provide an OUI the function EthSwt_GetPortIdentifier() shall return E_NOT_OK.

-[SWS_EthSwt_00304]as # 77349 introduces a general req

~[SWS_EthSwt_91029] EthSwt_SetPortTestMode

Description:

Activates a given test mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00328]

The function EthSwt_SetPortTestMode shall forward the call with the given test mode by calling the function EthTrcv_SetPhyTestMode() of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00333] as # 77349 introduces a general req

~[SWS_EthSwt_91023] EthSwt_SetPortLoopbackMode

Description:

Activates a given test loop-back mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00334]

The function EthSwt_SetPortLoopbackMode() shall forward the call with the given loop-back mode by calling the function EthTrcv_SetPhyLoopbackMode() of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00339] as # 77349 introduces a general req

~[SWS_EthSwt_91024] EthSwt_SetPortTxMode

Description:

Activates a given transmission mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00340]

The function EthSwt_SetPortTxMode() shall forward the call with the given transmission mode by calling the function EthTrcv_SetPhyTxMode() of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00345] as # 77349 introduces a general req

~[SWS_EthSwt_91025] EthSwt_GetPortCableDiagnosticsResult

Description:

Retrieves the cable diagnostics result of the indexed Ethernet switch port respectively the referenced Ethernet Transceiver Driver.

~[SWS_EthSwt_00346]

The function EthSwt_GetPortCableDiagnosticsResult() shall obtain the cable diagnostics result by calling the function EthTrcv_GetCableDiagnosticsResult() of the referenced Ethernet Transceiver Driver. If no Ethernet transceiver is referenced by the Ethernet switch port and development error detection is not enabled, the cable diagnostic result shall be set to ETHTRCV_CABLEDIAG_OK.

-[SWS_EthSwt_00351] as # 77349 introduces a general req

=== EthTrcv ===

add certain parameter to enable/disable API functions:

+ SWS item ECUC_EthTrcv_xxxx1

Name EthTrcvGetPhySignalQualityApi

Description Enables / Disables EthTrcv_GetPhySignalQuality API

+ SWS item ECUC_EthTrcv_xxxx2

Name EthTrcvGetPhyIdentifierApi

Description Enables / Disables EthTrcv_GetPhyIdentifier API

+ SWS item ECUC_EthTrcv_xxxx3

Name EthTrcvSetPhyTestModeApi

Description Enables / Disables EthTrcv_SetPhyTestMode API

+ SWS item ECUC_EthTrcv_xxxx4

Name EthTrcvSetPhyTxModeApi

Description Enables / Disables EthTrcv_SetPhyTxMode API

+ SWS item ECUC_EthTrcv_xxxx5

Name EthTrcvGetCableDiagnosticsResultApi

Description Enables / Disables EthTrcv_GetCableDiagnosticsResult API

add the following specification to the configuration parameter above:

Multiplicity 1

Type EcucBooleanParamDef

Default value –

Post-Build Variant Value false

Value Configuration

Class Pre-compile time X

All Variants Link time –

Post-build time –

Scope / Dependency scope: local

–Last change on issue 77628 comment 31–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.6 Specification Item ECUC_EthTrcv_00046

Trace References:

none

Content:

Name	EthTrcvGetPhyIdentifierApiEthTrcvGeneral.EthTrcvGetPhyIdentifierApi
------	---

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

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77628: [EthSwt] Behaviour of certain APIs for ports without EthSwtPortTrcvRef (part I)

Problem description:

The following APIs have partly a description regarding the handling of ports without EthSwtPortTrcvRef:

EthSwt_GetPortSignalQuality
EthSwt_GetPortIdentifier
EthSwt_SetPortTestMode
EthTrcv_SetPhyTestMode
EthSwt_SetPortTxMode
EthSwt_GetPortCableDiagnosticsResult

The description should be adjusted and harmonized with related requirements.
–Last change on issue 77628 comment 2–

Agreed solution:

=== EthSwt ===
~[SWS_EthSwt_91014]EthSwt_GetPortSignalQuality
Description:

The function retrieves the signal quality of the link of the indexed Ethernet switch port.

~[SWS_EthSwt_00293]

The function EthSwt_GetPortSignalQuality() shall obtain the signal quality by calling the function EthTrcv_GetPhySignalQuality() of the referenced Ethernet Transceiver Driver. If the current signal quality is not available, the signal quality shall be set to 0xFF.

-[SWS_EthSwt_00298]as # 77349 introduces a general req

~[SWS_EthSwt_91015]EthSwt_GetPortIdentifier

Return value

E_NOT_OK: organizationally unique identifier of the Ethernet transceiver could not be obtained (i.e. OUI is not available)

Description:

This function retrieves the OUI (24 bit) of the indexed Ethernet switch port.

~[SWS_EthSwt_00299]

The function EthSwt_GetPortIdentifier() shall return the value of the organizationally unique identifier (OUI 24 bit) of the indexed Ethernet switch port that is connected to the indexed Ethernet switch. It shall set the 8 most significant bits of the OUI to 0xFFxxxxxx. If the Ethernet switch port references an Ethernet transceiver, the function shall obtain the OUI by calling the function EthTrcv_GetPhyIdentifier() and set the 8 most significant bits of the OUI to 0x00xxxxxx.

SWS_EthSwt_00300] If neither the Ethernet switch port nor the Ethernet Transceiver Driver can provide an OUI the function EthSwt_GetPortIdentifier() shall return E_NOT_OK.

-[SWS_EthSwt_00304]as # 77349 introduces a general req

~[SWS_EthSwt_91029] EthSwt_SetPortTestMode

Description:

Activates a given test mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00328]

The function EthSwt_SetPortTestMode shall forward the call with the given test mode by calling the function EthTrcv_SetPhyTestMode() of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00333]as # 77349 introduces a general req

~[SWS_EthSwt_91023] EthSwt_SetPortLoopbackMode

Description:

Activates a given test loop-back mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00334]

The function EthSwt_SetPortLoopbackMode() shall forward the call with the given loop-back mode by calling the function EthTrcv_SetPhyLoopbackMode() of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00339] as # 77349 introduces a general req

~[SWS_EthSwt_91024] EthSwt_SetPortTxMode

Description:

Activates a given transmission mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00340]

The function EthSwt_SetPortTxMode() shall forward the call with the given transmission mode by calling the function EthTrcv_SetPhyTxMode() of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00345] as # 77349 introduces a general req

~[SWS_EthSwt_91025]EthSwt_GetPortCableDiagnosticsResult

Description:

Retrieves the cable diagnostics result of the indexed Ethernet switch port respectively the referenced Ethernet Transceiver Driver.

~[SWS_EthSwt_00346]

The function EthSwt_GetPortCableDiagnosticsResult() shall obtain the cable diagnostics result by calling the function EthTrcv_GetCableDiagnosticsResult() of the referenced Ethernet Transceiver Driver. If no Ethernet transceiver is referenced by the Ethernet switch port and development error detection is not enabled, the cable diagnostic result shall be set to ETHTRCV_CABLEDIAG_OK.

-[SWS_EthSwt_00351] as # 77349 introduces a general req

=== EthTrcv ===

add certain parameter to enable/disable API functions:

+ SWS item ECUC_EthTrcv_xxxx1

Name EthTrcvGetPhySignalQualityApi

Description Enables / Disables EthTrcv_GetPhySignalQuality API

+ SWS item ECUC_EthTrcv_xxxx2

Name EthTrcvGetPhyIdentifierApi

Description Enables / Disables EthTrcv_GetPhyIdentifier API

+ SWS item ECUC_EthTrcv_xxxx3

Name EthTrcvSetPhyTestModeApi

Description Enables / Disables EthTrcv_SetPhyTestMode API

+ SWS item ECUC_EthTrcv_xxxx4

Name EthTrcvSetPhyTxModeApi

Description Enables / Disables EthTrcv_SetPhyTxMode API

+ SWS item ECUC_EthTrcv_xxxx5

Name EthTrcvGetCableDiagnosticsResultApi

Description Enables / Disables EthTrcv_GetCableDiagnosticsResult API

add the following specification to the configuration parameter above:

Multiplicity 1

Type EcucBooleanParamDef

Default value –

Post-Build Variant Value false

Value Configuration

Class Pre-compile time X

All Variants Link time –

Post-build time –

Scope / Dependency scope: local

–Last change on issue 77628 comment 31–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.7 Specification Item ECUC_EthTrcv_00047

Trace References:

none

Content:

Name	EthTrcvSetPhyTestModeApiEthTrcvGeneral.EthTrcvSetPhyTestModeApi
Parent Container	EthTrcvGeneral
Description	Enables / Disables EthTrcv_SetPhyTestMode API.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Post-Build Variant Multiplicity	false

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

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77628: [EthSwt] Behaviour of certain APIs for ports without EthSwtPortTrcvRef (part I)

Problem description:

The following APIs have partly a description regarding the handling of ports without EthSwtPortTrcvRef:

EthSwt_GetPortSignalQuality
EthSwt_GetPortIdentifier
EthSwt_SetPortTestMode
EthTrcv_SetPhyTestMode
EthSwt_SetPortTxMode
EthSwt_GetPortCableDiagnosticsResult

The description should be adjusted and harmonized with related requirements.
–Last change on issue 77628 comment 2–

Agreed solution:

=== EthSwt ===

~[SWS_EthSwt_91014]EthSwt_GetPortSignalQuality

Description:

The function retrieves the signal quality of the link of the indexed Ethernet switch port.

~[SWS_EthSwt_00293]

The function EthSwt_GetPortSignalQuality() shall obtain the signal quality by calling the function EthTrcv_GetPhySignalQuality() of the referenced Ethernet Transceiver Driver. If the current signal quality is not available, the signal quality shall be set to 0xFF.

-[SWS_EthSwt_00298]as # 77349 introduces a general req

~[SWS_EthSwt_91015]EthSwt_GetPortIdentifier

Return value

E_NOT_OK: organizationally unique identifier of the Ethernet transceiver could not be obtained (i.e. OUI is not available)

Description:

This function retrieves the OUI (24 bit) of the indexed Ethernet switch port.

~[SWS_EthSwt_00299]

The function EthSwt_GetPortIdentifier() shall return the value of the organizationally unique identifier (OUI 24 bit) of the indexed Ethernet switch port that is connected to the indexed Ethernet switch. It shall set the 8 most significant bits of the OUI to 0xFFxxxxxx. If the Ethernet switch port references an Ethernet transceiver, the function shall obtain the OUI by calling the function EthTrcv_GetPhyIdentifier() and set the 8 most significant bits of the OUI to 0x00xxxxxx.

SWS_EthSwt_00000] If neither the Ethernet switch port nor the Ethernet Transceiver Driver can provide an OUI the function EthSwt_GetPortIdentifier() shall return E_NOT_OK.

-[SWS_EthSwt_00304]as # 77349 introduces a general req

~[SWS_EthSwt_91029] EthSwt_SetPortTestMode

Description:

Activates a given test mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00328]

The function EthSwt_SetPortTestMode shall forward the call with the given test mode by calling the function EthTrcv_SetPhyTestMode() of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00333]as # 77349 introduces a general req

~[SWS_EthSwt_91023] EthSwt_SetPortLoopbackMode

Description:

Activates a given test loop-back mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00334]

The function EthSwt_SetPortLoopbackMode() shall forward the call with the given loop-back mode by calling the function EthTrcv_SetPhyLoopbackMode() of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00339] as # 77349 introduces a general req

~[SWS_EthSwt_91024] EthSwt_SetPortTxMode

Description:

Activates a given transmission mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00340]

The function EthSwt_SetPortTxMode() shall forward the call with the given transmission mode by calling the function EthTrcv_SetPhyTxMode() of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00345] as # 77349 introduces a general req

~[SWS_EthSwt_91025]EthSwt_GetPortCableDiagnosticsResult

Description:

Retrieves the cable diagnostics result of the indexed Ethernet switch port respectively the referenced Ethernet Transceiver Driver.

~[SWS_EthSwt_00346]

The function EthSwt_GetPortCableDiagnosticsResult() shall obtain the cable diagnostics result by calling the function EthTrcv_GetCableDiagnosticsResult() of the referenced Ethernet Transceiver Driver. If no Ethernet transceiver is referenced by the Ethernet switch port and development error detection is not enabled, the cable diagnostic result shall be set to ETHTRCV_CABLEDIAG_OK.

-[SWS_EthSwt_00351] as # 77349 introduces a general req

=== EthTrcv ===

add certain parameter to enable/disable API functions:

+ SWS item ECUC_EthTrcv_xxxx1

Name EthTrcvGetPhySignalQualityApi

Description Enables / Disables EthTrcv_GetPhySignalQuality API

+ SWS item ECUC_EthTrcv_xxxx2

Name EthTrcvGetPhyIdentifierApi

Description Enables / Disables EthTrcv_GetPhyIdentifier API

+ SWS item ECUC_EthTrcv_xxxx3

Name EthTrcvSetPhyTestModeApi

Description Enables / Disables EthTrcv_SetPhyTestMode API

+ SWS item ECUC_EthTrcv_xxxx4

Name EthTrcvSetPhyTxModeApi

Description Enables / Disables EthTrcv_SetPhyTxMode API

+ SWS item ECUC_EthTrcv_xxxx5

Name EthTrcvGetCableDiagnosticsResultApi

Description Enables / Disables EthTrcv_GetCableDiagnosticsResult API

add the following specification to the configuration parameter above:

Multiplicity 1

Type EcucBooleanParamDef

Default value –

Post-Build Variant Value false

Value Configuration

Class Pre-compile time X

All Variants Link time –

Post-build time –

Scope / Dependency scope: local

–Last change on issue 77628 comment 31–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.8 Specification Item ECUC_EthTrcv_00048

Trace References:

none

Content:

Name	EthTrcvSetPhyTxModeApiEthTrcvGeneral.EthTrcvSetPhyTxModeApi		
Parent Container	EthTrcvGeneral		
Description	Enables / Disables EthTrcv_SetPhyTxMode API.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

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

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77628: [EthSwt] Behaviour of certain APIs for ports without EthSwtPortTrcvRef (part I)

Problem description:

The following APIs have partly a description regarding the handling of ports without EthSwtPortTrcvRef:

EthSwt_GetPortSignalQuality
EthSwt_GetPortIdentifier
EthSwt_SetPortTestMode
EthTrcv_SetPhyTestMode
EthSwt_SetPortTxMode
EthSwt_GetPortCableDiagnosticsResult

The description should be adjusted and harmonized with related requirements.
–Last change on issue 77628 comment 2–

Agreed solution:

=== EthSwt ===

~[SWS_EthSwt_91014]EthSwt_GetPortSignalQuality

Description:

The function retrieves the signal quality of the link of the indexed Ethernet switch port.

~[SWS_EthSwt_00293]

The function EthSwt_GetPortSignalQuality() shall obtain the signal quality by calling the function EthTrcv_GetPhySignalQuality() of the referenced Ethernet Transceiver Driver. If the current signal quality is not available, the signal quality shall be set to 0xFF.

-[SWS_EthSwt_00298]as # 77349 introduces a general req

~[SWS_EthSwt_91015]EthSwt_GetPortIdentifier

Return value

E_NOT_OK: organizationally unique identifier of the Ethernet transceiver could not

be obtained (i.e. OUI is not available)

Description:

This function retrieves the OUI (24 bit) of the indexed Ethernet switch port.

~[SWS_EthSwt_00299]

The function EthSwt_GetPortIdentifier() shall return the value of the organizationally unique identifier (OUI 24 bit) of the indexed Ethernet switch port that is connected to the indexed Ethernet switch. It shall set the 8 most significant bits of the OUI to 0xFFxxxxxx. If the Ethernet switch port references an Ethernet transceiver, the function shall obtain the OUI by calling the function EthTrcv_GetPhyIdentifier() and set the 8 most significant bits of the OUI to 0x00xxxxxx.

SWS_EthSwt_00300] If neither the Ethernet switch port nor the Ethernet Transceiver Driver can provide an OUI the function EthSwt_GetPortIdentifier() shall return E_NOT_OK.

-[SWS_EthSwt_00304] as # 77349 introduces a general req

~[SWS_EthSwt_91029] EthSwt_SetPortTestMode

Description:

Activates a given test mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00328]

The function EthSwt_SetPortTestMode shall forward the call with the given test mode by calling the function EthTrcv_SetPhyTestMode() of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00333] as # 77349 introduces a general req

~[SWS_EthSwt_91023] EthSwt_SetPortLoopbackMode

Description:

Activates a given test loop-back mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00334]

The function EthSwt_SetPortLoopbackMode() shall forward the call with the given loop-back mode by calling the function EthTrcv_SetPhyLoopbackMode() of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00339] as # 77349 introduces a general req

~[SWS_EthSwt_91024] EthSwt_SetPortTxMode

Description:

Activates a given transmission mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00340]

The function `EthSwt_SetPortTxMode()` shall forward the call with the given transmission mode by calling the function `EthTrcv_SetPhyTxMode()` of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00345] as # 77349 introduces a general req

~[SWS_EthSwt_91025]`EthSwt_GetPortCableDiagnosticsResult`

Description:

Retrieves the cable diagnostics result of the indexed Ethernet switch port respectively the referenced Ethernet Transceiver Driver.

~[SWS_EthSwt_00346]

The function `EthSwt_GetPortCableDiagnosticsResult()` shall obtain the cable diagnostics result by calling the function `EthTrcv_GetCableDiagnosticsResult()` of the referenced Ethernet Transceiver Driver. If no Ethernet transceiver is referenced by the Ethernet switch port and development error detection is not enabled, the cable diagnostic result shall be set to `ETHTRCV_CABLEDIAG_OK`.

-[SWS_EthSwt_00351] as # 77349 introduces a general req

=== `EthTrcv` ===

add certain parameter to enable/disable API functions:

+ SWS item `ECUC_EthTrcv_xxxx1`

Name `EthTrcvGetPhySignalQualityApi`

Description Enables / Disables `EthTrcv_GetPhySignalQuality` API

+ SWS item `ECUC_EthTrcv_xxxx2`

Name `EthTrcvGetPhyIdentifierApi`

Description Enables / Disables `EthTrcv_GetPhyIdentifier` API

+ SWS item `ECUC_EthTrcv_xxxx3`

Name `EthTrcvSetPhyTestModeApi`

Description Enables / Disables `EthTrcv_SetPhyTestMode` API

+ SWS item `ECUC_EthTrcv_xxxx4`

Name `EthTrcvSetPhyTxModeApi`

Description Enables / Disables `EthTrcv_SetPhyTxMode` API

+ SWS item `ECUC_EthTrcv_xxxx5`

Name `EthTrcvGetCableDiagnosticsResultApi`

Description Enables / Disables `EthTrcv_GetCableDiagnosticsResult` API

add the following specification to the configuration parameter above:

Multiplicity 1

Type `EcucBooleanParamDef`

Default value –
Post-Build Variant Value false
Value Configuration
Class Pre-compile time X
All Variants Link time –
Post-build time –
Scope / Dependency scope: local
–Last change on issue 77628 comment 31–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.9 Specification Item ECUC_EthTrcv_00049

Trace References:

none

Content:

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

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77628: [EthSwt] Behaviour of certain APIs for ports without EthSwtPortTrcvRef (part I)

Problem description:

The following APIs have partly a description regarding the handling of ports without EthSwtPortTrcvRef:

EthSwt_GetPortSignalQuality
EthSwt_GetPortIdentifier
EthSwt_SetPortTestMode
EthTrcv_SetPhyTestMode
EthSwt_SetPortTxMode
EthSwt_GetPortCableDiagnosticsResult

The description should be adjusted and harmonized with related requirements.
–Last change on issue 77628 comment 2–

Agreed solution:

=== EthSwt ===

~[SWS_EthSwt_91014]EthSwt_GetPortSignalQuality

Description:

The function retrieves the signal quality of the link of the indexed Ethernet switch port.

~[SWS_EthSwt_00293]

The function EthSwt_GetPortSignalQuality() shall obtain the signal quality by calling the function EthTrcv_GetPhySignalQuality() of the referenced Ethernet Transceiver Driver. If the current signal quality is not available, the signal quality shall be set to 0xFF.

-[SWS_EthSwt_00298]as # 77349 introduces a general req

~[SWS_EthSwt_91015]EthSwt_GetPortIdentifier

Return value

E_NOT_OK: organizationally unique identifier of the Ethernet transceiver could not be obtained (i.e. OUI is not available)

Description:

This function retrieves the OUI (24 bit) of the indexed Ethernet switch port.

~[SWS_EthSwt_00299]

The function EthSwt_GetPortIdentifier() shall return the value of the organizationally unique identifier (OUI 24 bit) of the indexed Ethernet switch port that is connected to the indexed Ethernet switch. It shall set the 8 most significant bits of the OUI

to 0xFFxxxxxx. If the Ethernet switch port references an Ethernet transceiver, the function shall obtain the OUI by calling the function `EthTrcv_GetPhyIdentifier()` and set the 8 most significant bits of the OUI to 0x00xxxxxx.

SWS_EthSwt_00000] If neither the Ethernet switch port nor the Ethernet Transceiver Driver can provide an OUI the function `EthSwt_GetPortIdentifier()` shall return `E_NOT_OK`.

-[SWS_EthSwt_00304] as # 77349 introduces a general req

~[SWS_EthSwt_91029] `EthSwt_SetPortTestMode`

Description:

Activates a given test mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00328]

The function `EthSwt_SetPortTestMode` shall forward the call with the given test mode by calling the function `EthTrcv_SetPhyTestMode()` of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00333] as # 77349 introduces a general req

~[SWS_EthSwt_91023] `EthSwt_SetPortLoopbackMode`

Description:

Activates a given test loop-back mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00334]

The function `EthSwt_SetPortLoopbackMode()` shall forward the call with the given loop-back mode by calling the function `EthTrcv_SetPhyLoopbackMode()` of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00339] as # 77349 introduces a general req

~[SWS_EthSwt_91024] `EthSwt_SetPortTxMode`

Description:

Activates a given transmission mode of the indexed Ethernet switch port.

~[SWS_EthSwt_00340]

The function `EthSwt_SetPortTxMode()` shall forward the call with the given transmission mode by calling the function `EthTrcv_SetPhyTxMode()` of the referenced Ethernet Transceiver Driver.

-[SWS_EthSwt_00345] as # 77349 introduces a general req

~[SWS_EthSwt_91025] `EthSwt_GetPortCableDiagnosticsResult`

Description:

Retrieves the cable diagnostics result of the indexed Ethernet switch port respec-

tively the referenced Ethernet Transceiver Driver.

~[SWS_EthSwt_00346]

The function EthSwt_GetPortCableDiagnosticsResult() shall obtain the cable diagnostics result by calling the function EthTrcv_GetCableDiagnosticsResult() of the referenced Ethernet Transceiver Driver. If no Ethernet transceiver is referenced by the Ethernet switch port and development error detection is not enabled, the cable diagnostic result shall be set to ETHTRCV_CABLEDIAG_OK.

-[SWS_EthSwt_00351] as # 77349 introduces a general req

=== EthTrcv ===

add certain parameter to enable/disable API functions:

+ SWS item ECUC_EthTrcv_xxxx1

Name EthTrcvGetPhySignalQualityApi

Description Enables / Disables EthTrcv_GetPhySignalQuality API

+ SWS item ECUC_EthTrcv_xxxx2

Name EthTrcvGetPhyIdentifierApi

Description Enables / Disables EthTrcv_GetPhyIdentifier API

+ SWS item ECUC_EthTrcv_xxxx3

Name EthTrcvSetPhyTestModeApi

Description Enables / Disables EthTrcv_SetPhyTestMode API

+ SWS item ECUC_EthTrcv_xxxx4

Name EthTrcvSetPhyTxModeApi

Description Enables / Disables EthTrcv_SetPhyTxMode API

+ SWS item ECUC_EthTrcv_xxxx5

Name EthTrcvGetCableDiagnosticsResultApi

Description Enables / Disables EthTrcv_GetCableDiagnosticsResult API

add the following specification to the configuration parameter above:

Multiplicity 1

Type EcucBooleanParamDef

Default value –

Post-Build Variant Value false

Value Configuration

Class Pre-compile time X

All Variants Link time –

Post-build time –

Scope / Dependency scope: local

–Last change on issue 77628 comment 31–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.10 Specification Item SWS_EthTrcv_00017**Trace References:**

none

Content:

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

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors

–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API
UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW
General).

–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.11 Specification Item SWS_EthTrcv_00027

Trace References:

none

Content:

Module	Imported Type
ComStack_Types	BufReq_ReturnType
Dem	Dem_EventIdType
	Dem_EventStatusType
EcuM	EcuM_WakeupSourceType

Module	Imported Type
Eth_GeneralTypes	EthTrcv_BaudRateType
	EthTrcv_CableDiagResultType
	EthTrcv_ConfigType
	EthTrcv_DuplexModeType
	EthTrcv_LinkStateType
	EthTrcv_ModeType
	EthTrcv_PhyLoopbackModeType
	EthTrcv_PhyTestModeType
	EthTrcv_PhyTxModeType
	EthTrcv_WakeupModeType
Eth_BuflIdxType	
Eth_ConfigType	
Eth_FrameType	
Eth_ModeType	
Eth_RxStatusType	
Icu	Icu_ChannelType
Std_Types	Std_ReturnType
	Std_VersionInfoType

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76879: [EthTrcv] Add EthSwt API's to optional interfaces

Problem description:

Optional interfaces are missing, in case the Ethernet transceiver is connected to a Ethernet switch

Agreed solution:

add the following API's to optional interfaces:

EthSwt_ReadTrcvRegister

EthSwt_WriteTrcvRegister

move the following API's from mandatory interfaces to optional interfaces:

Eth_ReadMii

Eth_WriteMii

–Last change on issue 76879 comment 1–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.12 Specification Item SWS_EthTrcv_00044

Trace References:

none

Content:

If **default 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 **default development** error ETHTRCV_E_ **NOT_INITIALIZED UNINIT** otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors
–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.13 Specification Item SWS_EthTrcv_00050

Trace References:

none

Content:

If **default 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 **default development** error ETHTRCV_E_ **NOT_INITIALIZED UNINIT** otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors

–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

***** BSW UML Model *******SWS_CanNm:**

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

***** ECUC XML *****

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.14 Specification Item SWS_EthTrcv_00057

Trace References:

none

Content:

If **default 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 **default de-**

development error ETHTRCV_E_NOT_INITIALIZED UNINIT otherwise (if DET is disabled)
return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors
–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API
UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.15 Specification Item SWS_EthTrcv_00063

Trace References:

none

Content:

If **default 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 **default development** error ETHTRCV_E_NOT_INITIALIZED UNINIT otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors
–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API
UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW
General).

–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.16 Specification Item SWS_EthTrcv_00070

Trace References:

none

Content:

If **default 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 **default de-**
velopment error ETHTRCV_E_ **NOT_INITIALIZED UNINIT** otherwise (if DET is disabled)
return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors

–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API
UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

—Last change on issue 59085 comment 88—

BW-C-Level:

Application	Specification	Bus
1	4	1

1.17 Specification Item SWS_EthTrcv_00077

Trace References:

none

Content:

If **default 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 **default development** error ETHTRCV_E_ **NOT_INITIALIZED UNINIT** otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors
—Last change on issue 59085 comment 26—

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API
UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.18 Specification Item SWS_EthTrcv_00085

Trace References:

none

Content:

API function	Description
Dem_SetEventStatus	Called by SW-Cs or BSW modules to report monitor status information to the Dem. BSW modules calling Dem_SetEvent Status can safely ignore the return value.
Eth_GetControllerMode	Obtains the state of the indexed controller
Eth_GetPhysAddr	Obtains the physical source address used by the indexed controller
Eth_GetVersionInfo	Returns the version information of this module
Eth_Init	Initializes the Ethernet Driver
Eth_ProvideTxBuffer	Provides access to a transmit buffer of the FIFO related to the specified priority
Eth_ReadMii	Reads a transceiver register
Eth_Receive	Receive a frame from the related fifo.
Eth_SetControllerMode	Enables / disables the indexed controller
Eth_Transmit	Triggers transmission of a previously filled transmit buffer
Eth_TxConfirmation	Triggers frame transmission confirmation
Eth_WriteMii	Configures a transceiver register or triggers a function offered by the receiver
EthIf_TrvcModeIndication	Called asynchronously when mode a mode change has been read out. Triggered by previous EthIf the function is triggered by previous call of EthTrcv_SetTransceiverMode call. Can it can directly be called within the trigger functionsfunction.
SchM_Enter_EthTrcv	Invokes the SchM_Enter function to enter a module local exclusive area.
SchM_Exit_EthTrcv	Invokes the SchM_Exit function to exit an exclusive area.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors
–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API
UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

- RfC #76528: [EthTrcv] Description of mandatory API "EthIf_TrcvModeIndication" seem to be partly wrong

Problem description:

The description for the mandatory API "EthIf_TrcvModeIndication" says:

"

...

Triggered by Eth_SetTranceiverMode ...

...

"

I think this should be changed to "Triggered by EthTrcv_SetTranceiverMode"

Agreed solution:

Change API name in the description of EthIf_TrcvModeIndication in requirement SWS_EthTrcv_00085
from ... Eth_SetTranceiverMode ...
to ... EthTrcv_SetTranceiverMode ...

Also change the description in the EthIf
~SWS_EthIf_00232
–Last change on issue 76528 comment 2–

BW-C-Level:

Application	Specification	Bus
1	1	1

- RfC #76628: [EthTrcv] Remove certain Ethernet API's from chapter mandatory interfaces

Problem description:

Some mandatory API's provided by the Ethernet Driver are not used in the Ethernet Transceiver Driver:

Eth_GetControllerMode

Eth_SetControllerMode

Eth_GetPhysAddr

Eth_GetVersionInfo

Eth_Init

Eth_ProvideTxBuffer

Eth_Receive

Eth_Transmit

Eth_TxConfirmation

–Last change on issue 76628 comment 1–

Agreed solution:

~SWS_EthTrcv_00085: remove the following mandatory API's :

Eth_GetControllerMode

Eth_SetControllerMode

Eth_GetPhysAddr

Eth_GetVersionInfo

Eth_Init

Eth_ProvideTxBuffer

Eth_Receive

Eth_Transmit

Eth_TxConfirmation

BW-C-Level:

Application	Specification	Bus
1	1	1

- RfC #77250: [EthIf][EthSwt][EthTrcv] EthSwt_SetSwitchPortMode() broken for ports without Ethernet transceiver

Problem description:

SWS_EthSwt_00019 states that EthSwt_SetSwitchPortMode() shall (unconditionally) invoke EthTrcv_SetTransceiverMode() of the Ethernet Transceiver Driver.

This is bogus for the following two reasons:

1) a particular switch port does not necessarily have an associated Ethernet transceiver. - In that case EthSwtPortTrcvRef won't be present. This is probably the case for MAC2MAC links if EthSwtPortRole is present and either set to ETH-SWT_HOST_PORT or to ETHSWT_UP_LINK_PORT.

2) Even if the particular switch port has an associated Ethernet transceiver, SWS_EthTrcv_00043 states that the EthTrcv shall invoke EthIf_TrvcModeIndication() as a response to EthTrcv_SetTransceiverMode().

IMHO 1) should be solved by extending SWS_EthSwt_00019 distinguishing between ports which are associated with a transceiver (i.e., EthSwtPortTrcvRef present) - in that case EthTrcv_SetTransceiverMode() should be called - and ports which are not associated with a transceiver (i.e., EthSwtPortTrcvRef is not present) - in that case something else has to happen and this needs to be specified (probably the switch port shall be internally just disabled - maybe this needs to happen in the first case as well?)

2) leads to the following call chain: EthIf_SwitchPortGroupRequestMode()
-> EthSwt_SetSwitchPortMode() -> EthTrcv_SetTransceiverMode() ->
EthIf_TrvcvModeIndication(). Thus the indication actually bypasses the EthSwt
which IMHO is extremely ugly. Additionally for switch ports without associated
transceiver, the EthIf will never get a EthIf_TrvcvModeIndication() ...

Agreed solution:

=== System Template ===

+ Add upstream mapping of EthSwtPortTrcvRef to CouplingPort.physicalLayerType

=== EthSwt ===

~ch. 7.2.4 Production Errors

move SWS_EthSwt_00113 to 7.2.5 Extended production Errors

~SWS_EthSwt_00113 change detection criteria to

Fail:When access to the Ethernet Switch fails the module shall report the extended
production error with event status DEM_EVENT_STATUS_PREFAILED to DEM.

Pass:When access to the Ethernet Switch succeeds the module shall report the
extended production error with event status DEM_EVENT_STATUS_PREPASSED
to DEM.

+ SWS_EthSwt_xxxxx new ext. production Error

Error Name: ETHSWT_E_SYNCPORT2PHY

Short Description: Ethernet switch port and the referenced Ethernet transceiver are
in contradicting modes.

Long description: While getting the Ethernet switch port mode, the Ethernet
switch driver detected an inconsistent state between Ethernet switch port and the
referenced Ethernet transceiver Mode.

Detection Criteria: Fail: When getting the Ethernet switch port mode together with
the Ethernet transceiver mode and the mode of the two referenced modules was
found

inconsistent the module shall report the extended production error with event status
DEM_EVENT_STATUS_PREFAILED to DEM.

Pass:When getting the Ethernet switch port mode together with the Ethernet
transceiver mode and the mode of the two referenced modules was found

consistent the module shall report the extended production error with event status
DEM_EVENT_STATUS_PREPASSED to DEM.

all other attributer: N/A

~ ch. 8.3.2 EthSwt_SwitchInit

~[SWS_EthSwt_00016] change "production error" to "extended production error"

~ch. 8.3.3 EthSwt_SetSwitchPortMode

~[SWS_EthSwt_00019]The function EthSwt_SetSwitchPortMode() shall put the indexed port of the switch into the specified mode. If EthSwtPort references an EthTrcv then the function EthTrcv_SetTransceiverMode() of the Ethernet Transceiver Driver shall additionally be called with the corresponding transceiver mode.

+ [SWS_EthSwt_xxxxx] When calling the function EthSwt_SetSwitchPortMode() with mode ETHTRCV_MODE_DOWN, the EthSwt shall disable the Ethernet switch port directly for reduction of power consumption, if it is possible.

+ [SWS_EthSwt_xxxxx] When calling the function EthSwt_SetSwitchPortMode(), the function shall check the access to the Ethernet switch driver. If the check fails, the function shall raise the extended production error ETHSWT_E_ACCESS and return E_NOT_OK, otherwise pass the extended production error ETHSWT_E_ACCESS and return E_OK.

+ [SWS_EthSwt_xxxxxx] If EthSwtPort does not references an EthTrcv, EthSwt shall indicate a mode of the port by the API EthIf_SwitchPortModeIndication latest during the next EthSwt_MainFunction.(SRS_ETH_00118)

~ch. 10.1.3 EthSwtDemEventParameterRefs

+ SWS Item ECUC_EthSwt_xxxxx1 : Name ETHSWT_E_SYNCPORT2PHY

Parent Container EthSwtDemEventParameterRefs

Description Reference to the DemEventParameter which shall be issued when the error "Ethernet switch port and the referenced Ethernet transceiver are in contradicting modes" has occurred.

Multiplicity 0..1

Type Symbolic name reference to [DemEventParameter]

Post-Build Variant Multiplicity true

Post-Build Variant Value true

Multiplicity Configuration

Class Pre-compile time X VARIANT-PRE-COMPILE

Link time X VARIANT-LINK-TIME

Post-build time X VARIANT-POST-BUILD

Value Configuration Class

Pre-compile time X VARIANT-PRE-COMPILE

Link time X VARIANT-LINK-TIME

Post-build time X VARIANT-POST-BUILD

Scope / Dependency

scope: local

~ch. 10.1.5 EthSwtPort

~ECUC_EthSwt_00041 EthSwtPortTrcvRef change following attributes:
Multiplicity Configuration Class | Link time | VARIANT-LINK-TIME, VARIANT-POST-BUILD
| Post build | –
add to dependency: If EthSwtPortPhysicalLayerType is defined, then EthSwtPortTrcvRef holds the reference to the corresponding EthTrcv.

~ECUC_EthSwt_00054 add dependency to EthSwtPortPhysicalLayerType: If a EthSwtPort has an EthSwtPortPhysicalLayerType then EthSwtPort shall reference an EthTrcv.

=== EthIf ===

Add to chapter 8.4 (callback notifications):

+ EthIf_SwitchPortModeIndication

Syntax: void EthIf_SwitchPortModeIndication(uint8 SwitchIdx, uint8 SwitchPortIdx, EthSwt_ModeType PortMode)

Service ID: pick a free one

Sync/Async: Synchronous

Reentrancy: Non Reentrant

Parameters in: SwitchIdx: Index of the switch within the context of the Ethernet Switch Driver

SwitchPortIdx: Index of the port at the addressed switch

PortMode: notified Ethernet Switch port mode.

Description: The EthIf shall determine the expected notifications based on the EthSwtPort configuration. In case the EthSwtPort references an EthTrcv the EthIf expects a notification from the EthTrcv via API EthIf_TrvcModeIndication(). Otherwise the EthIf expects a notification from the EthSwt via API EthIf_SwitchPortModeIndication().

~SWS_EthIf_00232

change from

Called asynchronously when mode has been read out. Triggered by previous EthTrcv_SetTransceiverMode call. Can directly be called within the trigger functions.

To

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.

–Last change on issue 77250 comment 50–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.19 Specification Item SWS_EthTrcv_00086

Trace References:

none

Content:

API function	Description
Det_ReportError	Service to report development errors.
EcuM_SetWakeupEvent	Sets the wakeup event.
Eth_ReadMii	Reads a transceiver register
Eth_WriteMii	Configures a transceiver register or triggers a function offered by the receiver
EthSwt_ReadTrcvRegister	Generic API for reading the content of a transceiver register
EthSwt_WriteTrcvRegister	Generic API for writing the content of a transceiver register
Icu_DisableNotification	This function disables the notification of a channel.
Icu_EnableNotification	This function enables the notification on the given channel.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76404: [Det] Clarifications on runtime errors

Problem description:

There are several uncertainties/problems in the SWS DET:

1. According to SWS_Det_00180, the callouts should have the same signatures as the corresponding DET functions, but they are void(void) (SWS_Det_00181, SWS_Det_00184, SWS_Det_00187).
2. Section 8.2.3.1 does not describe how the instance ID is passed to DET.
3. Configuration of header files for all three error type callouts are missing.
4. Why does the development error callout reside in DetNotification, while the other two callouts reside in DetGeneral?
5. The limitation in section 4.1 regarding "supervisor mode" does not really make sense. It is assumed that the DET is ignorant regarding the call context, and the software receiving DET callbacks (like DLT or the implementers of the callouts) need to take care of resolving the calling context, if necessary (e.g. in multi-core environments).
6. SWS_Det_00302 defines several runtime errors. But apart from DET_E_CANNOT_REPORT, it is unclear in which situation these errors could be reported by DET: For errors reported by BSW, the DET has no means to validate anything that could lead to such an error. And for SWCs, the modeling already takes care that DET_E_WRONG_MODULE and DET_E_WRONG_INSTANCE can-

not occur, while the other two errors can also not be checked by DET without further configuration.

7. Det_ReportTransientFault (SWS_Det_01003) shall return the return value of a configured callout. But what shall happen if more than one callout exists, and the return different values?

8. SWS_Det_00052: The only API that can result in DET_E_PARAM_POINTER is Det_GetVersionInfo (as the error description mentions correctly). Please reformulate this requirement and move it to section 8.1.3.6 "Det_GetVersionInfo".

–Last change on issue 76404 comment 13–

Agreed solution:

1.

~change SWS_Det_00181/184/187 such that signatures match the APIs

~Figures 3,5, and 7 to be corrected (return missing)

5. remove from 4.1. the sentence: "It is assumed that the whole Basic Software runs in supervisor mode or the switch to supervisor mode is done by a system call within the error reporting function of the DET module."

6. remove SWS_Det_00302 and SWS_Det_00303 and all included errors

7. change SWS_Det_01003 (Return Value-Part only): "Std_ReturnType" If no callout exists it shall return E_OK, otherwise it shall return the value of the configured callout. In case several callouts are configured the logical or (sum) of the callout return values shall be returned. Rationale: since E_OK=0, E_OK will be only returned if all are E_OK, and for multiple error codes there is a good chance to detect several of them.

8. change SWS_Det_00052 from "in case a null pointer error occurs." to "in case a null pointer error occurs in Det_GetVersionInfo." Do not move the requirement, since otherwise the section 7.7 would be empty, but add the following sentence to 8.1.3.6: "In case a null pointer is passed, DET_E_PARAM_POINTER is returned, see SWS_Det_00052."

–Last change on issue 76404 comment 30–

BW-C-Level:

Application	Specification	Bus
1	4	1

- RfC #76879: [EthTrcv] Add EthSwt API's to optional interfaces

Problem description:

Optional interfaces are missing, in case the Ethernet transceiver is connected to a Ethernet switch

Agreed solution:

add the following API's to optional interfaces:

EthSwt_ReadTrcvRegister

EthSwt_WriteTrcvRegister

move the following API's from mandatory interfaces to optional interfaces:

Eth_ReadMii

Eth_WriteMii

–Last change on issue 76879 comment 1–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.20 Specification Item SWS_EthTrcv_00122

Trace References:

none

Content:

If **default development** error detection is enabled: The function EthTrcv_SetTransceiverWakeupMode() shall check that the service EthTrcv_Init was previously called. If the check fails, the function shall raise the **default development** error ETHTRCV_E_NOT_INITIALIZED UNINIT otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors

–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.

- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.21 Specification Item SWS_EthTrcv_00129

Trace References:

none

Content:

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

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors
–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API
UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.22 Specification Item SWS_EthTrcv_00131

Trace References:

none

Content:

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

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76524: `[EthIf][EthTrcv]` missing definition for `ETHIF_E_INV_POINTER` / `ETHTRCV_E_INV_POINTER`

Problem description:

`ETHIF_E_INV_POINTER` appears in `[SWS_EthIf_00242]` `[SWS_EthIf_00323]` `[SWS_EthIf_00333]` `[SWS_EthIf_00337]` `[SWS_EthIf_00338]` `[SWS_EthIf_00339]`.

However, there's no definition for `ETHIF_E_INV_POINTER`.

As **default error list** (`[SWS_EthIf_00017]`) already have `ETHIF_E_PARAM_POINTER` `0x05` (I believe it fits to the purpose of `ETHIF_E_INV_POINTER`), could you consider to replace `ETHIF_E_INV_POINTER` with `ETHIF_E_PARAM_POINTER`, please?

Also, same issue exists at `EthTrcv` (`ETHTRCV_E_INV_POINTER` not defined at `[SWS_EthTrcv_00131]`, `ETHTRCV_E_PARAM_POINTER` exists).

Note:

- `EthSwt` has `ETHSWT_E_INV_POINTER`, but no `ETHSWT_E_PARAM_POINTER` (different default error identifier style has been applied, is handled by RfC # 76195).
- `WdgIf` has both `WDGIF_E_INV_POINTER` (this is used) and `WDGIF_E_PARAM_POINTER` (added by RfC # 59818 and RfC # 62544), but `WDGIF_E_PARAM_POINTER` is not used in the SWS (as mentioned in 62544).

- Xcp @ R4.2.2 has both XCP_E_INV_POINTER (this is used) and XCP_E_PARAM_POINTER, but XCP_E_PARAM_POINTER is not used in the SWS.
- Xcp @ R4.3.0 has both XCP_E_PARAM_POINTER (this is used) and XCP_E_INV_POINTER, but XCP_E_INV_POINTER is not used in the SWS. (changed from R4.2.2)

Remarks:

Even though there're two similar errors, <MA>_E_INV_POINTER and <MA>_E_PARAM_POINTER, I think there's no strong need to replace one of them with other. It's sufficient if this "undefined" issue was resolved, I think.

–Last change on issue 76524 comment 6–

Agreed solution:

SWS EthIf

=====

replace ETHIF_E_INV_POINTER by ETHIF_E_PARAM_POINTER

SWS EthTrcv

=====

replace ETHTRCV_E_INV_POINTER by ETHTRCV_E_PARAM_POINTER

SWS Xcp

=====

remove XCP_E_INV_POINTER from the table [SWS_Xcp_00857]

–Last change on issue 76524 comment 8–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.23 Specification Item SWS_EthTrcv_00137

Trace References:

none

Content:

If **default development** error detection is enabled: The function EthTrcv_CheckWakeup() shall check that the service EthTrcv_Init was previously called. If the check fails, the func-

tion shall raise the **default development** error ETHTRCV_E_ **NOT_INITIALIZED UNINIT** otherwise (if DET is disabled) return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors
–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API
UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.24 Specification Item SWS_EthTrcv_00142

Trace References:

none

Content:

If **default development** error detection is enabled: The function EthTrcv_MainFunction() shall check that the service EthTrcv_Init was previously called. If the check fails, the function shall raise the **default development** error ETHTRCV_E_NOT_INITIALIZEDUNINIT.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors
–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.25 Specification Item SWS_EthTrcv_91001

Trace References:

SRS_Eth_00117

Content:

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

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77676: [EthIf][EthSwt][EthTrcv] Complete handling to determine signal quality

Problem description:

AR4.3.0 introduce APIs EthSwt_GetPortSignalQuality and EthTrcv_GetPhySignalQuality. This API's return the result of the signal quality measured by a dedicated Ethernet transceiver. But there is a lack of how the measurement for the signal quality shall be triggered respectively handled.

Agreed solution:

=== EthTrcv ===

~[SWS_EthTrcv_91001] EthTrcv_GetPhySignalQuality
~ uint8* SignalQualityPtr change to uint32* SignalQualityPtr
~ change the description of parameter SignalQualityPtr to "Pointer to the memory where the signal quality shall be stored."

=== EthSwt ===

~[SWS_EthSwt_91014] EthSwt_GetPortSignalQuality
~ uint8* SignalQualityPtr change to uint32* SignalQualityPtr
~ change the description of parameter SignalQualityPtr to "Pointer to the memory where the signal quality shall be stored."
~ change the description: "...If no transceiver is referenced the signal quality shall be set to 0xFFFFFFFF."
~[SWS_EthSwt_00293] change "0xFF" to "0xFFFFFFFF"

=== EthIf ===

~ch.8.2

+ [EthIf_XXXXX1] EthIf_SignalQualityResultType
Type Structure

Element uint32 HighestSignalQuality the highest signal quality of a link since last clear

Element uint32 LowestSignalQuality the lowest link signal quality of a link since last clear

Element uint32 ActualSignalQuality the actual signal quality

~ch. 8.3

~ [SWS_EthIf_91019] set EthIf_GetPhySignalQuality to "deprecated"

+ [EthIf_XXXXX2] EthIf_GetTrcvSignalQuality

Syntax: Std_ReturnType EthIf_GetTrcvSignalQuality(
uint8 TrcvIdx,

EthIf_SignalQualityResultType* ResultPtr)

Sync/Async: Synchronous

Reentrancy: Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.

Parameters (in): TrcvIdx Index of the transceiver within the context of the Ethernet Interface

Parameters (inout): None

Parameters (out): ResultPtr Pointer to the memory where the signal quality in percent shall be stored.

Return value: Std_ReturnType

E_OK: The signal quality retrieved successfully

E_NOT_OK: The signal quality not retrieved successfully

Description: Retrieves the signal quality of the link of the given Ethernet transceiver

+ [EthIf_xxxxx3] EthIf_GetSwitchPortSignalQuality

Syntax: Std_ReturnType EthIf_GetSwitchPortSignalQuality(

uint8 SwitchIdx,

uint8 SwitchPortIdx,

EthIf_SignalQualityResultType* ResultPtr)

Sync/Async: Synchronous

Reentrancy: Reentrant for different Ethernet switch indexes and Ethernet Switch port indexes. Non reentrant for the same SwitchPortIdx.

Parameters (in): SwitchIdx Index of the Ethernet switch within the context of the Ethernet Interface

SwitchPortIdx Index of the Ethernet switch port within the context of the Ethernet Interface

Parameters (inout): None

Parameters (out): ResultPtr Pointer to the memory where the signal quality in percent shall be stored.

Return value: Std_ReturnType

E_OK: The signal quality retrieved successfully

E_NOT_OK: The signal quality not retrieved successfully

Description: Retrieves the signal quality of the link of the given Ethernet switch port

+ [EthIf_xxxxx4] EthIf_ClearTrcvSignalQuality

Syntax: Std_ReturnType EthIf_ClearTrcvSignalQuality(

uint8 TrcvIdx)

Sync/Async: Synchronous

Reentrancy: Reentrant for different TrcvIdx. Non reentrant for the same TrcvIdx.

Parameters (in): TrcvIdx Index of the transceiver within the context of the Ethernet Interface

Parameters (inout): None

Parameters (out): None

Return value: Std_ReturnType

E_OK: The signal quality cleared successfully

E_NOT_OK: The signal quality cleared not successfully

Description: Clear the stored signal quality of the link of the given Ethernet transceiver

+**[EthIf_XXXXX5]** EthIf_ClearSwitchPortSignalQuality

Syntax: Std_ReturnType EthIf_ClearSwitchPortSignalQuality(

uint8 SwitchIdx,

uint8 SwitchPortIdx)

Sync/Async: Synchronous

Reentrancy: Reentrant for different Ethernet switch indexes and Ethernet Switch port indexes. Non reentrant for the same SwitchPortIdx.

Parameters (in): SwitchIdx Index of the Ethernet switch within the context of the Ethernet Interface

SwitchPortIdx Index of the Ethernet switch port within the context of the Ethernet Interface

Parameters (inout): None

Parameters (out): None

Return value: Std_ReturnType

E_OK: The signal quality cleared successfully

E_NOT_OK: The signal quality cleared not successfully

Description: Clear the stored signal quality of the link of the given Ethernet switch port

~ch. 8.5 Scheduled functions

~ SWS_EthIf_91104 EthIf_MainFunctionState

~Description: The function is polling different communication hardware (Ethernet transceiver, Ethernet switch ports) related information, e.g. link state, signal quality.

+**[EthIf_XXXX12]** The EthIf_MainFunctionState shall poll Ethernet communication hardware related information with the period of EthIfMainFunctionStatePeriod.

+**[EthIf_XXXXX6]** For each Ethernet switch port where a link state of ETH-SWT_LINK_STATE_ACTIVE is yielded and references an Ethernet Transceiver the function shall poll the signal quality by calling EthSwt_GetPortSignalQuality().

+**[EthIf_XXXXX7]** For each Ethernet transceiver where a link state of ETHTRCV_LINK_STATE_ACTIVE is yielded the function shall poll the signal quality by calling EthTrcv_GetPhySignalQuality().

+**[EthIf_XXXXX9]** The obtained signal quality value shall be stored as type of EthIf_SignalQualityResultType. The value shall always be stored as ActualSig-

nalQuality. If the obtained signal quality is higher than the stored highest signal quality (HighestSignalQuality), then HighestSignalQuality shall be updated with the obtained signal quality. If the obtained signal quality is lower than the lowest signal quality (LowestSignalQuality), then LowestSignalQuality shall be updated with the obtained signal quality.

~ch. 10 Configuration specification

add the following parameters to section 10.1.2 EthIfGeneral

+ [ECUC_EthIf_XXXXX1] EthIfEnableSignalQualityApi

+ Description: enable/disable the APIs read and clear the signal quality

+ Multiplicity : 1

+ Type : EcucBooleanParamDef

+ Scope / Dependency : scope: local

+ [ECUC_EthIf_XXXXX2] EthIfSignalQualityCheckPeriod

+ Description : Specifies the period in units of seconds in which the signal quality it polled in the context of EthIf_MainfunctionState. The value shall be an integral multiple of EthIfMainFunctionStatePeriod.

+ Multiplicity : 0..1

+ Type : EcucFloatParamDef

+ Scope / Dependency : scope: local

dependency: If this parameter is defined, the EthIf_MainFunctionState shall be generated and parameter EthIfEnableSignalQualityApi shall be set to TRUE.

—Last change on issue 77676 comment 38—

BW-C-Level:

Application	Specification	Bus
1	4	1