Document Title	SWS_TimeSyncOverFlexRay: 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_TimeSyncOverFlexRay

1.1 Specification Item SWS_FrTSyn_00004

Trace References:

SRS_BSW_00337

Content:

When DET reporting is enabled (see FrTSynDevErrorDetect), the Time Synchronization over FlexRay shall call Det_ReportError() with the error code FRTSYN_E_NOT_INITIALIZED UNINIT when any API other than FrTSyn_GetVersionInfo() or FrTSyn_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_FrTSyn_00028

Trace References:

SRS_StbM_20043

Content:

The transmitter of a Synchronized Time Base (Time Master) shall perform the following steps to distribute the Synchronized Time Base:

- Get currentCycle and currentMacroticks from FlexRay driver via FrIf_GetGlobalTime()
- Retrieve current Synchronized Time Base value as TSYNC via StbM_GetCurrentTime()
- Calculate the (future) time value of the Time Base at the start of the next FlexRay cycle by T0 = TSYNC + (MacroticksPerCycle * (64 - currentCycle) - currentMacroticks) * MacrotickDuration
- Calculate SyncTimeSec (second portion of T0) and SyncTimeNSec (nanosecond portion of T0)

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77345: Various FrTSyn related editorial/corrective issues in R4.3.0

  Problem description:

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

These issues shall be solved for R4.3.1

Agreed solution:

Change in [SWS_FrTSyn_00094]
from
"If the SGW value (FUP and OFS) is set to SyncToSubDomain, the SYNC_TO_GATEWAY bit within timeBaseStatus shall be set to TRUE. Otherwise, it shall be set to FALSE."
to
"If the SGW value (SYNC and OFS) is set to SyncToSubDomain, the SYNC_TO_GATEWAY bit within timeBaseStatus shall be set to TRUE. Otherwise, it shall be set to FALSE."

Change in [SWS_FrTSyn_00028]
from
"1. Get currentCycle and currentMacroticks from FlexRay driver"
to
"1. Get currentCycle and currentMacroticks via FrIf_GetGlobalTime()"

Change in [SWS_FrTSyn_00046]
from
"2. Get currentCycle and currentMacroticks from FlexRay driver"
to
"2. Get currentCycle and currentMacroticks via FrIf_GetGlobalTime()"

Change in [SWS_FrTSyn_00046]
"4. If the currentCycle has passed the retrieved FCNT from the transmitter
side then the previously calculated T1 must be adjusted by the maximum of
the cycle counter to:
T1 = T1 (MacroticksPerCycle * 64 * MacrotickDuration)"
to
"4. If currentCycle is greater or equal than the retrieved FCNT value from the transmitter (Time Master)
then the calculated value T1 shall be subtracted by 64 times the FR cycle duration:
T1 = T1 (MacroticksPerCycle * 64 * MacrotickDuration)"

Change in [SWS_FrTSyn_00037]
"2. Calculate FCNT (for SYNC message)"
to
"2. Copy currentCycle ([SWS_FrTSyn_00028]) to FCNT (for SYNC message)"

Change in [SWS_FrTSyn_00090]
"FrTSynCyclicMsgResumeTime shall be decremented on each invocation of FrTSyn_MainFunction(),..."
to
"cyclicMsgResumeCounter shall be decremented on each invocation of FrTSyn_MainFunction(),..."
–Last change on issue 77345 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.3 Specification Item SWS_FrTSyn_00036

Trace References:

SRS_StbM_20041, SRS_StbM_20043, SRS_StbM_20044

Content:

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.

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 (btye 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

1.4 Specification Item SWS_FrTSyn_00037

Trace References:

SRS_StbM_20043, SRS_StbM_20044

Content:

For each transmission of a Time Synchronization message the FrTSyn module shall assemble the message as follows:

- Calculate SC
- Calculate Copy currentCycle ([SWS_FrTSyn_00028]) to FCNT (for SYNC message)
- Calculate SGW
- Copy all data to the appropriate position within the related message
- Calculate CRC (configuration dependent)

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77345: Various FrTSyn related editorial/corrective issues in R4.3.0

  Problem description:

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

  These issues shall be solved for R4.3.1

  Agreed solution:

  Change in [SWS_FrTSyn_00094]
  from
  "If the SGW value (FUP and OFS) is set to SyncToSubDomain, the SYNC_TO_GATEWAY bit within timeBaseStatus shall be set to TRUE. Otherwise, it shall be set to FALSE."
  to
  "If the SGW value (SYNC and OFS) is set to SyncToSubDomain, the SYNC_TO_GATEWAY bit within timeBaseStatus shall be set to TRUE. Otherwise, it shall be set to FALSE."

  Change in [SWS_FrTSyn_00028]
  from
  "1. Get currentCycle and currentMacroticks from FlexRay driver"
  to
  "1. Get currentCycle and currentMacroticks via FrIf_GetGlobalTime()"

  Change in [SWS_FrTSyn_00046]
  from
  "2. Get currentCycle and currentMacroticks from FlexRay driver"
  to
  "2. Get currentCycle and currentMacroticks via FrIf_GetGlobalTime()"

  Change in [SWS_FrTSyn_00046]
  "4. If the currentCycle has passed the retrieved FCNT from the transmitter
  side then the previously calculated T1 must be adjusted by the maximum of
  the cycle counter to:
  T1 = T1 (MacroticksPerCycle * 64 * MacrotickDuration)"
  to
  "4. If currentCycle is greater or equal than the retrieved FCNT value from the transmitter (Time Master)
  then the calculated value T1 shall be subtracted by 64 times the FR cycle duration:
  T1 = T1 (MacroticksPerCycle * 64 * MacrotickDuration)"

  Change in [SWS_FrTSyn_00037]

"2. Calculate FCNT (for SYNC message)"
to
"2. Copy currentCycle ([SWS_FrTSyn_00028]) to FCNT (for SYNC message)"

Change in [SWS_FrTSyn_00090]
"FrTSynCyclicMsgResumeTime shall be decremented on each invocation of FrTSyn_MainFunction(),..."
to
"cyclicMsgResumeCounter shall be decremented on each invocation of FrTSyn_MainFunction(),..."
–Last change on issue 77345 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.5 Specification Item SWS_FrTSyn_00046

Trace References:

SRS_StbM_20043

Content:

The receiver of a Synchronized Time Base shall perform the following steps to assemble the Synchronized Time Base:

- On SYNC message RX indication (or in the subsequent MainFunction call) store received time value T0 (SyncTimeSec, SyncTimeNSec)
- Get currentCycle and currentMacroticks from FlexRay driver via FrIf_GetGlobalTime()
- Calculate T1 to update the Time Slave's Local Time Base as: T1 = T0 + (((MacroticksPerCycle * currentCycle) + currentMacroticks) * MacrotickDuration)
- If the currentCycle has passed is greater or equal than the retrieved FCNT value from the transmitter side then the previously calculated (Time Master), then the calculated value T1 must be adjusted by the maximum of the cycle counter toshall be subtracted by 64 times the FR cycle duration: T1 = T1 - (MacroticksPerCycle * 64 * MacrotickDuration)

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77345: Various FrTSyn related editorial/corrective issues in R4.3.0

Problem description:

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

These issues shall be solved for R4.3.1

Agreed solution:

Change in [SWS_FrTSyn_00094]
from
"If the SGW value (FUP and OFS) is set to SyncToSubDomain, the SYNC_TO_GATEWAY bit within timeBaseStatus shall be set to TRUE. Otherwise, it shall be set to FALSE."
to
"If the SGW value (SYNC and OFS) is set to SyncToSubDomain, the SYNC_TO_GATEWAY bit within timeBaseStatus shall be set to TRUE. Otherwise, it shall be set to FALSE."

Change in [SWS_FrTSyn_00028]
from
"1. Get currentCycle and currentMacroticks from FlexRay driver"
to
"1. Get currentCycle and currentMacroticks via FrIf_GetGlobalTime()"

Change in [SWS_FrTSyn_00046]
from
"2. Get currentCycle and currentMacroticks from FlexRay driver"
to
"2. Get currentCycle and currentMacroticks via FrIf_GetGlobalTime()"

Change in [SWS_FrTSyn_00046]
"4. If the currentCycle has passed the retrieved FCNT from the transmitter
side then the previously calculated T1 must be adjusted by the maximum of
the cycle counter to:
T1 = T1 (MacroticksPerCycle * 64 * MacrotickDuration)"
to
"4. If currentCycle is greater or equal than the retrieved FCNT value from the transmitter (Time Master)
then the calculated value T1 shall be subtracted by 64 times the FR cycle duration:
T1 = T1 (MacroticksPerCycle * 64 * MacrotickDuration)"

Change in [SWS_FrTSyn_00037]
"2. Calculate FCNT (for SYNC message)"

to
"2. Copy currentCycle ([SWS_FrTSyn_00028]) to FCNT (for SYNC message)"

Change in [SWS_FrTSyn_00090]
"FrTSynCyclicMsgResumeTime shall be decremented on each invocation of FrTSyn_MainFunction(),..."
to
"cyclicMsgResumeCounter shall be decremented on each invocation of FrTSyn_MainFunction(),..."
–Last change on issue 77345 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.6 Specification Item SWS_FrTSyn_00048

Trace References:

SRS_StbM_20042, SRS_StbM_20043, SRS_StbM_20044

Content:

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

The FrTSynGlobalTimeSequenceCounterJumpWidth 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 is 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

1.7 Specification Item SWS_FrTSyn_00055

Trace References:

SRS_StbM_20042, SRS_StbM_20043, SRS_StbM_20044

Content:

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.

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 (btye 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

1.8 Specification Item SWS_FrTSyn_00056

Trace References:

SRS_StbM_20043, SRS_StbM_20044

Content:

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

- Type matches depending on the FrTSynRxCrcValidated parameter
- SC matches to the expected value value is within the accepted range (refer to [SWS_FrTSyn_00048] and [SWS_FrTSyn_00049])
- D matches to the defined Time Domain range for each Type
- D matches to one of the configured Time Domains
- SyncTimeNSec (SYNC message) or OfsTimeNSec (OFS message) matches the defined range of StbM_TimeStampType.nanoseconds.
- CRC (including DataID) matches depending on the FrTSynRxCrcValidated parameter.

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 is 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

1.9 Specification Item SWS_FrTSyn_00059

Trace References:

SRS_BSW_00385

Content:

FrTSyn shall use following development errors:

Type or error	Related error code	Value [hex]
API service called with wrong PDU or SDU.	FRTSYN_E_INVALID_PDUID	0x01
API service used in un-initialized state	FRTSYN_E_NOT_INITIALIZED UNINIT	0x20
A pointer is invalid	FRTSYN_E_NULL_POINTER	0x21
FrTSyn initialization failed	FRTSYN_E_INIT_FAILED	0x22
API called with invalid parameter	FRTSYN_E_PARAM	0x23
Invalid Controller index	FRTSYN_E_INV_CTRL_IDX	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

1.10 Specification Item SWS_FrTSyn_00064

Trace References:

SRS_StbM_20043

Content:

Service name:	FrTSyn_GetVersionInfoFrTSyn_GetVersionInfo
Syntax:	void FrTSyn_GetVersionInfo(Std_VersionInfoType* versioninfo)
Service ID[hex]:	0x02
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant

Parameters (in):	None	
Parameters (inout):	None	
Parameters (out):	versioninfoFrTSyn_GetVersion Info.versioninfo	Pointer to where to store the version information of this module.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Returns the version information of this module.	

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* Authenti-

fied, 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

1.11 Specification Item SWS_FrTSyn_00090

Trace References:

SRS_StbM_20039

Content:

FrTSynCyclicMsgResumeTime (ECUC_FrTSyn_00032 :) 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. FrTSynCycliccyclicMsgResumeTime Counter shall be decremented on each invocation of FrTSyn_MainFunction(), if no Timesync PDU is transmitted asynchronously.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77345: Various FrTSyn related editorial/corrective issues in R4.3.0

  Problem description:

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

  These issues shall be solved for R4.3.1

  Agreed solution:

  Change in [SWS_FrTSyn_00094]
  from
  "If the SGW value (FUP and OFS) is set to SyncToSubDomain, the SYNC_TO_GATEWAY bit within timeBaseStatus shall be set to TRUE. Otherwise, it shall be set to FALSE."
  to
  "If the SGW value (SYNC and OFS) is set to SyncToSubDomain, the SYNC_TO_GATEWAY bit within timeBaseStatus shall be set to TRUE. Otherwise, it shall be set to FALSE."

  Change in [SWS_FrTSyn_00028]
  from
  "1. Get currentCycle and currentMacroticks from FlexRay driver"
  to
  "1. Get currentCycle and currentMacroticks via FrIf_GetGlobalTime()"

  Change in [SWS_FrTSyn_00046]
  from
  "2. Get currentCycle and currentMacroticks from FlexRay driver"
  to
  "2. Get currentCycle and currentMacroticks via FrIf_GetGlobalTime()"

  Change in [SWS_FrTSyn_00046]
  "4. If the currentCycle has passed the retrieved FCNT from the transmitter

side then the previously calculated T1 must be adjusted by the maximum of
the cycle counter to:
T1 = T1 (MacroticksPerCycle * 64 * MacrotickDuration)"
to
"4. If currentCycle is greater or equal than the retrieved FCNT value from the transmitter (Time Master)
then the calculated value T1 shall be subtracted by 64 times the FR cycle duration:
T1 = T1 (MacroticksPerCycle * 64 * MacrotickDuration)"

Change in [SWS_FrTSyn_00037]
"2. Calculate FCNT (for SYNC message)"
to
"2. Copy currentCycle ([SWS_FrTSyn_00028]) to FCNT (for SYNC message)"

Change in [SWS_FrTSyn_00090]
"FrTSynCyclicMsgResumeTime shall be decremented on each invocation of FrTSyn_MainFunction(),..."
to
"cyclicMsgResumeCounter shall be decremented on each invocation of FrTSyn_MainFunction(),..."
–Last change on issue 77345 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.12 Specification Item SWS_FrTSyn_00094

Trace References:

SRS_StbM_20040, SRS_StbM_20042

Content:

If the SGW value (FUP SYNC and OFS) is set to SyncToSubDomain, the SYNC_TO_GATEWAY bit within timeBaseStatus shall be set to TRUE. Otherwise, it shall be set to FALSE.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77345: Various FrTSyn related editorial/corrective issues in R4.3.0

  Problem description:

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

These issues shall be solved for R4.3.1

Agreed solution:

Change in [SWS_FrTSyn_00094]
from
"If the SGW value (FUP and OFS) is set to SyncToSubDomain, the SYNC_TO_GATEWAY bit within timeBaseStatus shall be set to TRUE. Otherwise, it shall be set to FALSE."
to
"If the SGW value (SYNC and OFS) is set to SyncToSubDomain, the SYNC_TO_GATEWAY bit within timeBaseStatus shall be set to TRUE. Otherwise, it shall be set to FALSE."

Change in [SWS_FrTSyn_00028]
from
"1. Get currentCycle and currentMacroticks from FlexRay driver"
to
"1. Get currentCycle and currentMacroticks via FrIf_GetGlobalTime()"

Change in [SWS_FrTSyn_00046]
from
"2. Get currentCycle and currentMacroticks from FlexRay driver"
to
"2. Get currentCycle and currentMacroticks via FrIf_GetGlobalTime()"

Change in [SWS_FrTSyn_00046]
"4. If the currentCycle has passed the retrieved FCNT from the transmitter
side then the previously calculated T1 must be adjusted by the maximum of
the cycle counter to:
T1 = T1 (MacroticksPerCycle * 64 * MacrotickDuration)"
to
"4. If currentCycle is greater or equal than the retrieved FCNT value from the transmitter (Time Master)
then the calculated value T1 shall be subtracted by 64 times the FR cycle duration:
T1 = T1 (MacroticksPerCycle * 64 * MacrotickDuration)"

Change in [SWS_FrTSyn_00037]
"2. Calculate FCNT (for SYNC message)"
to
"2. Copy currentCycle ([SWS_FrTSyn_00028]) to FCNT (for SYNC message)"

Change in [SWS_FrTSyn_00090]
"FrTSynCyclicMsgResumeTime shall be decremented on each invocation of FrTSyn_MainFunction(),..."
to
"cyclicMsgResumeCounter shall be decremented on each invocation of FrTSyn_MainFunction(),..."
–Last change on issue 77345 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1