Document Title	SWS_TimeSyncOverEthernet: 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_TimeSyncOverEthernet

1.1 Specification Item ECUC_EthTSyn_00049

Trace References:

none

Content:

Name	EthTSynRxCrcValidatedEthTSynGlobalTimeSlave.EthTSynRxCrcValidated		
Parent Container	EthTSynGlobalTimeSlave		
Description	Definition of whether or not validation of the CRC is supportedtakes place.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	CRC_IGNOREDEthTSyn GlobalTimeSlave.EthTSynRx Crc Validated.CRC_IGNORED	EthTSyn accepts all defined Follow_Up.TLVAUTOSAR.Sub-TLV.Type values. The CRC .will be ignored. ignores any CRC inside the Sub-TLVs.	
	CRC_NOT_VALIDATEDEth TSynGlobalTimeSlave.Eth TSynRxCrcValidated.CRC_NOT_VALIDATED	If EthTSynaccepts a MessageCompliance is set to FALSE: Eth TSyn discards Follow_Up .TLVAUTOSAR.Sub-TLV.Type equal to 0x51, 0x61 and 0x34 without validating the CRC. All other Follow_Up.TLVAUTOSAR.Sub-TLV.Type are ignored. messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60.	
	CRC_OPTIONALEthTSyn GlobalTimeSlave.EthTSynRx Crc Validated.CRC_OPTIONAL	If EthTSynaccepts all defined MessageCompliance is set to FALSE: EthTSyn discards Follow_Up .TLVAUTOSARmessages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an incorrect CRC value. Sub-TLV.Type values. The CRC of the Follow_Up message TLV shall be either validated or not validated.	
	CRC_VALIDATEDEthTSyn GlobalTimeSlave.EthTSynRx Crc Validated.CRC_VALIDATED	If EthTSynaccepts a MessageCompliance is set to FALSE: Eth TSyn discards Follow_Up .TLVAUTOSAR.Sub-TLV.Type equal to messages with Sub-TLVs of Type 0x28, 0x50, 0x44, 0x50 or 0x60and 0x44 with correct CRC values. All other , that contain an incorrect CRC value. EthTSyn rejects Follow_Up .TLVAUTOSAR.Sub-TLV.Type are ignoredmessages with Sub-TLVs of Type 0x34, 0x51 or 0x61.	
Post-Build Variant Value	true		
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 #77619: [EthTSyn] Clarification of handling of unexpected Sub-TLVs

  Problem description:

Clarification of handling of unexpected Sub-TLVs

Agreed solution:

Change SWS Item ECUC_EthTSyn_00049 to
ECUC_EthTSyn_00049 :
Name EthTSynRxCrcValidated
Description Definition of whether or not validation of the CRC takes place.
Multiplicity 1
Type EcucEnumerationParamDef
Range CRC_IGNORED EthTSyn ignores any CRC inside the Sub-TLVs.
CRC_NOT_VALIDATED If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60.
CRC_OPTIONAL If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an
incorrect CRC value.
CRC_VALIDATED If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an
incorrect CRC value.
EthTSyn rejects Follow_Up messages with Sub-TLVs of Type 0x34, 0x51 or 0x61.
Change in [SWS_EthTSyn_00157]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00113]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00114]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00115]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00116]
"is set to CRC_VALIDATED"

to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00117]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"
–Last change on issue 77619 comment 12–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.2 Specification Item ECUC_EthTSyn_00068

Trace References:

none

Content:

Container Name	EthTSynPdelayConfigEthTSynPdelayConfig		
Description	Configuration of cyclic propagation delay measurement.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
EthTSynGlobalTimePdelayRespEnable	ECUC_EthTSyn_00069
EthTSynGlobalTimePropagationDelay	ECUC_EthTSyn_00070
EthTSynGlobalTimeTxPdelayReqPeriod	ECUC_EthTSyn_00071
EthTSynPdelayLatencyThreshold	ECUC_EthTSyn_00076
EthTSynPdelayRespAndRespFollowUpTimeout	ECUC_EthTSyn_00074

Included containers:

No Included Containers

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76603: Removal of fixed Pdelay latency threshold

  Problem description:

  [SWS_EthTSyn_00154] currently defines that
  "If the Pdelay latency calculation (Pdelay is not statically defined) exceeds 10μs, the measured value shall be discarded and the previous value shall be kept."

  The value of 10μs was arbitrarily chosen due to typical link delays of Ethernet PHYs and cable delays. Using
  a fixed value instead of making it configurable should avoid
  adding another configuration parameter to the already existing ones.

  However there are Ethernet links with a regular link delay greater than 10μs, especially HDBaseT.

  Proposal: either remove this SWS item or make the Pdelay latency threshold configurable.

  Agreed solution:

  change [SWS_EthTSyn_00154] to
  If EthTSynGlobalTimeTxPdelayReqPeriod is not equal to 0 and if the Pdelay latency calculation result exceeds EthTSynPdelayLatencyThreshold, the measured value shall be discarded and the previous value shall be kept.

  add a parameter EthTSynPdelayConfig:EthTSynPdelayLatencyThreshold
  Description: Threshold for calculated Pdelay. If a measured Pdelay exceeds EthTSynPdelayLatencyThreshold, this value is discarded.
  Unit:seconds
  Multiplicity 0..1
  Type EcucFloatParamDef
  Range]0 .. INF[
  Default 0.00001
  Post-Build Variant Value true
  Value Configuration Class pre-compile time for variant pre-compile link-time for variant link-time and post-build for variant post-build
  scope:local
  –Last change on issue 76603 comment 13–

  BW-C-Level:

Application	Specification	Bus
1	1	1

1.3 Specification Item ECUC_EthTSyn_00072

Trace References:

none

Content:

Name	EthTSynIsSystemWideGlobalTimeMasterEthTSynGlobalTimeMaster.EthTSynIsSystemWideGlobalTimeMaster		
Description	This represents the configuration whether or not the global time master represents the root of a tree of global time domains. It is possible that several global time masters exist that have set this parameter set to true because the global time masters exist once per global time domain and one ECU may start several global time domains on different busses it is connected to. Tags: atp.Status=obsolete atp.StatusRevisionBegin=4.3.1		
Multiplicity	1 0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	true		
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 #77435: [EthTSyn] Clarification for configuration parameter EthTSynIsSystemWideGlobalTimeMaster

  Problem description:

  What is the usecase for the configuration parameter EthTSynIsSystemWideGlobalTimeMaster?
  There is no SWS point what should be done when this parameter is set as TRUE or FALSE.

  In RFC 66490 this parameter was removed

  Agreed solution:

  obsolete ECUC_EthTSyn_00072 EthTSynIsSystemWideGlobalTimeMaster again.
  –Last change on issue 77435 comment 7–

  BW-C-Level:

Application	Specification	Bus
1	1	1

1.4 Specification Item ECUC_EthTSyn_00076

Trace References:

none

Content:

Name	EthTSynPdelayLatencyThresholdEthTSynPdelayConfig.EthTSynPdelayLatencyThreshold		
Description	Threshold for calculated Pdelay. If a measured Pdelay exceeds EthTSynPdelayLatency Threshold, this value is discarded. Unit: seconds		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	]0 .. INF[		
Default value	1E-5		
Post-Build Variant Value	true		
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 #76603: Removal of fixed Pdelay latency threshold

  Problem description:

  [SWS_EthTSyn_00154] currently defines that
  "If the Pdelay latency calculation (Pdelay is not statically defined) exceeds 10μs, the measured value shall be discarded and the previous value shall be kept."

  The value of 10μs was arbitrarily chosen due to typical link delays of Ethernet PHYs and cable delays. Using
  a fixed value instead of making it configurable should avoid
  adding another configuration parameter to the already existing ones.

  However there are Ethernet links with a regular link delay greater than 10μs, especially HDBaseT.

  Proposal: either remove this SWS item or make the Pdelay latency threshold configurable.

Agreed solution:

change [SWS_EthTSyn_00154] to
If EthTSynGlobalTimeTxPdelayReqPeriod is not equal to 0 and if the Pdelay latency calculation result exceeds EthTSynPdelayLatencyThreshold, the measured value shall be discarded and the previous value shall be kept.

add a parameter EthTSynPdelayConfig:EthTSynPdelayLatencyThreshold
Description: Threshold for calculated Pdelay. If a measured Pdelay exceeds EthTSynPdelayLatencyThreshold, this value is discarded.
Unit:seconds
Multiplicity 0..1
Type EcucFloatParamDef
Range]0 .. INF[
Default 0.00001
Post-Build Variant Value true
Value Configuration Class pre-compile time for variant pre-compile link-time for variant link-time and post-build for variant post-build
scope:local
–Last change on issue 76603 comment 13–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.5 Specification Item SWS_EthTSyn_00007

Trace References:

SRS_BSW_00337

Content:

When DET reporting is enabled (refer EthTSynDevErrorDetect (ECUC_EthTSyn_00002 :)), the EthTSyn module shall call Det_ReportError() with the error code ETHTSYN_E_NOT_INITIALIZED UNINIT when any API other than EthTSyn_GetVersionInfo() or EthTSyn_Init() is called in uninitialized state.

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

1.6 Specification Item SWS_EthTSyn_00030

Trace References:

SRS_BSW_00337, SRS_BSW_00385, SRS_BSW_00323

Content:

EthTSyn shall use following development errors:

Type or error	Related error code	Value [hex]
API service used in un-initialized state	ETHTSYN_E_NOT_INITIALIZED UNINIT	0x20
EthTSyn initialization failed	ETHTSYN_E_INIT_FAILED	0x21
API called with invalid controller index	ETHTSYN_E_CTRL_IDX	0x22
API called with invalid pointer	ETHTSYN_E_PARAM_POINTER	0x23
API called with invalid parameter	ETHTSYN_E_PARAM	0x24

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

1.7 Specification Item SWS_EthTSyn_00040

Trace References:

SRS_StbM_20048

Content:

Service name:	EthTSyn_RxIndicationEthTSyn_RxIndication
Syntax:	void EthTSyn_RxIndication(uint8 CtrlIdx, Eth_FrameType FrameType, boolean IsBroadcast, const uint8* PhysAddrPtr, const uint8* DataPtr, uint16 LenByte)
Service ID[hex]:	0x06
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant

Parameters (in):	CtrlIdxEthTSyn_RxIndication.CtrlIdx	Index of the Ethernet controller
	FrameTypeEthTSyn_RxIndication.FrameType	frame type of received Ethernet frame
	IsBroadcastEthTSyn_RxIndication.IsBroadcast	parameter to indicate a broadcast frame
	PhysAddrPtrEthTSyn_RxIndication.PhysAddrPtr	pointer to Physical source address (MAC address in network byte order) of received Ethernet frame
	DataPtrEthTSyn_RxIndication.DataPtr	Pointer to payload of the received Ethernet frame (i.e. Ethernet header is not provided).
	LenByteEthTSyn_RxIndication.LenByte	Length of received data.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	By this API service the EthTSyn gets an indication and the data of a received frame.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #68035: [diverse] Introduce rules defining which input parameters shall be passed per value and which ones per const reference

  Problem description:

  SWS_BSW_00186 especially states that input pointer parameters shall use the const qualifier (i.e., shall be P2CONST).

  In addition to that there shall be a SWS item that states that input parameters of integral and enum type shall be passed by value whereas input parameters of structure type shall be passed by reference.

  The various transformer SWS documents shall be adapted accordingly.
  –Last change on issue 68035 comment 4–

  Agreed solution:

  BSW UML model
  ————-

  The attachment "Changed Proposal in WP-A meeting" contains a list of changes to the APIs in the model (see column H). Afterwards all related documents (included in impact list) shall update their
  generated artifacts.

General Requirements on Basic Software Modules
~~~~~~

Introduce the following requirements prior to SRS_BSW_00371:

SRS_BSW_xxxxx: Input parameters of scalar and enum types shall be passed as a value.
Type: valid
Description: All input parameters of scalar or enum type shall be passed as a value.
Rationale:

Use case: For example a function named <Mip>_SomeFunction with a return type of Std_ReturnType and a single parameter named SomeParameter of type uint8 is defined with the following signature:

Std_ReturnType <Mip>_SomeFunction(uint8 SomeParameter);
Dependencies: –
Supporting Material: —

SRS_BSW_yyyyy: Input parameters of structure type shall be passed as a reference to a constant structure
Type: valid
Description: All input parameters of structure type shall be passed as a reference constant structure
Rationale: Passing input parameters of structure type by value would result in additional run-time overhead due to efforts for copying the whole structure.
Use case: For example a function named <Mip>_SomeFunction with a return type of Std_ReturnType and a single parameter named SomeParameter of type SomeStructure (where SomeStructure is a struct) is defined with the following signature:

Std_ReturnType <Mip>_SomeFunction(P2CONST(SomeStructure, AUTOMATIC, <MIP>_APPL_DATA) SomeParameter);
Dependencies: –
Supporting Material: —

SRS_BSW_zzzzz: Input parameters of array type shall be passed as a reference to the constant array base type
Type: valid
Description: All input parameters of array type shall be passed as a reference to the constant array base type
~~~~~~

Rationale: This effectively matches the behavior specified in the ISO-C:90 namely that a "declaration of a parameter as ’array of type’ shall be adjusted to ’qualified pointer to
type’".
Use case: For example a function named <Mip>_SomeFunction with a return type of Std_ReturnType and a single parameter named SomeParameter of type array of uint8 is defined with the following signature:

Std_ReturnType <Mip>_SomeFunction(P2CONST(uint8, AUTOMATIC, <MIP>_APPL_DATA) SomeParameter);
Dependencies: –
Supporting Material: —

General Specification of Transformers
~~~~~~

In SWS_Xfrm_00036 change

const <type>* dataElement

to

<paramtype> dataElement

and add the following to the where clause after the API table after the bullet "type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

In SWS_Xfrm_00038 change

[<type> data_1,] ...
[<type> data_n]

to

[<paramtype> data_1,] ...
~~~~~~

[<paramtype> data_n]

and add the following to the where clause after the API table after the bullet "type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the transformer as data_1, ..., data_n the requirements to API parameters stated in chapter API Parameters of [5, SWS RTE] are valid (especially [SWS_Rte_01017], [SWS_Rte_01018] and [SWS_Rte_05107]).

In SWS_Xfrm_00040 change

[<originalData1>, ...
<originalDataN>]

to

[<paramtype> originalData1,] ...
[<paramtype> originalDataN]

and add the following to the where clause after the API table after the bullet "type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

In SWS_Xfrm_00044 change

<type> *data_1, ...

<type> *data_n

to

[<paramtype> data_1,] ...
[<paramtype> data_n]

and add the following to the where clause after the API table after the bullet "type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the transformer as data_1, ..., data_n the requirements to API parameters stated in chapter API Parameters of [5, SWS RTE] are valid (especially [SWS_Rte_01017], [SWS_Rte_01018] and [SWS_Rte_05107]).

Speci?cation of SOME/IP Transformer
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

In SWS_SomeIpXf_00138 change

const <type>* dataElement

to

<paramtype> dataElement

and add the following to the where clause after the API table after the bullet "type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

SWS_BSW_00187).

In SWS_SomeIpXf_00141 change

[<type> data_1,] ...
[<type> data_n]

to

[<paramtype> data_1,] ...
[<paramtype> data_n]

and add the following to the where clause after the API table after the bullet "type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the transformer as data_1, ..., data_n the requirements to API parameters stated in chapter API Parameters of [5, SWS RTE] are valid (especially [SWS_Rte_01017], [SWS_Rte_01018] and [SWS_Rte_05107]).

In SWS_SomeIpXf_00145 change

<type> *data_1, ...
<type> *data_n

to

[<paramtype> data_1,] ...
[<paramtype> data_n]

and add the following to the where clause after the API table after the bullet "type is data type of the data element

"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the transformer as data_1, ..., data_n the requirements to API parameters stated in chapter API Parameters of [5, SWS RTE] are valid (especially [SWS_Rte_01017], [SWS_Rte_01018] and [SWS_Rte_05107]).

Specification of COM Based Transformer
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

In SWS_ComXf_00007 change

const <type>* dataElement

to

<paramtype> dataElement

and add the following to the where clause after the API table after the bullet "type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

Specification of Time Sync over Ethernet
~~~~~~

In SWS_EthTSyn_00040 make the parameter DataPtr of EthTSyn_RxIndication const.

Specification of SWS FlexRay Interface
~~~~~~

Change SWS_FrIf_05073 from
FrIf_NumOfStartupFramesPtr (IN)
to
FrIf_NumOfStartupFramesPtr (OUT)

Specification of ADC
~~~~~~
~[SWS_Adc_00419] Adc_SetupResultBuffer: change Adc_ValueGroupType* to const Adc_ValueGroupType*
~[SWS_Adc_00369] Adc_ReadGroup: move Adc_ValueGroupType * from Parameters (in) to Parameters (out)

There is no need to change parameter from IN to INOUT in Adc_SetupResultBuffer

Specification of Com
~~~~~~
Change type of parameter MetaData of Com_TriggerIPDUSendWithMetaData from uint8* to const uint8*

Specification of ComM
~~~~~~
no change required

Specification of Dem
~~~~~~
no change required

Specification of DLT
~~~~~~
no change required

Specification of DoIP
~~~~~~
From:
Std_ReturnType <User>_DoIPRoutingActivationConfirmation(boolean* Confirmed, uint8* ConfirmationReqData, uint8* ConfirmationResData)
Std_ReturnType <User>_DoIPRoutingActivationAuthentication(boolean* Authentified, uint8* AuthenticationReqData, uint8* AuthenticationResData)

To:
Std_ReturnType <User>_DoIPRoutingActivationConfirmation(boolean* Confirmed, const uint8* ConfirmationReqData, uint8* ConfirmationResData)
Std_ReturnType <User>_DoIPRoutingActivationAuthentication(boolean* Authentified, const uint8* AuthenticationReqData, uint8* AuthenticationResData)

Specification of E2ELibrary
~~~~~~
no change required

Specification of Eth
~~~~~~
no change required

Specification of EthIf
~~~~~~
no change required

Specification of EthSwitchDriver
~~~~~~
no change required

Specification of ICUDriver
~~~~~~
SWS_Icu_00201: Icu_StartTimestamp
Parameter (IN): Icu_ValueType* BufferPtr shall be changed to Parameters (out) type

Specification of LdCom
~~~~~~
~~~~~~

[SWS_LDCOM_00027]: LdCom_CopyTxData
BufReq_ReturnType LdCom_CopyTxData(PduIdType id, const PduInfoType* info, RetryInfoType* retry, PduLengthType* availableDataPtr) shall be changed to
BufReq_ReturnType LdCom_CopyTxData(PduIdType id, const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr)

[SWS_LDCOM_00036]: Rte_LdComCbkCopyTxData_<sn>
BufReq_ReturnType Rte_LdComCbkCopyTxData_<sn>(const PduInfoType* info, RetryInfoType* retry, PduLengthType* availableDataPtr) shall be changed to
BufReq_ReturnType Rte_LdComCbkCopyTxData_<sn>(const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr)

Specification of Lin
~~~~~~
PduInfoPtr needs to be const in Std_ReturnType Lin_SendFrame(uint8 Channel, const Lin_PduType* PduInfoPtr)

Specification of PduR
~~~~~~
* PduR_<User:LoTp>CopyTxData
add const to "RetryInfoType* retry"

Specification of J1939Nm
~~~~~~
Change parameter ’name’ of User_AddressClaimedIndication to type ’const uint8*’

Specification of SoAd
~~~~~~
=> everything already fixed with RfC 65633

Specification of SPIHandlerDriver
~~~~~~
==> nothing to change for SWS SPI

Specification of SynchronizedTimeBaseManager
~~~~~~
"StbM not affected. All issues listed in the WP-A attachment have been already

implemented by IT 69124 in context of RfC 65633"

Specification of TcpIp
~~~~~~
~[SWS_TCPIP_00040] TcpIp_DhcpReadOption: change DataPtr from (IN) to (OUT)
~[SWS_TCPIP_00189] TcpIp_DhcpV6ReadOption: change DataPtr from (IN) to (OUT)
=> everything else already fixed with RfC 65633

Specification of TimeSyncOverFlexRay
~~~~~~
"Change SWS_FrTSyn_00064: parameter versioninfo of type Std_VersionInfoType* is marked wrongly as IN. Change to OUT"

Specification of EFX
~~~~~~~~~~~~~~~~~~~~~~
~ [SWS_Efx_00355] Efx_Debounce_u8_u8: Include constant for pointer Input-parameter as like below.
uint8 Efx_Debounce_u8_u8(boolean X, Efx_DebounceState_Type * State, const Efx_DebounceParam_Type * Param, sint32 dT)

~ [SWS_Efx_00376] Efx_MedianSort: The parameter <InType>* Array should be InOut instead of In parameter as like below.
Parameters (in): N Size of an array
Parameters (inout): Array Pointer to an array

~ [SWS_Efx_00309] Efx_RampCheckActivity: Include constant for pointer Input-parameter as like below.
boolean Efx_RampCheckActivity(const Efx_StateRamp_Type* State_cpst)

~ [SWS_Efx_00307] Efx_RampGetSwitchPos: Include constant for pointer Input-parameter as like below.
boolean Efx_RampGetSwitchPos(const Efx_StateRamp_Type* State_cpst)

~ [SWS_Efx_00193] Efx_Array_Average: Include constant for pointer Input-parameter as like below.
<OutType> Efx_Array_Average_<InTypeMn>_<OutTypeMn>(const <InType>* Array, uint16 Count)
~~~~~~~~~~~~~~~~~~~~~~

Specification of MFL
~~~~~~~~~~~~~~~~~~~~~~
~ [SWS_Mfl_00192] Mfl_Debounce_u8_u8: Include constant for pointer Input-parameter as like below.
boolean Mfl_Debounce_u8_u8(boolean X, Mfl_DebounceState_Type* State, const Mfl_DebounceParam_Type* Param, float32 dT)

~ [SWS_Mfl_00266] Mfl_DebounceInit: The parameter Mfl_DebounceState_Type* State should be Out instead of In parameter as like below.
Parameters (in): X Initial value for the input state
Parameters (out): State Pointer to structure for debouncing state variables

~ [SWS_Mfl_00246] Mfl_HystDeltaRight_f32_u8: Include constant for pointer Input-parameter as like below.
boolean Mfl_HystDeltaRight_f32_u8(float32 X, float32 Delta, float32 Rsp, const uint8* State)

~ [SWS_Mfl_00285] Mfl_MedianSort_f32_f32: The parameter Array should be InOut instead of In parameter as like below.
Parameters (in): N Size of an array
Parameters (inout): Array Pointer to an array

~ [SWS_Mfl_00037] Mfl_PT1SetState: The parameter State_cpst should be Out instead of In parameter as like below.
Parameters (in): X1_f32 Initial value for input state
Y1_f32 Initial value for output state
Parameters (out): State_cpst Pointer to internal state structure

~ [SWS_Mfl_00225] Mfl_RampCheckActivity: Include constant for pointer Input-parameter as like below.
boolean Mfl_RampCheckActivity(const Mfl_StateRamp_Type* State_cpst)

~ [SWS_Mfl_00223] Mfl_RampGetSwitchPos: Include constant for pointer Input-parameter as like below.
boolean Mfl_RampGetSwitchPos(const Mfl_StateRamp_Type* State_cpst)
–Last change on issue 68035 comment 135–

BW-C-Level:

Application	Specification	Bus
1	4	1
~~~~~~~~~~~~~~~~~~~~~~

1.8 Specification Item SWS_EthTSyn_00047

Trace References:

SRS_StbM_20048, SRS_StbM_20059

Content:

API function	Description
Crc_CalculateCRC8H2F	This service makes a CRC8 calculation with the Polynomial 0x2F on Crc_Length
Det_ReportError	Service to report development errors.
Det_ReportRuntimeError	Service to report runtime errors. If a callout has been configured then this callout shall be called.
EthIf_EnableEgressTimeStamp	Activates egress time stamping on a dedicated message object. Some HW does store once the egress time stamp marker and some HW needs it always before transmission. There will be no "disable" functionality, due to the fact, that the message type is always "time stamped" by network design.
EthIf_GetCurrentTime	Returns a time value out of the HW registers according to the capability of the HW. Is the HW resolution is lower than the Eth_TimeStampType resolution resp. range, the remaining bits will be filled with 0.
EthIf_GetEgressTimeStamp	Reads back the egress time stamp on a dedicated message object. It must be called within the TxConfirmation() function.
EthIf_GetIngressTimeStamp	Reads back the ingress time stamp on a dedicated message object. It must be called within the RxIndication() function.
EthIf_ProvideTxBuffer	Provides access to a transmit buffer of the specified Ethernet controller.
EthIf_SetSwitchMgmtInfo	Provides additional management information along to an Ethernet frame that requires special treatment within the Switch. It has to be called between EthIf_ProvideTxBuffer() and EthIf_Transmit() of the related frame.
EthIf_SwitchEnableTimeStamping	Activates egress time stamping on a dedicated message object, addressed by CtrlIdx and BufIdx.
EthIf_Transmit	Triggers transmission of a previously filled transmit buffer
StbM_BusSetGlobalTime	Allows the Time Base Provider Modules to forward a new Global Time value to the StbM, which has been received from a bus.
StbM_GetCurrentTime	Returns a time value (Local Time Base derived from Global Time Base) in standard format.
StbM_GetCurrentTimeDiff	Returns the time difference of current time raw that is valid at this time minus given time raw by using a most accurate time sourcethe nanoseconds part of the Virtual Local Time of the referenced Time Base minus the time given by the parameter givenTimeStamp.
StbM_GetCurrentTimeRaw	Returns a time value in raw format from the most accurate time sourcenanosecond part of the Virtual Local Time of the referenced Time Base.
StbM_GetOffset	Allows the Timesync Modules to get the current Offset Time and User Data.

API function	Description
StbM_GetTimeBaseStatus	Returns the detailed status of the Time Base. For Offset Time Bases the status of the Offset Time Base itself and the status of the underlying Synchronized Time Base is returned.
StbM_GetTimeBaseUpdateCounter	Allows the Timesync Modules to detect, whether a Time Base should be transmitted immediately in the subsequent <Bus>TSyn_MainFunction() cycle.

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 cannot 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 exisits, 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 #77248: [StbM]: ’timeBaseId’ IN parameter required in APIs ’StbM_GetCurrentTimeRaw’ and ’StbM_GetCurrentTimeDiff’

  Problem description:

  The syntax for the API: ’StbM_GetCurrentTimeRaw’ as per the requirement ID [SWS_StbM_00205] is "Std_ReturnType StbM_GetCurrentTimeRaw(StbM_TimeStampRawType* timeStampRawPtr)"
  In 4.3.0, Each ’StbMSynchronizedTimeBase’(ECUC_StbM_00003) container may be configured with reference to viz. OS counter, a GPT or a referenced Ethernet controller as per the configuration of the container ’StbMLocalTimeClock’(ECUC_StbM_00047).
  There is No IN parameter for this API to identify the reference(OS counter, a GPT or a referenced Ethernet controller) based on which the current Time Base shall be derived.

  Similarly for the API "StbM_GetCurrentTimeDiff" as per the requirement ID [SWS_StbM_00209]. There is No IN parameter to identify the reference(OS counter, a GPT or a referenced Ethernet controller) based on which the current Time Base shall be derived and the time difference shall be calculated.

Agreed solution:

For SWS StbM:

1.) Add a new IN parameter timeBaseId to the 2 APIs 'StbM_GetCurrentTimeRaw' and 'StbM_GetCurrentTimeDiff' (requirements SWS_StbM_00205 and 209):

Std_ReturnType StbM_GetCurrentTimeRaw(
StbM_SynchronizedTimeBaseType timeBaseId,
StbM_TimeStampRawType* timeStampRawPtr)

Std_ReturnType StbM_GetCurrentTimeDiff(
StbM_SynchronizedTimeBaseType timeBaseId,
StbM_TimeStampRawType givenTimeStamp,
StbM_TimeStampRawType* timeStampDiffPtr)

Add a description "Time Base reference" for that parameter to the 'Parameters (in)' part of the API description of both APIs.
Update the "Description" field of APIs SWS_StbM_00205 and SWS_StbM_00209:
- SWS_StbM_00205 : "Returns nanosecond part of the referenced Time Base."
- SWS_StbM_00209 : "Returns time difference of the nanoseconds part of the referenced Time Base minus the time given by the parameter givenTimeStamp."

2.) Add corresponding development error requirements for parameter checking

[SWS_StbM_00xxx]
If the switch StbMDevErrorDetect (ECUC_StbM_00012 :) is set to TRUE, StbM_GetCurrentTimeRaw() shall report to DET the development error STBM_E_PARAM,
if called with a parameter timeBaseID, which
is referring to Offset time base
is not configured or
is within the reserved value range.
(SRS_BSW_00386, SRS_BSW_00323)

[SWS_StbM_00xxx]
If the switch StbMDevErrorDetect (ECUC_StbM_00012 :) is set to TRUE, StbM_GetCurrentTimeDiff() shall report to DET the development error STBM_E_PARAM,

if called with a parameter timeBaseID, which
is referring to Offset time base
is not configured or
is within the reserved value range.
(SRS_BSW_00386, SRS_BSW_00323)

For SWS CanTSyn:
Update description of 'StbM_GetCurrentTimeRaw' and 'StbM_GetCurrentTimeDiff' in table SWS_CanTSyn_00105
Update sequence diagrams in chapter 9.1 and 9.2 (add "timeBaseId:StbM_SynchronizedTimeBaseType" as new 1st Parameter for StbM_GetCurrentTimeDiff and StbM_GetCurrentTimeRaw)

For SWS EthTSyn:
Update description of 'StbM_GetCurrentTimeRaw' and 'StbM_GetCurrentTimeDiff' in table SWS_EthTSyn_00047
Update sequence diagrams in chapter 9.2 and 9.3 (add "timeBaseId:StbM_SynchronizedTimeBaseType" as new 1st Parameter for StbM_GetCurrentTimeDiff and StbM_GetCurrentTimeRaw)

Change [SWS_StbM_00174] from
StbM_GetCurrentTimeRaw() shall return the nanoseconds part of the referenced Time Base unit (refer [SWS_StbM_00173]).
to
StbM_GetCurrentTimeRaw() shall return the nanoseconds part of the Virtual Local Time of the associated Time Base (refer [SWS_StbM_00173]).

Change [SWS_StbM_00175] from
StbM_GetCurrentTimeDiff() shall return the time difference of the nanoseconds part of the referenced Time Base unit (refer to [SWS_StbM_00173]) minus the time given by the paramerter givenTimeStamp in raw format.
to
StbM_GetCurrentTimeDiff() shall return the time difference of the nanoseconds part of the Virtual Local Time of the associated Time Base (refer to [SWS_StbM_00173]) minus the time given by the parameter givenTimeStamp in raw format.

Change [SWS_StbM_00173] from
For Time Domains 0 to 15 StbM_GetCurrentTime() and StbM_GetCurrentTimeExtended() shall return for the requested Time Domain the current Time Base, the related Status and the User Data. The current Time Base shall be derived from either the referenced OS counter, a GPT or a referenced Ethernet controller (refer to StbMLocalTimeHardware).
to

For Time Domains 0 to 15 StbM_GetCurrentTime() and StbM_GetCurrentTimeExtended() shall return for the requested Time Domain the current time of the Time Base, the related Status and the User Data. The current time of the Time Base shall be derived from the related Virtual Local Time with itself is derived from either the referenced OS counter, a GPT or a referenced Ethernet controller (refer to StbMLocalTimeHardware).
–Last change on issue 77248 comment 27–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.9 Specification Item SWS_EthTSyn_00060

Trace References:

SRS_StbM_20048, SRS_StbM_20059

Content:

Time measurement with Switches supporting the use case "Time Aware Bridge with GTM not as Management CPU" considers the inner Switch delay by a modification of the correctionField as well as Pdelay timestamping for requestReceiptTimestamp and response OriginTimestamp like shown in Figure 16.

If the Follow_Up message contains an AUTOSAR TLV, which contains a Sub-TLV: Time Secured it shall be checked, if the element CRC_Time_Flags contains BitMask 0x04 (i.e., the content of correctionField is CRC protected).

If this bit is set then the validation of the CRC_Time_1 element shall be done as follows:

- The CRC Validation shall be done as specified in section [REF].
- The data elements used for the calculation and thus validation of the CRC shall be applied with the following order:
  1. the value of CRC_Time_Flags
  2. the messageLength inside the Follow_Up Message Header, if the element CRC_Time_Flags contains BitMask 0x01
  3. the correctionField inside the Follow_Up Message Header
  4. the sequenceId inside the Follow_Up Message Header, if the element CRC_Time_Flags contains BitMask 0x10
  5. the DataID (refer to [SWS_EthTSyn_00112])

If the validation fails, the Follow_Up message shall be dropped instead of being forwarded.

If the validation is successful, the correctionField shall be modified and the element CRC_Time_1 inside the Sub-TLV: Time Secured shall be calculated due to the content of the CRC_Time_Flags element acc. to the section below the table in [SWS_EthTSyn_00100].

Figure [REF]: Timestamping sequence for Time Aware Bridge with GTM not as Management CPU

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77983: Clarification of handling FUP messages with Sub-TLV CRCs

  Problem description:

  It is currently unclear how EthTSyn shall react,
  - if CRCs are enabled within Sub-TLVs
  - and the ECU is not the Global Time Master
  - and the ECU contains a switch ("switch ECU"), i.e., has to forward SYNC and FUP messages
  - and receives a FUP message containing a Sub-TLV:Time Secured with an invalid CRC.

  Even if the ECU is configured to ignore Sub-TLV CRCs (e.g., EthTSynRxCrcValidated equals CRC_IGNORED) it needs to be defined what shall happen especially if Sub-TLV:Time Secured contains an invalid CRC and CRC_Time_Flags has the 'correctionField' bit set.
  Since the correctionField is egress port specific adapted in every switch, the resident time will be added to the correctionField and a new (valid!) CRC will be calculated, including the correction field.
  However, if Sub-TLV:Time Secured contains an invalid CRC generating a valid CRC and thus ignoring an error condition is not a viable option.

  Since there may be receivers that ignore the CRC anyway and receivers that check it, just dropping the FUP message is not beneficial.
  I think that updating the correctionField and generating an invalid! CRC could be an option since both kind of receivers would react as configured.

  Note that only Sub-TLV:Time Secured TLVs are subject to CRC regeneration since the other CRCs in the other CRC protected Sub-TLVs are calculated by the GTM and remain untouched.

  See also bug 77619, comment 6

I think that a switch ECU does not need to check the CRCs in other CRC protected Sub-TLVs when forwarding, since it may be that the receiver ignores the CRC anyway (even if the switch ECU is configured to check for correct Sub-TLV CRCs).

Agreed solution:

Change [SWS_EthTSyn_00060] from

Time measurement with Switches supporting the use case Time Aware Bridge with GTM not as Management CPU
considers the inner Switch delay by a modification of the correctionField as well as Pdelay timestamping
for requestReceiptTimestamp and responseOriginTimestamp like shown in Figure 16.
<FIGURE>
Figure 16: Timestamping sequence for Time Aware Bridge with GTM not as Management CPU
to

Time measurement with Switches supporting the use case Time Aware Bridge with GTM not as Management CPU
considers the inner Switch delay by a modification of the correctionField as well as Pdelay timestamping
for requestReceiptTimestamp and responseOriginTimestamp like shown in Figure 16.

If the Follow_Up message contains an AUTOSAR TLV which contains a Sub-TLV: Time Secured it shall be checked
if the element CRC_Time_Flags contains BitMask 0x04 (i.e., the content of correctionField is CRC protected).
If this bit is set then the validation of the CRC_Time_1 element shall be done as follows:
The CRC Validation shall be done as specified in section 7.7.2.3.
The data elements used for the calculation and thus validation of the CRC shall be applied with the following order:
1. the value of CRC_Time_Flags
2. the messageLength inside the Follow_Up Message Header, if the element CRC_Time_Flags contains BitMask 0x01
3. the correctionField inside the Follow_Up Message Header
4. the sequenceId inside the Follow_Up Message Header, if the element CRC_Time_Flags contains BitMask 0x10
5. the DataID (refer to [SWS_EthTSyn_00112])

If the validation fails, the Follow_Up message shall be dropped instead of being forwarded.

If the validation is successful, the correctionField shall be modified and the element CRC_Time_1 inside
the Sub-TLV: Time Secured shall be calculated due to the content of the CRC_Time_Flags element acc. to
the section below the table in [SWS_EthTSyn_00100].

<FIGURE>
Figure 16: Timestamping sequence for Time Aware Bridge with GTM not as Management CPU
–Last change on issue 77983 comment 4–

BW-C-Level:

Application	Specification	Bus
1	4	2

1.10 Specification Item SWS_EthTSyn_00113

Trace References:

SRS_StbM_20061

Content:

If EthTSynRxCrcValidated (ECUC_EthTSyn_00049 :) is set to CRC_VALIDATED or CRC_OPTIONAL, the Time Slave shall validate the CRC for CRC_Time_0 by considering the contents of CRC_Time_Flags itself, the contents of the dependent fields as defined in EthTSynCrcFlagsRxValidated (ECUC_EthTSyn_00050 :) acc. to the rule in the table below and the DataID.

	For CRC_Time_0 verification required contents:	
If EthTSynCrcFlagsRxValidated element is set to TRUE:	Follow_Up Message Header	Follow_Up Message Field
EthTSynCrcMessageLength (ECUC_EthTSyn_00051 :)	n.a.	n.a.
EthTSynCrcDomainNumber (ECUC_EthTSyn_00052 :)	domainNumber	n.a.
EthTSynCrcCorrectionField (ECUC_EthTSyn_00053 :)	n.a.	n.a.

EthTSynCrcSourcePortIdentity (ECUC_EthTSyn_00054 :)	sourcePortIdentity	n.a.
EthTSynCrcSequenceId (ECUC_EthTSyn_00055 :)	n.a.	n.a.
EthTSynCrcPreciseOriginTimestamp (ECUC_EthTSyn_00056 :)	n.a.	preciseOriginTimestamp

The data elements used for the calculation and thus validation of the CRC shall apply the following order:

- the value of CRC_Time_Flags
- the domainNumber inside the Follow_Up Message Header, if EthTSynCrcDomainNumber (ECUC_EthTSyn_00052 :) is set to TRUE
- the sourcePortIdentity inside the Follow_Up Message Header, if EthTSynCrcSourcePortIdentity (ECUC_EthTSyn_00054 :) is set to TRUE
- the preciseOriginTimestamp inside the Follow_Up Message Field, if EthTSynCrcPreciseOriginTimestamp (ECUC_EthTSyn_00056 :) is set to TRUE
- the DataID (refer to [SWS_EthTSyn_00112])

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77619: [EthTSyn] Clarification of handling of unexpected Sub-TLVs

  Problem description:

  Clarification of handling of unexpected Sub-TLVs

  Agreed solution:

  Change SWS Item ECUC_EthTSyn_00049 to
  ECUC_EthTSyn_00049 :
  Name EthTSynRxCrcValidated
  Description Definition of whether or not validation of the CRC takes place.
  Multiplicity 1
  Type EcucEnumerationParamDef
  Range CRC_IGNORED EthTSyn ignores any CRC inside the Sub-TLVs.
  CRC_NOT_VALIDATED If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60.
  CRC_OPTIONAL If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an
  incorrect CRC value.

CRC_VALIDATED If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an
incorrect CRC value.
EthTSyn rejects Follow_Up messages with Sub-TLVs of Type 0x34, 0x51 or 0x61.
Change in [SWS_EthTSyn_00157]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00113]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00114]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00115]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00116]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00117]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"
–Last change on issue 77619 comment 12–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.11 Specification Item SWS_EthTSyn_00114

Trace References:

SRS_StbM_20061

Content:

If EthTSynRxCrcValidated (ECUC_EthTSyn_00049 :) is set to CRC_VALIDATED or CRC_OPTIONAL, the Time Slave shall validate the CRC for CRC_Time_1 by considering the contents of CRC_Time_Flags itself, the contents of the dependent fields as defined in EthTSynCrcFlagsRxValidated (ECUC_EthTSyn_00050 :) acc. to the rule in the table below and the DataID.

	For CRC_Time_1 verification required contents:	
If EthTSynCrcFlagsRxValidated element is set to TRUE:	Follow_Up Message Header	Follow_Up Message Field
EthTSynCrcMessageLength (ECUC_EthTSyn_00051 :)	messageLength	n.a.
EthTSynCrcDomainNumber (ECUC_EthTSyn_00052 :)	n.a.	n.a.
EthTSynCrcCorrectionField (ECUC_EthTSyn_00053 :)	correctionField	n.a.
EthTSynCrcSourcePortIdentity (ECUC_EthTSyn_00054 :)	n.a.	n.a.
EthTSynCrcSequenceId (ECUC_EthTSyn_00055 :)	sequenceId	n.a.
EthTSynCrcPreciseOriginTimestamp (ECUC_EthTSyn_00056 :)	n.a.	n.a.

The data elements used for the calculation and thus validation of the CRC shall apply the following order:

- the value of CRC_Time_Flags
- the messageLength inside the Follow_Up Message Header, if EthTSynCrcMessage Length (ECUC_EthTSyn_00051 :) is set to TRUE
- the correctionField inside the Follow_Up Message Header, if EthTSynCrcCorrection Field (ECUC_EthTSyn_00053 :) is set to TRUE
- the sequenceId inside the Follow_Up Message Header, if EthTSynCrcSequenceId (ECUC_EthTSyn_00055 :) is set to TRUE
- the DataID (refer to [SWS_EthTSyn_00112])

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77619: [EthTSyn] Clarification of handling of unexpected Sub-TLVs

  Problem description:

  Clarification of handling of unexpected Sub-TLVs

  Agreed solution:

  Change SWS Item ECUC_EthTSyn_00049 to
  ECUC_EthTSyn_00049 :
  Name EthTSynRxCrcValidated
  Description Definition of whether or not validation of the CRC takes place.
  Multiplicity 1
  Type EcucEnumerationParamDef
  Range CRC_IGNORED EthTSyn ignores any CRC inside the Sub-TLVs.
  CRC_NOT_VALIDATED If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60.
  CRC_OPTIONAL If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an
  incorrect CRC value.
  CRC_VALIDATED If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an
  incorrect CRC value.
  EthTSyn rejects Follow_Up messages with Sub-TLVs of Type 0x34, 0x51 or 0x61.
  Change in [SWS_EthTSyn_00157]
  "is set to CRC_VALIDATED"
  to
  "is set to CRC_VALIDATED or CRC_OPTIONAL"

  Change in [SWS_EthTSyn_00113]
  "is set to CRC_VALIDATED"
  to
  "is set to CRC_VALIDATED or CRC_OPTIONAL"

  Change in [SWS_EthTSyn_00114]
  "is set to CRC_VALIDATED"
  to
  "is set to CRC_VALIDATED or CRC_OPTIONAL"

  Change in [SWS_EthTSyn_00115]
  "is set to CRC_VALIDATED"

to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00116]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00117]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"
–Last change on issue 77619 comment 12–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.12 Specification Item SWS_EthTSyn_00115

Trace References:

SRS_StbM_20061

Content:

If EthTSynRxCrcValidated (ECUC_EthTSyn_00049 :) is set to CRC_VALIDATED or CRC_OPTIONAL, the Time Slave shall validate the CRC for CRC_Status by considering the contents of Status and DataID (in this order).

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77619: [EthTSyn] Clarification of handling of unexpected Sub-TLVs

  Problem description:

  Clarification of handling of unexpected Sub-TLVs

  Agreed solution:

  Change SWS Item ECUC_EthTSyn_00049 to
  ECUC_EthTSyn_00049 :
  Name EthTSynRxCrcValidated
  Description Definition of whether or not validation of the CRC takes place.
  Multiplicity 1

Type EcucEnumerationParamDef
Range CRC_IGNORED EthTSyn ignores any CRC inside the Sub-TLVs.
CRC_NOT_VALIDATED If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60.
CRC_OPTIONAL If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an
incorrect CRC value.
CRC_VALIDATED If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an
incorrect CRC value.
EthTSyn rejects Follow_Up messages with Sub-TLVs of Type 0x34, 0x51 or 0x61.
Change in [SWS_EthTSyn_00157]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00113]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00114]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00115]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00116]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00117]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"
–Last change on issue 77619 comment 12–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.13 Specification Item SWS_EthTSyn_00116

Trace References:

SRS_StbM_20061

Content:

If EthTSynRxCrcValidated (ECUC_EthTSyn_00049 :) is set to CRC_VALIDATED or CRC_OPTIONAL, the Time Slave shall validate the CRC for CRC_UserData by considering the contents of UserDataLength, UserByte_0, UserByte_1, UserByte_2 and DataID (in this order).

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77619: [EthTSyn] Clarification of handling of unexpected Sub-TLVs

  Problem description:

  Clarification of handling of unexpected Sub-TLVs

  Agreed solution:

  Change SWS Item ECUC_EthTSyn_00049 to
  ECUC_EthTSyn_00049 :
  Name EthTSynRxCrcValidated
  Description Definition of whether or not validation of the CRC takes place.
  Multiplicity 1
  Type EcucEnumerationParamDef
  Range CRC_IGNORED EthTSyn ignores any CRC inside the Sub-TLVs.
  CRC_NOT_VALIDATED If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60.
  CRC_OPTIONAL If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an
  incorrect CRC value.
  CRC_VALIDATED If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an
  incorrect CRC value.

EthTSyn rejects Follow_Up messages with Sub-TLVs of Type 0x34, 0x51 or 0x61.
Change in [SWS_EthTSyn_00157]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00113]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00114]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00115]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00116]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00117]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"
–Last change on issue 77619 comment 12–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.14 Specification Item SWS_EthTSyn_00117

Trace References:

SRS_StbM_20061, SRS_StbM_20063

Content:

If EthTSynRxCrcValidated (ECUC_EthTSyn_00049 :) is set to CRC_VALIDATED or CRC_OPTIONAL, the Time Slave shall validate the CRC for CRC_OFS by considering the contents of OfsTimeDomain, OfsTimeSec, OfsTimeNSec, Status, UserDataLength, UserByte_0, UserByte_1, UserByte_2 and DataID (in this order).

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77619: [EthTSyn] Clarification of handling of unexpected Sub-TLVs

  Problem description:

  Clarification of handling of unexpected Sub-TLVs

  Agreed solution:

  Change SWS Item ECUC_EthTSyn_00049 to
  ECUC_EthTSyn_00049 :
  Name EthTSynRxCrcValidated
  Description Definition of whether or not validation of the CRC takes place.
  Multiplicity 1
  Type EcucEnumerationParamDef
  Range CRC_IGNORED EthTSyn ignores any CRC inside the Sub-TLVs.
  CRC_NOT_VALIDATED If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60.
  CRC_OPTIONAL If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an
  incorrect CRC value.
  CRC_VALIDATED If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an
  incorrect CRC value.
  EthTSyn rejects Follow_Up messages with Sub-TLVs of Type 0x34, 0x51 or 0x61.
  Change in [SWS_EthTSyn_00157]
  "is set to CRC_VALIDATED"
  to
  "is set to CRC_VALIDATED or CRC_OPTIONAL"

  Change in [SWS_EthTSyn_00113]
  "is set to CRC_VALIDATED"
  to
  "is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00114]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00115]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00116]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00117]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"
–Last change on issue 77619 comment 12–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.15 Specification Item SWS_EthTSyn_00154

Trace References:

SRS_StbM_20048

Content:

If EthTSynGlobalTimeTxPdelayReqPeriod is not equal to 0 and if the Pdelay latency calculation (Pdelay is not statically defined) exceeds 10μsresult exceeds EthTSynPdelayLatencyThreshold, the measured value shall be discarded and the previous value shall be kept.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76603: Removal of fixed Pdelay latency threshold

  Problem description:

[SWS_EthTSyn_00154] currently defines that
"If the Pdelay latency calculation (Pdelay is not statically defined) exceeds 10μs, the measured value shall be discarded and the previous value shall be kept."

The value of 10μs was arbitrarily chosen due to typical link delays of Ethernet PHYs and cable delays. Using
a fixed value instead of making it configurable should avoid
adding another configuration parameter to the already existing ones.

However there are Ethernet links with a regular link delay greater than 10μs, especially HDBaseT.

Proposal: either remove this SWS item or make the Pdelay latency threshold configurable.

Agreed solution:

change [SWS_EthTSyn_00154] to
If EthTSynGlobalTimeTxPdelayReqPeriod is not equal to 0 and if the Pdelay latency calculation result exceeds EthTSynPdelayLatencyThreshold, the measured value shall be discarded and the previous value shall be kept.

add a parameter EthTSynPdelayConfig:EthTSynPdelayLatencyThreshold
Description: Threshold for calculated Pdelay. If a measured Pdelay exceeds EthTSynPdelayLatencyThreshold, this value is discarded.
Unit:seconds
Multiplicity 0..1
Type EcucFloatParamDef
Range]0 .. INF[
Default 0.00001
Post-Build Variant Value true
Value Configuration Class pre-compile time for variant pre-compile link-time for variant link-time and post-build for variant post-build
scope:local
–Last change on issue 76603 comment 13–

BW-C-Level:

Application	**Specification**	**Bus**
1	1	1

1.16 Specification Item SWS_EthTSyn_00157

Trace References:

SRS_StbM_20061

Content:

If EthTSynRxCrcValidated (ECUC_EthTSyn_00049 :) is set to CRC_VALIDATED or CRC_OPTIONAL, the Time Slave shall validate the CRC as defined in EthTSynCrcFlags RxValidated (ECUC_EthTSyn_00050 :) acc. to the following rule:

	Validate if EthTSynCrcFlagsRxValidated element is set to TRUE:	
Element	Follow_Up Message Header	Follow_Up Message Field
EthTSynCrcMessageLength (ECUC_EthTSyn_00051 :)	messageLength	n.a.
EthTSynCrcDomainNumber (ECUC_EthTSyn_00052 :)	domainNumber	n.a.
EthTSynCrcCorrectionField (ECUC_EthTSyn_00053 :)	correctionField	n.a.
EthTSynCrcSourcePortIdentity (ECUC_EthTSyn_00054 :)	sourcePortIdentity	n.a.
EthTSynCrcSequenceId (ECUC_EthTSyn_00055 :)	sequenceId	n.a.
EthTSynCrcPreciseOriginTimestamp (ECUC_EthTSyn_00056 :)	n.a.	preciseOriginTimestamp

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77619: [EthTSyn] Clarification of handling of unexpected Sub-TLVs

  Problem description:

  Clarification of handling of unexpected Sub-TLVs

  Agreed solution:

  Change SWS Item ECUC_EthTSyn_00049 to
  ECUC_EthTSyn_00049 :
  Name EthTSynRxCrcValidated
  Description Definition of whether or not validation of the CRC takes place.
  Multiplicity 1
  Type EcucEnumerationParamDef
  Range CRC_IGNORED EthTSyn ignores any CRC inside the Sub-TLVs.

CRC_NOT_VALIDATED If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60.
CRC_OPTIONAL If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an
incorrect CRC value.
CRC_VALIDATED If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an
incorrect CRC value.
EthTSyn rejects Follow_Up messages with Sub-TLVs of Type 0x34, 0x51 or 0x61.
Change in [SWS_EthTSyn_00157]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00113]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00114]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00115]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00116]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"

Change in [SWS_EthTSyn_00117]
"is set to CRC_VALIDATED"
to
"is set to CRC_VALIDATED or CRC_OPTIONAL"
–Last change on issue 77619 comment 12–

BW-C-Level:

Application	Specification	Bus
1	4	1