

Document Title	SWS_TimeSyncOverCAN: 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_TimeSyncOverCAN	3
1.1	Specification Item SWS_CanTSyn_00005	3
1.2	Specification Item SWS_CanTSyn_00055	4
1.3	Specification Item SWS_CanTSyn_00078	6
1.4	Specification Item SWS_CanTSyn_00085	8
1.5	Specification Item SWS_CanTSyn_00086	11
1.6	Specification Item SWS_CanTSyn_00089	14
1.7	Specification Item SWS_CanTSyn_00100	16
1.8	Specification Item SWS_CanTSyn_00105	18
1.9	Specification Item SWS_CanTSyn_00120	21
1.10	Specification Item SWS_CanTSyn_00135	22
1.11	Specification Item SWS_CanTSyn_00136	23

1 SWS_TimeSyncOverCAN

1.1 Specification Item SWS_CanTSyn_00005

Trace References:

SRS_BSW_00337

Content:

When DET reporting is enabled (see CanTSynDevErrorDetect), the Time Synchronization over CAN shall call Det_ReportError() with the error code CANTSYN_E_NOT_INITIALIZED UNINIT when any API other than CanTSyn_GetVersionInfo() or CanTSyn_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

1.2 Specification Item SWS_CanTSyn_00055

Trace References:

SRS_StbM_20033, SRS_StbM_20035, SRS_StbM_20036, SRS_StbM_20068

Content:

If CanTSynUseExtendedMsgFormat is FALSE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 7 and DataID, **where Byte 2 is applied first, followed by the other bytes in ascending order, and DataID last.**

If CanTSynUseExtendedMsgFormat is TRUE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID for Extended Timesync message formats, **where Byte 2 is applied first, followed by the other bytes in ascending order, and DataID last.**

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77257: [Time Synchronization] Order of elements for CRC calculation is not mentioned in CanTSyn and FrTSyn

Problem description:

Order of elements (byte 2 to 15, Data-ID) for CRC calculation is not mentioned in CanTSyn and FrTSyn specification.

Should the order be:

1. Byte 2 to 15
 2. DataID
- OR
1. DataID
 2. Byte 2 to 15

CRC values are different, for different orders.

Note: The order of elements is mentioned for EthTSyn (SWS_EthTSyn_00100).

Agreed solution:

Replace

[SWS_FrTSyn_00036]

The CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID.

by

[SWS_FrTSyn_00036]

The CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID where Byte 2 is applied first, followed by the other Bytes in ascending order, and DataID last.

Replace

[SWS_FrTSyn_00055]

The CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID.

by

[SWS_FrTSyn_00055]

The CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID where Byte 2 is applied first, followed by the other Bytes in ascending order, and DataID last.

Replace

[SWS_CanTSyn_00055]

If CanTSynUseExtendedMsgFormat is FALSE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 7 and DataID.

If CanTSynUseExtendedMsgFormat is TRUE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID for Extended Timesync message formats.

by

[SWS_CanTSyn_00055]

If CanTSynUseExtendedMsgFormat is FALSE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 7 and DataID where Byte 2 is applied first, followed by the other Bytes

in ascending order, and DataID last.

If CanTSynUseExtendedMsgFormat is TRUE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID for Extended Timesync message formats where Byte 2 is applied first, followed by the other Bytes in ascending order, and DataID last.

Replace

[SWS_CanTSyn_00085]

If CanTSynUseExtendedMsgFormat is FALSE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 7 and DataID.

If CanTSynUseExtendedMsgFormat is TRUE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID for Extended Timesync message formats.

by

[SWS_CanTSyn_00085]

If CanTSynUseExtendedMsgFormat is FALSE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 7 and DataID where Byte 2 is applied first, followed by the other Bytes

in ascending order, and DataID last.

If CanTSynUseExtendedMsgFormat is TRUE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID for Extended Timesync message formats where Byte 2 is applied first, followed by the other Bytes in ascending order, and DataID last.

–Last change on issue 77257 comment 9–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.3 Specification Item SWS_CanTSyn_00078

Trace References:

SRS_StbM_20034, SRS_StbM_20035, SRS_StbM_20036

Content:

The Sequence Counter Jump Width between two consecutive SYNC or two consecutive OFS messages of the same Time Domain shall **always be greater than 0 and** smaller than or equal to CanTSynGlobalTimeSequenceCounterJumpWidth. Otherwise, a Time Slave shall ignore the respective SYNC / OFS message.

The CanTSynGlobalTimeSequenceCounterJumpWidth value 0 is not allowed.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77674: Sequence Counter Jump Width validation is ambiguous

Problem description:

In both specifications there are requirements like

"The Sequence Counter Jump Width between two consecutive SYNC or two consecutive OFS messages of the same Time Domain shall always be smaller than or equal to CanTSynGlobalTimeSequenceCounterJumpWidth. Otherwise a Time Slave shall ignore the respective SYNC / OFS message."

However it is ambiguous what shall be done if the Sequence Counter Jump Width equals 0, i.e. there was no jump.

Since the transmitter of a Time Base shall increment the SC on every transmission by 1, receiving a consecutive Timesync message with the same SC signals that something has probably gone wrong.

One scenario where this could happen indeed is a restart of the Time Master (e.g., after a reset) where the Time Master is most likely not aware of the previously transmitted SC and uses the same SC for its first Timesync message (probably this is SC 0).

In both specifications there is a requirement in the Message Disassembling section that the <Bus>TSyn module shall validate that the "SC matches to the expected value".

This is ambiguous since the expected value would be the previous SC plus 1, and it is in contradiction to the requirements before that allow Sequence Counter Jump Widths that are larger than one but smaller or equal to <Bus>TSynGlobalTimeSequenceCounterJumpWidth.

Note: In Release 4.2.2 the statement was slightly different: "The Sequence Counter Jump Width between two SYNC resp. two OFS messages must be always smaller or equal to CanTSynGlobalTimeSequenceCounterJumpWidth. The value 0 is not allowed."

Here it was ambiguous whether the value 0 was related to the actual Sequence Counter Jump Width or to the configured parameter (in chapter 10 0 was not allowed for the parameter).

Agreed solution:

SWS_CanTSyn:

Reword in SWS_CanTSyn_00078

"The Sequence Counter Jump Width between two consecutive SYNC or two

consecutive OFS messages of the same Time Domain shall always be smaller than or equal to CanTSynGlobalTimeSequenceCounterJumpWidth. ..."

to

"The Sequence Counter Jump Width between two consecutive SYNC or two consecutive OFS messages of the same Time Domain shall be greater than 0 and smaller than or equal to CanTSynGlobalTimeSequenceCounterJumpWidth. ..."

Reword in SWS_CanTSyn_00086

"2. SC matches to the expected value"

to

"2. SC value is within the accepted range (refer to SWS_CanTSyn_00078 and SWS_CanTSyn_00079)"

SWS_FrTSyn:

Reword in SWS_FrTSyn_00048

"The Sequence Counter Jump Width between two consecutive SYNC or two consecutive OFS messages of the same Time Domain shall always be smaller than or equal to FrTSynGlobalTimeSequenceCounterJumpWidth. ..."

to

"The Sequence Counter Jump Width between two consecutive SYNC or two consecutive OFS messages of the same Time Domain shall be greater than 0 and smaller than or equal to FrTSynGlobalTimeSequenceCounterJumpWidth. ..."

Reword in SWS_FrTSyn_00056

"2. SC matches to the expected value"

to

"2. SC value is within the accepted range (refer to SWS_FrTSyn_00048 and SWS_FrTSyn_00049)"

–Last change on issue 77674 comment 2–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.4 Specification Item SWS_CanTSyn_00085

Trace References:

SRS_StbM_20034, SRS_StbM_20035, SRS_StbM_20036, SRS_StbM_20068

Content:

If CanTSynUseExtendedMsgFormat is FALSE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 7 and DataID, **where Byte 2 is applied first, followed by the other Bytes in ascending order, and DataID last.**

If CanTSynUseExtendedMsgFormat is TRUE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID for Extended Timesync message formats, **where Byte 2 is applied first, followed by the other bytes in ascending order, and DataID last.**

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77257: [Time Synchronization] Order of elements for CRC calculation is not mentioned in CanTSyn and FrTSyn

Problem description:

Order of elements (byte 2 to 15, Data-ID) for CRC calculation is not mentioned in CanTSyn and FrTSyn specification.

Should the order be:

1. Byte 2 to 15

2. DataID

OR

1. DataID

2. Byte 2 to 15

CRC values are different, for different orders.

Note: The order of elements is mentioned for EthTSyn (SWS_EthTSyn_00100).

Agreed solution:

Replace

[SWS_FrTSyn_00036]

The CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID.

by

[SWS_FrTSyn_00036]

The CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID where Byte 2 is applied first, followed by the other Bytes in ascending order, and DataID last.

Replace

[SWS_FrTSyn_00055]

The CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID.

by

[SWS_FrTSyn_00055]

The CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID where Byte 2 is applied first, followed by the other Bytes in ascending order, and DataID last.

Replace

[SWS_CanTSyn_00055]

If CanTSynUseExtendedMsgFormat is FALSE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 7 and DataID.

If CanTSynUseExtendedMsgFormat is TRUE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID for Extended Timesync message formats.

by

[SWS_CanTSyn_00055]

If CanTSynUseExtendedMsgFormat is FALSE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 7 and DataID where Byte 2 is applied first, followed by the other Bytes in ascending order, and DataID last.

If CanTSynUseExtendedMsgFormat is TRUE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID for Extended Timesync message formats where Byte 2 is applied first, followed by the other Bytes in ascending order, and DataID last.

Replace

[SWS_CanTSyn_00085]

If CanTSynUseExtendedMsgFormat is FALSE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 7 and DataID.

If CanTSynUseExtendedMsgFormat is TRUE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID for Extended Timesync message formats.

by

[SWS_CanTSyn_00085]

If CanTSynUseExtendedMsgFormat is FALSE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 7 and DataID where Byte 2 is applied first, followed by the other Bytes in ascending order, and DataID last.

If CanTSynUseExtendedMsgFormat is TRUE, the CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID for Extended Timesync message formats where Byte 2 is applied first, followed by the other Bytes

in ascending order, and DataID last.

–Last change on issue 77257 comment 9–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.5 Specification Item SWS_CanTSyn_00086

Trace References:

SRS_StbM_20035, SRS_StbM_20036

Content:

For each received Time Synchronization message the CanTSyn shall validate the message as follows (all conditions must match):

- Type matches depending on the CanTSynRxCrcValidated parameter
- SC matches to the expected value value is within the accepted range (refer to [SWS_CanTSyn_00078] and [SWS_CanTSyn_00079])
- D matches to the defined Time Domain range for each Type
- D matches to one of the configured Time Domains (given by parameter CanTSynGlobalTimeDomainId)
- SyncTimeNSec (FUP / OFNS / Extended OFS only) matches the defined range of StbM_TimeStampType.nanoseconds.
- CRC (including DataID) matches depending on the CanTSynRxCrcValidated parameter

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76896: Deferred review findings for CanTSyn

Problem description:

Name: Jörn Baden

Phone:

Role: Doc Owner

Description/Motivation:

According to comment # 7 in RfC # 75567, the following 3 R4.3.0 Review findings have been deferred to R4.3.1:

- 1.)
Requirements missing for CanTSynVersionInfoApi, CanTSynGlobalTimePduRef, CanTSynGlobalTimeDomainId, CanTSynSynchronizedTimeBaseRef, CanTSynGlobalTimeMasterConfirmationHandleId
- 2.)
optional interface CanIf_Transmit does not have any requirement mentioning the API.
- 3.)
CANTSYN_E_INIT_FAILED is defined but not used.

Agreed solution:

- 1.) Add Requirements for ECUC parameters
 - CanTSynSynchronizedTimeBaseRef:
Add a new requirement: "If CanTSyn calls an API of the StbM, it shall use the Time Base ID of the Time Base referenced via the parameter CanTSynSynchronizedTimeBaseRef of the corresponding Time Domain".
 - CanTSynGlobalTimeDomainId:
In SWS_CanTSyn_00086 change "4. D matches to one of the configured Time Domains" to "4. D matches to one of the configured Time Domains (given by parameter CanTSynGlobalTimeDomainId)".
 - CanTSynGlobalTimePduRef:
See 2.)
 - CanTSynGlobalTimeMasterConfirmationHandleId:
Add in SWS_CanTSyn_00100 after "... - Invalid PDU ID (CANTSYN_E_INVALID_PDUID)": "..., i.e., a PDU ID not configured by parameter CanTSynGlobalTimeMasterConfirmationHandleId"
- 2.) Add a new requirement "A master shall transmit SYNC, FUP, OFS and OFNS messages by calling CanIf_Transmit with the PduId derived via CanTSynGlobalTimePduRef of the corresponding Time Domain".
- 3.) Add a statement beneath SWS_CanTSyn_00093: "CANTSYN_E_INIT_FAILED is reported as specified in [reference to SWS BSW General] by SWS_BSW_00050"
—Last change on issue 76896 comment 10—

BW-C-Level:

Application	Specification	Bus
1	4	1

- RfC #77674: Sequence Counter Jump Width validation is ambiguous

Problem description:

In both specifications there are requirements like

"The Sequence Counter Jump Width between two consecutive SYNC or two consecutive OFS messages of the same Time Domain shall always be smaller than or equal to CanTSynGlobalTimeSequenceCounterJumpWidth. Otherwise a Time Slave shall ignore the respective SYNC / OFS message."

However it is ambiguous what shall be done if the Sequence Counter Jump Width equals 0, i.e. there was no jump.

Since the transmitter of a Time Base shall increment the SC on every transmission by 1, receiving a consecutive Timesync message with the same SC signals that something has probably gone wrong.

One scenario where this could happen indeed is a restart of the Time Master (e.g., after a reset) where the Time Master is most likely not aware of the previously transmitted SC and uses the same SC for its first Timesync message (probably this is SC 0).

In both specifications there is a requirement in the Message Disassembling section that the <Bus>TSyn module shall validate that the "SC matches to the expected value".

This is ambiguous since the expected value would be the previous SC plus 1, and it is in contradiction to the requirements before that allow Sequence Counter Jump Widths that are larger than one but smaller or equal to <Bus>TSynGlobalTimeSequenceCounterJumpWidth.

Note: In Release 4.2.2 the statement was slightly different: "The Sequence Counter Jump Width between two SYNC resp. two OFS messages must be always smaller or equal to CanTSynGlobalTimeSequenceCounterJumpWidth. The value 0 is not allowed."

Here it was ambiguous whether the value 0 was related to the actual Sequence Counter Jump Width or to the configured parameter (in chapter 10 0 was not allowed for the parameter).

Agreed solution:

SWS_CanTSyn:

Reword in SWS_CanTSyn_00078

"The Sequence Counter Jump Width between two consecutive SYNC or two consecutive OFS messages of the same Time Domain shall always be smaller than or equal to CanTSynGlobalTimeSequenceCounterJumpWidth. ..."

to

"The Sequence Counter Jump Width between two consecutive SYNC or two consecutive OFS messages of the same Time Domain shall be greater than 0 and smaller than or equal to CanTSynGlobalTimeSequenceCounterJumpWidth. ..."

Reword in SWS_CanTSyn_00086

"2. SC matches to the expected value"

to

"2. SC value is within the accepted range (refer to SWS_CanTSyn_00078 and SWS_CanTSyn_00079)"

SWS_FrTSyn:

Reword in SWS_FrTSyn_00048

"The Sequence Counter Jump Width between two consecutive SYNC or two consecutive OFS messages of the same Time Domain shall always be smaller than or equal to FrTSynGlobalTimeSequenceCounterJumpWidth. ..."

to

"The Sequence Counter Jump Width between two consecutive SYNC or two consecutive OFS messages of the same Time Domain shall be greater than 0 and smaller than or equal to FrTSynGlobalTimeSequenceCounterJumpWidth. ..."

Reword in SWS_FrTSyn_00056

"2. SC matches to the expected value"

to

"2. SC value is within the accepted range (refer to SWS_FrTSyn_00048 and SWS_FrTSyn_00049)"

–Last change on issue 77674 comment 2–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.6 Specification Item SWS_CanTSyn_00089

Trace References:

SRS_BSW_00385

Content:

CanTSyn shall use the following errors:

Type or error	Related error code	Value [hex]
API service called with wrong PDU or SDU	CANTSYN_E_INVALID_PDUID	0x01
API service used in un-initialized state	CANTSYN_E_NOT_INITIALIZED UNINIT	0x02
A pointer is NULL	CANTSYN_E_NULL_POINTER	0x03
CanTSyn initialization failed	CANTSYN_E_INIT_FAILED	0x04
API called with invalid parameter	CANTSYN_E_PARAM	0x05
Invalid Controller index	CANTSYN_E_INV_CTRL_IDX	0x06

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_CanTSyn_00100

Trace References:

SRS_BSW_00323, SRS_BSW_00337

Content:

The callback function CanTSyn_TxConfirmation() shall inform the DET, if development error detection is enabled (CanTSynDevErrorDetect is set to TRUE) and if the function call has failed because of the following reason:

- Invalid PDU ID (CANTSYN_E_INVALID_PDUID), i.e., a PDU ID not configured by parameter CanTSynGlobalTimeMasterConfirmationHandleId

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76896: Deferred review findings for CanTSyn

Problem description:

Name: Jörn Baden

Phone:

Role: Doc Owner

Description/Motivation:

According to comment # 7 in RfC # 75567, the following 3 R4.3.0 Review findings have been deferred to R4.3.1:

1.)
Requirements missing for CanTSynVersionInfoApi, CanTSynGlobalTimePduRef, CanTSynGlobalTimeDomainId, CanTSynSynchronizedTimeBaseRef, CanTSynGlobalTimeMasterConfirmationHandleId

2.)
optional interface CanIf_Transmit does not have any requirement mentioning the API.

3.)
CANTSYN_E_INIT_FAILED is defined but not used.

Agreed solution:

1.) Add Requirements for ECUC parameters

- CanTSynSynchronizedTimeBaseRef:

Add a new requirement: "If CanTSyn calls an API of the StbM, it shall use the Time Base ID of the Time Base referenced via the parameter CanTSynSynchronizedTimeBaseRef of the corresponding Time Domain".

- CanTSynGlobalTimeDomainId:

In SWS_CanTSyn_00086 change "4. D matches to one of the configured Time Domains" to "4. D matches to one of the configured Time Domains (given by parameter CanTSynGlobalTimeDomainId)"

- CanTSynGlobalTimePduRef:

See 2.)

- CanTSynGlobalTimeMasterConfirmationHandleId:

Add in SWS_CanTSyn_00100 after "... - Invalid PDU ID (CANTSYN_E_INVALID_PDUID)": "..., i.e., a PDU ID not configured by parameter CanTSynGlobalTimeMasterConfirmationHandleId"

2.) Add a new requirement "A master shall transmit SYNC, FUP, OFS and OFNS messages by calling CanIf_Transmit with the PduId derived via CanTSynGlobalTimePduRef of the corresponding Time Domain".

3.) Add a statement beneath SWS_CanTSyn_00093: "CANTSYN_E_INIT_FAILED is reported as specified in [reference to SWS BSW General] by SWS_BSW_00050"
–Last change on issue 76896 comment 10–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.8 Specification Item SWS_CanTSyn_00105

Trace References:

SRS_StbM_20035

Content:

API function	Description
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 source the 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 source nanosecond part of the Virtual Local Time of the referenced Time Base.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77248: [StbM]: 'timeBaseld' 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 'StbMLocalTime-Clock'(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 timeBaseld 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 parameter 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_CanTSyn_00120

Trace References:

SRS_StbM_20031

Content:

CanTSynCyclicMsgResumeTime (ECUC_CanTSyn_00044 :) represents the timeout value of a cyclicMsgResumeCounter that shall be started when either a SYNC or OFS message has been sent immediately, asynchronous to the cyclic Timesync message transmission. **CanTSynCyclicMsgResumeTime Counter** shall be decremented on each invocation of CanTSyn_MainFunction(), if no Timesync PDU is transmitted asynchronously.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77775: Various CanTSyn related editorial/corrective issues in R4.3.0

Problem description:

This RfC collects various editorial and corrective issues within the CanTSyn SWS of R4.3.0

These issues shall be solved for R4.3.1

Agreed solution:

Change in [SWS_CanTSyn_00120]

"CanTSynCyclicMsgResumeTime shall be decremented on each invocation of CanTSyn_MainFunction(),..."

to

"cyclicMsgResumeCounter shall be decremented on each invocation of CanTSyn_MainFunction(),..."

–Last change on issue 77775 comment 1–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.10 Specification Item SWS_CanTSyn_00135

Trace References:

SRS_StbM_20032, SRS_StbM_20038

Content:

If CanTSyn calls an API of the StbM, it shall use the Time Base ID of the Time Base referenced via the parameter CanTSynSynchronizedTimeBaseRef of the corresponding Time Domain.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76896: Deferred review findings for CanTSyn

Problem description:

Name: Jörn Baden

Phone:

Role: Doc Owner

Description/Motivation:

According to comment # 7 in RfC # 75567, the following 3 R4.3.0 Review findings have been deferred to R4.3.1:

1.)

Requirements missing for CanTSynVersionInfoApi, CanTSynGlobalTimePduRef, CanTSynGlobalTimeDomainId, CanTSynSynchronizedTimeBaseRef, CanTSynGlobalTimeMasterConfirmationHandleId

2.)

optional interface CanIf_Transmit does not have any requirement mentioning the API.

3.)

CANTSYN_E_INIT_FAILED is defined but not used.

Agreed solution:

1.) Add Requirements for ECUC parameters

- CanTSynSynchronizedTimeBaseRef:

Add a new requirement: "If CanTSyn calls an API of the StbM, it shall use the Time Base ID of the Time Base referenced via the parameter CanTSynSynchronized-

TimeBaseRef of the corresponding Time Domain".

- CanTSynGlobalTimeDomainId:

In SWS_CanTSyn_00086 change "4. D matches to one of the configured Time Domains" to "4. D matches to one of the configured Time Domains (given by parameter CanTSynGlobalTimeDomainId)"

- CanTSynGlobalTimePduRef:

See 2.)

- CanTSynGlobalTimeMasterConfirmationHandleId:

Add in SWS_CanTSyn_00100 after "... - Invalid PDU ID (CANTSYN_E_INVALID_PDUID)": "..., i.e., a PDU ID not configured by parameter CanTSynGlobalTimeMasterConfirmationHandleId"

2.) Add a new requirement "A master shall transmit SYNC, FUP, OFS and OFNS messages by calling CanIf_Transmit with the PduId derived via CanTSynGlobalTimePduRef of the corresponding Time Domain".

3.) Add a statement beneath SWS_CanTSyn_00093: "CANTSYN_E_INIT_FAILED is reported as specified in [reference to SWS BSW General] by SWS_BSW_00050"
—Last change on issue 76896 comment 10—

BW-C-Level:

Application	Specification	Bus
1	4	1

1.11 Specification Item SWS_CanTSyn_00136

Trace References:

[SRS_StbM_20031](#)

Content:

A master shall transmit SYNC, FUP, OFS and OFNS messages by calling CanIf_Transmit with the PduId derived via CanTSynGlobalTimePduRef of the corresponding Time Domain

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76896: Deferred review findings for CanTSyn

Problem description:

Name: Jörn Baden

Phone:

Role: Doc Owner

Description/Motivation:

According to comment # 7 in RfC # 75567, the following 3 R4.3.0 Review findings have been deferred to R4.3.1:

1.)

Requirements missing for CanTSynVersionInfoApi, CanTSynGlobalTimePduRef, CanTSynGlobalTimeDomainId, CanTSynSynchronizedTimeBaseRef, CanTSynGlobalTimeMasterConfirmationHandleId

2.)

optional interface CanIf_Transmit does not have any requirement mentioning the API.

3.)

CANTSYN_E_INIT_FAILED is defined but not used.

Agreed solution:

1.) Add Requirements for ECUC parameters

- CanTSynSynchronizedTimeBaseRef:

Add a new requirement: "If CanTSyn calls an API of the StbM, it shall use the Time Base ID of the Time Base referenced via the parameter CanTSynSynchronizedTimeBaseRef of the corresponding Time Domain".

- CanTSynGlobalTimeDomainId:

In SWS_CanTSyn_00086 change "4. D matches to one of the configured Time Domains" to "4. D matches to one of the configured Time Domains (given by parameter CanTSynGlobalTimeDomainId)"

- CanTSynGlobalTimePduRef:

See 2.)

- CanTSynGlobalTimeMasterConfirmationHandleId:

Add in SWS_CanTSyn_00100 after "... - Invalid PDU ID (CANTSYN_E_INVALID_PDUID)": "..., i.e., a PDU ID not configured by parameter CanTSynGlobalTimeMasterConfirmationHandleId"

2.) Add a new requirement "A master shall transmit SYNC, FUP, OFS and OFNS messages by calling CanIf_Transmit with the PduId derived via CanTSynGlobalTimePduRef of the corresponding Time Domain".

3.) Add a statement beneath SWS_CanTSyn_00093: "CANTSYN_E_INIT_FAILED

is reported as specified in [reference to SWS BSW General] by SWS_BSW_00050"
–Last change on issue 76896 comment 10–

BW-C-Level:

Application	Specification	Bus
1	4	1