Document Title	SWS_SynchronizedTimeBaseManager: 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_SynchronizedTimeBaseManager

1.1 Specification Item ECUC_StbM_00041

Trace References:

none

Content:

Name	StbMTimeLeapFutureThresholdStbMSynchronizedTimeBase.StbMTimeLeapFutureThreshold		
Description	This represents the maximum allowed positive difference between the current Local Time Base value and a newly received Global Time Base value and the current Local Time Base value [unit: seconds].		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0 .. INF[		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77346: Difference calculation for StbMTimeLeapFutureThreshold

  Problem description:

  In [SWS_StbM_00182] it is stated that
  "if the time difference between the current and the updated Time Base value exceeds the configured threshold of StbMTimeLeapFutureThreshold" the "TIMELEAP_FUTURE bit within timeBaseStatus of the Time Base" shall be set.

  The statement "time difference between the current and the updated Time Base value" suggests that this means subtracting the updated Time Base value from the current value.
  For a time leap into the future this will result in a negative value.
  The threshold value StbMTimeLeapFutureThreshold however has been defined in [ECUC_StbM_00041] as "This represents the maximum allowed positive difference

between the current Local Time Base value and a newly received Global Time Base value [unit: seconds]."

Agreed solution:

Change SWS_StbM_00182
from:
"..., if the time difference between the current and the updated Time Base value exceeds the configured threshold of StbMTimeLeapFutureThreshold (ECUC_StbM_00041 :),
if at least one Time Base value has been successfully received before..."
to
"..., if the time difference between the updated Time Base and the current value exceeds the configured threshold of StbMTimeLeapFutureThreshold (ECUC_StbM_00041 :),
i.e. TG - TLSync > StbMTimeLeapFutureThreshold, if at least one Time Base value has been successfully received before.

(where:
- TLSync: Value of the local instance of the Time Base before the new value of the Global Time is applied
- TG: Received value of the Global Time)

..."

change SWS_StbM_00305
from:
"... if the time difference between the current and the updated Time Base value exceeds the configured threshold of StbMTimeLeapPastThreshold (ECUC_StbM_00042 :),
if at least one Time Base value has been successfully received before..."
to
"..., if the time difference between the current and the updated Time Base value exceeds the configured threshold of StbMTimeLeapPastThreshold (ECUC_StbM_00042 :),
i.e. TLSync - TG > StbMTimeLeapPastThreshold, if at least one Time Base value has been successfully received before.

(where:
- TLSync = Value of the local instance of the Time Base before the new value of the Global Time is applied
- TG = Received value of the Global Time)

Change Description in ECUC_StbM_00041

from "This represents the maximum allowed positive difference between the current Local Time Base value and a newly received Global Time Base value [unit: seconds]."
to
"This represents the maximum allowed positive difference between a newly received Global Time Base value and the current Local Time Base value [unit: seconds]."
–Last change on issue 77346 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.2 Specification Item ECUC_StbM_00044

Trace References:

none

Content:

Name	StbMMasterRateDeviationMaxStbMSynchronizedTimeBase.StbMMasterRateDeviationMax		
Parent Container	StbMSynchronizedTimeBase		
Description	This attribute describes the maximum allowed absolute value of the rate deviation value to be set by StbM_SetRateCorrection() [unit: secondsppm].		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 32000		
Default value	0		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77174: Various StbM related editorial issues in R4.3.0

  Problem description:

This RfC collects various editorial issues within the
StbM SWS of R4.3.0

These issues shall be solved for R4.3.1

Agreed solution:

In ch. 10.2.5 add StbMLocalTimeClock.emf

In [SWS_StbM_00413]
remove additional '['.

In note after [SWS_StbM_00180]
format the word "StbMStoreTimebaseNonVolatile" as parameter.

In figure 12 after [SWS_StbM_00371]
- replace "Meas. interval" by "StbMRateCorrectionMeasurementDuration"
- add a horizontal bracket about the yellow measurement n+1 and denote it as 'Measurement'
- replace "BusSetGlobalTime" by "StbM_BusSetGlobalTime()"
- move the 3rd BusSetGlobalTime invocation a bit to the right so that it is not aligned with the measurement interval

Add a new requirement before SWS_StbM_00270
SWS_StbM_XXXX: "If any StbM_TimeNotificationCallback is configured, the StbM shall use one additional GPT source (referenced by ECUC_StbM_00039 : StbMGptTimerRef), which is not used for other purposes"

In [SWS_StbM_00196]
correct the word "timeBaseID" to "timeBaseId" and format it as parameter.

In [SWS_StbM_00196]
format the word "timeBaseId" as parameter.

In [SWS_StbM_00201], [SWS_StbM_00214], [SWS_StbM_00340], [SWS_StbM_00219], [SWS_StbM_00224],
[SWS_StbM_00229], [SWS_StbM_00234], [SWS_StbM_00379], [SWS_StbM_00391], [SWS_StbM_00268], [SWS_StbM_00264],
[SWS_StbM_00296], [SWS_StbM_00394], [SWS_StbM_00327], [SWS_StbM_00349], [SWS_StbM_00348]
correct the word "timeBaseID" to "timeBaseId"

Change in [SWS_StbM_00197]
"...STBM_E_PARAM_POINTER, if called with a NULL pointer of parameter timeS-

tamp and userData."
to
"STBM_E_PARAM_POINTER, if called with a NULL pointer of parameter timeStamp or userData."

Change in [SWS_StbM_00202]
"...STBM_E_PARAM_POINTER, if called with a NULL pointer of parameter timeStamp and userData."
to
"STBM_E_PARAM_POINTER, if called with a NULL pointer of parameter timeStamp or userData."

Change in [ECUC_StbM_00044]
"[unit: seconds]"
to
"[unit: ppm]"
–Last change on issue 77174 comment 17–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.3 Specification Item ECUC_StbM_00053

Trace References:

none

Content:

Name	StbMLocalTimeHardwareStbMLocalTimeClock.StbMLocalTimeHardware		
Parent Container	StbMLocalTimeClock		
Description	Reference to the local time hardware.		
Multiplicity	0..1 1		
Type	Choice reference to [EthTSynGlobalTimeDomain , GptChannelConfiguration , OsCounter]		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Scope / Dependency	scope: local

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77794: [StbM]: Query on Multiplicity of the parameter ’StbMLocalTimeHardware’ which is inside the container ’StbMLocalTimeClock’

  Problem description:

  As per the Configuration specifications in section 10: The container ’StbMLocalTimeClock’(ECUC_StbM_00047) is included container of the container ’StbMSynchronizedTimeBase’(ECUC_StbM_00003).
  Multiplicity of the container ’StbMSynchronizedTimeBase’ is ’1..*’ and that of the container ’StbMLocalTimeClock’ is ’0..1’.
  Multiplicity of the parameter ’StbMLocalTimeHardware’(ECUC_StbM_00053) which is inside the container ’StbMLocalTimeClock’ is ’0..1’.

  Is the below understanding is correct as per the above configuration?
  For the ’StbMSynchronizedTimeBaseIdentifier’ (ECUC_StbM_00021) range of ’0 .. 15: Synchronized Time Bases’ and ’32 .. 127: Pure Local Time Bases’, the container ’StbMLocalTimeClock’ and the parameter ’StbMLocalTimeHardware’ has to be configured.
  For the ’StbMSynchronizedTimeBaseIdentifier’ (ECUC_StbM_00021) range of ’16 .. 31: Offset Time Bases’, the container ’StbMLocalTimeClock’ will not be configured.

  If the above understanding is correct, then the query is, in which case the container ’StbMLocalTimeClock’ is configured and the parameter ’StbMLocalTimeHardware’ which is inside this container is not configured?
  whether the parameter ’StbMLocalTimeHardware’ which is inside the container ’StbMLocalTimeClock’ should be made as mandatory parameter, by making it’s Multiplicity as ’1’?

  Agreed solution:

  Make ’StbMLocalTimeHardware’ mandatory Parameter. Change in ECUC_StbM_00053 Multiplicity from ’0 .. 1’ to ’1’
  –Last change on issue 77794 comment 4–

  BW-C-Level:

Application	Specification	Bus
1	4	1

1.4 Specification Item SWS_StbM_00173

Trace References:

SRS_StbM_20003, SRS_StbM_20013, SRS_StbM_20029

Content:

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, which is derived from either the referenced OS counter, a GPT or a referenced Ethernet controller (refer to StbMLocalTimeHardware).

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- 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 "timeBa-

seId: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

1.5 Specification Item SWS_StbM_00174

Trace References:

SRS_StbM_20013, SRS_StbM_20016, SRS_StbM_20021

Content:

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

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- 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

1.6 Specification Item SWS_StbM_00175

Trace References:

SRS_StbM_20016, SRS_StbM_20021, SRS_StbM_20013

Content:

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

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- 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

1.7 Specification Item SWS_StbM_00179

Trace References:

SRS_StbM_20014, SRS_StbM_20025

Content:

Each For Time Domains 0 to 15 each invocation of StbM_BusSetGlobalTime() shall update the corresponding Synchronized Time Base and set the User Data and the Time Base Status accordingly.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76975: [StbM] Clarification on updating Offset Time Base

  Problem description:

  The timebaseId received in the API StbM_BusSetGlobalTime() can be 0..15 for synchronized time base and 16..31 for
  offset time base. And based on the timebaseId, either the synchronized time base or the offset time base
  will be set in the API StbM_BusSetGlobalTime(). Is this understanding is correct?

  As per the requirement:
  [SWS_StbM_00393]
  Each invocation of StbM_BusSetGlobalTime() shall update the corresponding Offset Time Base and set the User Data
  and the Time Base Status accordingly.

  The offset time base need to be also updated even when the In parameter 'timebaseId' is referring to synchronized
  time base?

  Agreed solution:

  Change [SWS_StbM_00393]
  from "Each invocation of StbM_BusSetGlobalTime() shall update the corresponding Offset Time Base and set the User Data and the Time Base Status accordingly."
  to "For Time Domains 16 to 31 each invocation of StbM_BusSetGlobalTime() shall

update the corresponding Offset Time Base and set the User Data and the Time Base Status accordingly."

Change[SWS_StbM_00179]
from "Each invocation of StbM_BusSetGlobalTime() shall update the corresponding Synchronized Time Base and set the User Data and the Time Base Status accordingly."
to "For Time Domains 0 to 15 each invocation of StbM_BusSetGlobalTime() shall update the corresponding Synchronized Time Base and set the User Data and the Time Base Status accordingly."
–Last change on issue 76975 comment 2–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.8 Specification Item SWS_StbM_00181

Trace References:

SRS_StbM_20025

Content:

On a valid invocation of StbM_SetGlobalTime()or , StbM_UpdateGlobalTime(), or StbM_SetOffset() the StbM shall set the GLOBAL_TIME_BASE bit within timeBaseStatus of the corresponding Time Domain Base and shall clear all other bits.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77939: [StbM]: Correction required in ’timeBaseUpdateCounter’ increment for Pure Local Time Base

  Problem description:

  As per the requirement [SWS_StbM_00350], StbM shall increment the ’timeBaseUpdateCounter’ of the corresponding Time Base by 1 (one).

  But as per requirement [SWS_StbM_00347] and [SWS_StbM_00348], the API StbM_GetTimeBaseUpdateCounter() shall report to DET the development error STBM_E_PARAM, if called with a parameter ’timeBaseID’, which refers to a Pure Local Time Base.

  And in section ’2.2.12 Time Master’ as a ’Special Case Pure Local Time Master' it is mentioned that, "A Pure Local Time Master is an entity which is the master of a Pure Local Time Base and which does therefore not propagate this time base to any Time Slave."

So the requirement [SWS_StbM_00350] should also mention that, StbM shall not increment the 'timeBaseUpdateCounter' of the corresponding Time Base by 1 (one) if the 'timeBaseID' refers to Pure Local Time Base.

Is the above understanding is correct?

Agreed solution:

Change [SWS_StbM_00351] by adding the following subclause at the beginning of those requirements:
"For Synchronized and Offset Time Bases, ..."

Change SWS_StbM_00350 to (to also consider StbM_SetOffset) from:
"On a valid invocation of StbM_SetGlobalTime(), StbM_BusSetGlobalTime() or StbM_TriggerTimeTransmission() the StbM shall increment the timeBaseUpdateCounter of the corresponding Time Base by 1 (one). At 255 it shall wrap around to 0."
to
" For Synchronized and Offset Time Bases on a valid invocation of StbM_SetGlobalTime(), StbM_BusSetGlobalTime(), or StbM_TriggerTimeTransmission() and
for Offset Time Bases on a valid invocation of StbM_SetOffset()
the StbM shall increment the timeBaseUpdateCounter of the corresponding Time Base by 1 (one).
At 255 the timeBaseUpdateCounter shall wrap around to 0."

Add a note after SWS_StbM_00350:
Note: For Offset Time Bases the term corresponding Time Base refers to the Offset Time Base only and not to the underlying Synchronized Time Base.

Change SWS_StbM_00304 from
" shall call StbM_SetOffset() with the calculated Offset Time value and the User Data passed by StbM_SetGlobalTime() or StbM_UpdateGlobalTime()."
to
" shall update the corresponding Offset Time Base with the calculated Offset Time value and the User Data that was passed by StbM_SetGlobalTime() or StbM_UpdateGlobalTime()."

Change [SWS_StbM_00181] from
On a valid invocation of StbM_SetGlobalTime()or StbM_UpdateGlobalTime() the

StbM shall set the GLOBAL_TIME_BASE bit within timeBaseStatus of the corresponding Time Domain and shall clear all other bits.
to
On a valid invocation of StbM_SetGlobalTime(), StbM_UpdateGlobalTime(), or StbM_SetOffset() the StbM shall set the GLOBAL_TIME_BASE bit within timeBaseStatus of the corresponding Time Base and shall clear all other bits.

Change the sentence in the introduction of ch. 7.3.6
"Nevertheless, it might be necessary, that the StbM provides an interface,..."
to
"Nevertheless it is necessary, that the StbM provides an interface, ..."
–Last change on issue 77939 comment 9–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.9 Specification Item SWS_StbM_00182

Trace References:

SRS_StbM_20025

Content:

For each Time Domain where a Time Slave or a Time Gateway Slave Port belongs to, an invocation of StbM_BusSetGlobalTime() shall check, if the time difference between the current and the updated updated and the current Time Base value exceeds the configured threshold of StbMTimeLeapFutureThreshold (ECUC_StbM_00041 :), i.e. TG - TLSync > StbMTimeLeapFutureThreshold, if at least one Time Base value has been successfully received before.

With:

- TLSync = Value of the local instance of the Time Base before the new value of the Global Time is applied
- TG = Received value of the Global Time

In case the threshold is exceeded the StbM shall set the TIMELEAP_FUTURE bit within timeBaseStatus of the Time Base.

If the next StbMClearTimeleapCount updates are within the threshold of StbMTimeLeap FutureThreshold the StbM shall clear the TIMELEAP_FUTURE bit within timeBaseStatus of the Time Base.

A threshold of 0 shall deactivate this check.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77346: Difference calculation for StbMTimeLeapFutureThreshold

  Problem description:

  In [SWS_StbM_00182] it is stated that
  "if the time difference between the current and the updated Time Base value exceeds the configured threshold of StbMTimeLeapFutureThreshold" the "TIMELEAP_FUTURE bit within timeBaseStatus of the Time Base" shall be set.

  The statement "time difference between the current and the updated Time Base value" suggests that this means subtracting the updated Time Base value from the current value.
  For a time leap into the future this will result in a negative value.
  The threshold value StbMTimeLeapFutureThreshold however has been defined in [ECUC_StbM_00041] as "This represents the maximum allowed positive difference between the current Local Time Base value and a newly received Global Time Base value [unit: seconds]."

  Agreed solution:

  Change SWS_StbM_00182
  from:
  "..., if the time difference between the current and the updated Time Base value exceeds the configured threshold of StbMTimeLeapFutureThreshold (ECUC_StbM_00041 :),
  if at least one Time Base value has been successfully received before..."
  to
  "..., if the time difference between the updated Time Base and the current value exceeds the configured threshold of StbMTimeLeapFutureThreshold (ECUC_StbM_00041 :),
  i.e. TG - TLSync > StbMTimeLeapFutureThreshold, if at least one Time Base value has been successfully received before.

  (where:
  - TLSync: Value of the local instance of the Time Base before the new value of the Global Time is applied
  - TG: Received value of the Global Time)

  ..."

  change SWS_StbM_00305

from:
"... if the time difference between the current and the updated Time Base value exceeds the configured threshold of StbMTimeLeapPastThreshold (ECUC_StbM_00042 :),
if at least one Time Base value has been successfully received before..."
to
"..., if the time difference between the current and the updated Time Base value exceeds the configured threshold of StbMTimeLeapPastThreshold (ECUC_StbM_00042 :),
i.e. TLSync - TG > StbMTimeLeapPastThreshold, if at least one Time Base value has been successfully received before.

(where:
- TLSync = Value of the local instance of the Time Base before the new value of the Global Time is applied
- TG = Received value of the Global Time)

Change Description in ECUC_StbM_00041
from "This represents the maximum allowed positive difference between the current Local Time Base value and a newly received Global Time Base value [unit: seconds]."
to
"This represents the maximum allowed positive difference between a newly received Global Time Base value and the current Local Time Base value [unit: seconds]."
–Last change on issue 77346 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.10 Specification Item SWS_StbM_00183

Trace References:

SRS_StbM_20007, SRS_StbM_20025

Content:

For each Time Domain where a Time Slave belongs to, the StbM shall observe a timeout. The timeout StbMSyncLossTimeout (ECUC_StbM_00028 :) shall be measured based on the Virtual Local Time from last invocation of StbM_BusSetGlobalTime().

If the timeout occurs, the StbM shall set the TIMEOUT bit within timeBaseStatus of the Time Base.

An invocation of StbM_BusSetGlobalTime() shall clear the TIMEOUT bit.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77671: Various Time Base Status clarifications

  Problem description:

  1.) Both, [SWS_StbM_00183] and [SWS_StbM_00187]
  specify that "If the timeout occurs, the StbM shall set the TIMEOUT bit within timeBaseStatus of the Time Base."

  Since the specified timeout time expires asynchronous
  to any StbM invocation it should be defined when the TIMEOUT bit shall be set.
  Since there is no requirement to set up a dedicated timer outside the StbM which shall call the StbM on expiration there are in principle two occasions where the TIMEOUT bit can be set:
  1.) Inside the cyclically called StbM_MainFunction
  2.) Within every function that returns the Time Base Status, e.g. StbM_GetTimeBaseStatus(), StbM_GetCurrentTime().

  Doing it only in a function that returns the Time Base Status is not sufficient if EV_TIMEOUT_OCCURED was configured as a Status Event.
  Doing it only inside the StbM_MainFunction leads to a slightly deferred TIMEOUT detection (and the points in time at which multiple ECUs will detect a timeout will differ more from each other).

  It is also undefined which time shall be used to determine the timeout (either the Virtual Local Time or the Local Instance of the Synchronized Time Base itself).

  2.) [SWS_StbM_00374] specifies that "For Rate Correction measurements, the StbM shall evaluate state changes of the TIMEOUT flag. The StbM shall discard measurements when the state changes during a measurement."

  This requirement is overspecified and thus confusing since it is sufficient to only discard the measurement if the TIMEOUT flag gets set.
  If a timeout occurs and measurement is stopped, a new measurement is only started by "reception of time values for Synchronized or Offset Time Bases." ([SWS_StbM_00371]), which will then reset the TIMEOUT flag.

3.) [SWS_StbM_00376] specifies that "the StbM shall evaluate state changes of the TIMELEAP_FUTURE and TIMELEAP_PAST flags during measurements. The StbM shall discard the measurement if the flag state changes."

This requirement is overspecified and thus confusing since [SWS_StbM_00373] requires that the StbM shall not start a Rate Correction measurement when either of the flags is set.

Therefore it is sufficient to only discard the measurement if any of the TIMELEAP flags gets set.

Agreed solution:

For 1)
In SWS_StbM_00183 and SWS_StbM_00187 replace the sentence:
The timeout StbMSyncLossTimeout (ECUC_StbM_00028 :) shall be measured from last invocation of StbM_BusSetGlobalTime()
by
The timeout StbMSyncLossTimeout (ECUC_StbM_00028 :) shall be measured based on the Virtual Local Time from last invocation of StbM_BusSetGlobalTime()

Add a new requirement + note after SWS_StbM_00187

SWS_StbM_xxxx: The StbM shall check for a timeout condition of a Time Base within StbM_MainFunction() and all API functions, which return the Time Base Status (e.g. StbM_GetTimeBaseStatus() or StbM_GetCurrentTime())

Note: Since a Status Notification is triggered inside StbM_MainFunction(), the other functions like e.g StbM_GetTimeBaseStatus() might detect a timeout condition sooner than the corresponding Status Notification is actually triggered. Such a delayed Status Notification is considered acceptable.

For 2)
Change SWS_StbM_00374 from:
"..., the StbM shall evaluate state changes of the TIMEOUT flag.
The StbM shall discard measurements when the state changes during a measurement."
to:
"..., the StbM shall evaluate the TIMEOUT flag.
The StbM shall discard the measurement if the flag equals Set."

For 3)

Change SWS_StbM_00376
from:
"..., the StbM shall evaluate state changes of the TIMELEAP_FUTURE and TIMELEAP_PAST flags during measurements.
The StbM shall discard the measurement if the flag state changes."
to:
"..., the StbM shall evaluate the TIMELEAP_FUTURE and TIMELEAP_PAST flags during measurements.
The StbM shall discard the measurement if any of the flags equals Set."

Add a note after SWS_StbM_00362:
"A newly calculated Rate Correction rrc is only applied to following time calculations."
–Last change on issue 77671 comment 14–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.11 Specification Item SWS_StbM_00205

Trace References:

SRS_StbM_20013, SRS_StbM_20016

Content:

Service name:	StbM_GetCurrentTimeRawStbM_GetCurrentTimeRaw
Syntax:	Std_ReturnType StbM_GetCurrentTimeRaw(StbM_TimeSynchronizedTimeBaseType timeBaseId, StbM_TimeStampRawType* timeStampRawPtr)
Service ID[hex]:	0x09
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant

None

Parameters (in):	timeBaseIdStbM_GetCurrentTimeRaw.timeBaseId	Time Base reference
Parameters (inout):	None	
Parameters (out):	timeStampRawPtrStbM_GetCurrentTimeRaw.timeStampRawPtr	Current time stamp that is valid at this time
Return value:	Std_ReturnType	E_OK: successful E_NOT_OK: failed
Description:	Returns a time value in raw format from the most accurate time sourcenanosecond 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]: '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

1.12 Specification Item SWS_StbM_00209

Trace References:

SRS_StbM_20013, SRS_StbM_20016

Content:

Service name:	StbM_GetCurrentTimeDiffStbM_GetCurrentTimeDiff

Syntax:	Std_ReturnType StbM_GetCurrentTimeDiff(StbM_TimeSynchronizedTimeBaseType timeBaseId, StbM_TimeStampRawType givenTimeStamp, StbM_TimeStampRawType* timeStampDiffPtr)	
Service ID[hex]:	0x0a	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	timeBaseIdStbM_GetCurrentTimeDiff.timeBaseId	Time Base reference
	givenTimeStampStbM_GetCurrentTimeDiff.givenTimeStamp	Given time stamp as difference calculation basis
Parameters (inout):	None	
Parameters (out):	timeStampDiffPtrStbM_GetCurrentTimeDiff.timeStampDiffPtr	Time difference of current time stamp that is valid at this time minus given time stamp
Return value:	Std_ReturnType	E_OK: successful E_NOT_OK: failed
Description:	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.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- 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

1.13 Specification Item SWS_StbM_00228

Trace References:

SRS_StbM_20027

Content:

Service name:	StbM_GetOffsetStbM_GetOffset	
Syntax:	Std_ReturnType StbM_GetOffset(StbM_SynchronizedTimeBaseType timeBaseId, StbM_TimeStampType* timeStamp, StbM_UserDataType* userData)	
Service ID[hex]:	0x0e	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	timeBaseIdStbM_GetOffset.timeBaseId	Time Base reference
userDataStbM_GetOffset.userData	Current User Data	
Parameters (inout):	None	
Parameters (out):	timeStampStbM_GetOffset.timeStamp	Current Offset Time value
	userDataStbM_GetOffset.userData	Current User Data
Return value:	Std_ReturnType	E_OK: successful E_NOT_OK: failed
Description:	Allows the Timesync Modules to get the current Offset Time and User Data.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77566: [StbM]: Correction required for the 'StbM_GetOffset' service's parameter

  Problem description:

  As per the requirement [SWS_StbM_00228]: For the "StbM_GetOffset" service, 'userData' parameter is mentioned as 'IN parameter'.

Agreed solution:

Modify [SWS_StbM_00228]: make the 'IN' parameter 'userData' an 'OUT' parameter
–Last change on issue 77566 comment 4–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.14 Specification Item SWS_StbM_00247

Trace References:

SRS_StbM_20003, SRS_StbM_20010, SRS_StbM_20029, SRS_StbM_20056, SRS_StbM_20057

Content:

Name	GlobalTime_Slave_{Name}GlobalTime_Slave	
Comment	–	
IsService	true	
Variation	{ecuc(StbM/StbMSynchronizedTimeBase/StbMSynchronizedTimeBaseIdentifier)} < 128 Name = {ecuc(StbM/StbMSynchronizedTimeBase.SHORT-NAME)}	
Possible Errors	0	E_OK
	1	E_NOT_OK

Operations:

GetCurrentTimeGlobalTime_Slave.GetCurrentTime			
Comments	Returns a time value (Local Time Base derived from Global Time Base) in standard format.		
Variation	–		
Parameters	timeStampGlobalTime_Slave.GetCurrentTime.timeStamp	Comment	–
		Type	StbM_TimeStampType
		Variation	–
		Direction	OUT
	userDataGlobalTime_Slave.GetCurrentTime.userData	Comment	–
		Type	StbM_UserDataType
		Variation	–
		Direction	OUT
Possible Errors	E_OK	Operation successful	
	E_NOT_OK	Operation failed	

GetCurrentTimeExtendedGlobalTime_Slave.GetCurrentTimeExtended			
Comments	Returns a time value (Local Time Base derived from Global Time Base) in extended format.		
Variation	{ecuc(StbM/StbMGeneral/StbMGetCurrentTimeExtendedAvailable)}		
Parameters	timeStampGlobalTime_Slave.GetCurrentTimeExtended.timeStamp	Comment	–
		Type	StbM_TimeStampExtendedType
		Variation	–
		Direction	OUT
	userDataGlobalTime_Slave.GetCurrentTimeExtended.userData	Comment	–
		Type	StbM_UserDataType
		Variation	–
		Direction	OUT
Possible Errors	E_OK	Operation successful	
	E_NOT_OK	Operation failed	

GetOffsetTimeRecordHeadGlobalTime_Slave.GetOffsetTimeRecordHead			
Comments	Reads the header of the table with recorded measurement data belonging to the Offset Time Base		
Variation	{ecuc(StbM/StbMGeneral/StbMTimeRecordingSupport)} == True &&{ecuc(StbM/StbMSynchronizedTimeBase/StbMSynchronizedTimeBaseIdentifier)} > 15 &&{ecuc(StbM/StbMSynchronizedTimeBase/StbMSynchronizedTimeBaseIdentifier)} < 32		
Parameters	offsetRecordTableHeadGlobalTime_Slave.GetOffsetTimeRecordHead.offsetRecordTableHead	Comment	Header of the table
		Type	StbM_OffsetRecordTableHeadType
		Variation	–
		Direction	OUT
Possible Errors	E_OK	Operation successful	
	E_NOT_OK	Operation failed	

GetRateDeviationGlobalTime_Slave.GetRateDeviation			
Comments	Returns value of the current rate deviation of a Time Base		
Variation	–		
Parameters	rateDeviationGlobalTime_Slave.GetRateDeviation.rateDeviation	Comment	Value of the current rate deviation of a Time Base
		Type	StbM_RateDeviationType
		Variation	–
		Direction	OUT
Possible Errors	E_OK	Operation successful	
	E_NOT_OK	Operation failed	

GetSyncTimeRecordHeadGlobalTime_Slave.GetSyncTimeRecordHead	
Comments	Reads the header of the table with recorded measurement data belonging to the Synchronized Time Base

GetSyncTimeRecordHeadGlobalTime_Slave.GetSyncTimeRecordHead			
Variation	({ecuc(StbM/StbMGeneral/StbMTimeRecordingSupport)} == True) &&({ecuc(StbM/StbMSynchronizedTimeBase/StbMSynchronizedTimeBaseIdentifier)} < 16)		
Parameters	syncRecordTableHeadGlobalTime_Slave.GetSyncTimeRecordHead.syncRecordTableHead	Comment	Header of the table
		Type	StbM_SyncRecordTableHeadType
		Variation	–
		Direction	OUT
Possible Errors	E_OK	Record head read successfully.	
	E_NOT_OK	Read access to record head failed.	

GetTimeBaseStatusGlobalTime_Slave.GetTimeBaseStatus			
Comments	Returns detailed status information for a Synchronized Time Base and, if called for an Offset Time Base, for the Offset Time Base and the underlying Synchronized Time Base		
Variation	–		
Parameters	syncTimeBaseStatusGlobalTime_Slave.GetTimeBaseStatus.syncTimeBaseStatus	Comment	Status of the Synchronized Time Base
		Type	StbM_TimeBaseStatusType
		Variation	–
		Direction	OUT
	offsetTimeBaseStatusGlobalTime_Slave.GetTimeBaseStatus.offsetTimeBaseStatus	Comment	Status of the Offset Time Base.
		Type	StbM_TimeBaseStatusType
		Variation	–
		Direction	OUT
Possible Errors	E_OK	Operation successful	
	E_NOT_OK	Operation failed	

GetTimeLeapGlobalTime_Slave.GetTimeLeap			
Comments	Returns value of time leap, if StbMTimeLeapFuture/PastThreshold has been exceeded.		
Variation	{ecuc(StbM/StbMSynchronizedTimeBase/StbMSynchronizedTimeBaseIdentifier)} < 32		
Parameters	timeJumpGlobalTime_Slave.GetTimeLeap.timeJump	Comment	Time leap value
		Type	StbM_TimeDiffType
		Variation	–
		Direction	OUT
Possible Errors	E_OK	Operation successful	
	E_NOT_OK	Operation failed	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #78023: [StbM]: Clarification required on return value of the API 'StbM_GetTimeLeap()'

Problem description:

As per the requirement [SWS_StbM_00267] StbM_GetTimeLeap() returns value of Time Leap, if StbMTimeLeapFuture/PastThreshold is exceeded.

The value of Time Leap returned in 'timeJump' in API StbM_GetTimeLeap() would be the value which is calculated in API StbM_BusSetGlobalTime() only?

And if StbMTimeLeapFuture/PastThreshold is not exceeded then, we are assuming that StbM_GetTimeLeap() shall return E_NOT_OK.

Agreed solution:

Add a new requirement in ch. 7 for clarification after SWS_StbM_00305:

SWS_StbM_00xxx:
For Time Slaves and Time Gateways of Synchronized Time Bases StbM_GetTimeLeap() shall return the time difference between the newly received and the current Time Base value, i.e. TG - TL, which is calculated upon each, except the very first, valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

With
- TL = Current value of the local instance of the Time Base (before newly received Time Base value is applied)
- TG = Newly received Time Base value

For Time Slaves and Time Gateways of Offset Time Bases StbM_GetTimeLeap() shall return the time difference between the newly received and the current Time Base offset value, i.e. TOG - TOL, which is calculated upon each, except the very first, valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

With
- TOL = Current offset value of the local instance of the Time Base (before newly received Time Base offset value is applied)
- TOG = Newly received Time Base offset value

If the calculated time difference exceeds the value range of the timeJump parameter of StbM_GetTimeLeap() the returned time difference shall be limited to either the maximum negative or the maximum positive value of the type of timeJump [SWS_StbM_00300].

StbM_GetTimeLeap() shall return E_NOT_OK until the second valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

In Comments Field for GetTimeLeap Operation in SWS_StbM_00247
remove subclause ", if StbMTimeLeapFuture/PastThreshold has been exceeded"

In Description Field for API StbM_GetTimeLeap in SWS_StbM_00267
remove subclause ", if StbMTimeLeapFuture/PastThreshold is exceeded"

In Description field of [SWS_StbM_00273]
replace "StbM_SetTimer" by "StbM_StartTimer"

In "Parameters (in):" field of [SWS_StbM_00273]
replace
"Actual time value captured, when callback is called by StbM"
by
"Difference time value when callback is called by StbM"

In [SWS_StbM_00271] replace
"actualTime" by "deviationTime"
–Last change on issue 78023 comment 20–

BW-C-Level:

Application	Specification	Bus
4	4	1

1.15 Specification Item SWS_StbM_00267

Trace References:

SRS_StbM_20003

Content:

Service name:	StbM_GetTimeLeapStbM_GetTimeLeap	
Syntax:	Std_ReturnType StbM_GetTimeLeap(StbM_SynchronizedTimeBaseType timeBaseId, StbM_TimeDiffType* timeJump)	
Service ID[hex]:	0x13	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	timeBaseIdStbM_GetTimeLeap.timeBaseId	Time Base reference

Parameters (inout):	None	
Parameters (out):	timeJumpStbM_GetTimeLeap.timeJump	Time leap value
Return value:	Std_ReturnType	E_OK: successful E_NOT_OK: failed
Description:	Returns value of Time Leap, if StbMTimeLeapFuture/PastThreshold is exceeded .	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77874: [StbM] StbM_TimeBaseNotificationType_EV_TIMEOUT_OCCURRED_Mask needs correction

  Problem description:

  In StbM_TimeBaseNotificationType EV_TIMEOUT_OCCURRED has a Mask as 0x01 (reference SWS_StbM_00287)

  Agreed solution:

  In SWS_StbM_00287 (StbM_TimeBaseNotificationType) event EV_TIMEOUT_OCCURRED should have a Mask as 0x02 since it is second bit.
  –Last change on issue 77874 comment 2–

  BW-C-Level:

Application	Specification	Bus
1	4	1

1.16 Specification Item SWS_StbM_00271

Trace References:

SRS_StbM_20056

Content:

Upon invocation of StbM_TimerCallback, the StbM shall calculate the time difference between CustomerTimerExpireTime and the current value of the corresponding Time Base.

The StbM shall then call StbM_TimeNotificationCallback() to inform the corresponding Time Notification Customer and return the value of the calculated time difference by parameter actualdeviationTime.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #78023: [StbM]: Clarification required on return value of the API 'StbM_GetTimeLeap()'

  Problem description:

  As per the requirement [SWS_StbM_00267] StbM_GetTimeLeap() returns value of Time Leap, if StbMTimeLeapFuture/PastThreshold is exceeded.

  The value of Time Leap returned in 'timeJump' in API StbM_GetTimeLeap() would be the value which is calculated in API StbM_BusSetGlobalTime() only?

  And if StbMTimeLeapFuture/PastThreshold is not exceeded then, we are assuming that StbM_GetTimeLeap() shall return E_NOT_OK.

  Agreed solution:

  Add a new requirement in ch. 7 for clarification after SWS_StbM_00305:

  SWS_StbM_00xxx:
  For Time Slaves and Time Gateways of Synchronized Time Bases StbM_GetTimeLeap() shall return the time difference between the newly received and the current Time Base value, i.e. TG - TL, which is calculated upon each, except the very first, valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

  With
  - TL = Current value of the local instance of the Time Base (before newly received Time Base value is applied)
  - TG = Newly received Time Base value

  For Time Slaves and Time Gateways of Offset Time Bases StbM_GetTimeLeap() shall return the time difference between the newly received and the current Time Base offset value, i.e. TOG - TOL, which is calculated upon each, except the very first, valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

  With
  - TOL = Current offset value of the local instance of the Time Base (before newly received Time Base offset value is applied)
  - TOG = Newly received Time Base offset value

  If the calculated time difference exceeds the value range of the timeJump parameter of StbM_GetTimeLeap() the returned time difference shall be limited to either the maximum negative or the maximum positive value of the type of timeJump [SWS_StbM_00300].

StbM_GetTimeLeap() shall return E_NOT_OK until the second valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

In Comments Field for GetTimeLeap Operation in SWS_StbM_00247
remove subclause ", if StbMTimeLeapFuture/PastThreshold has been exceeded"

In Description Field for API StbM_GetTimeLeap in SWS_StbM_00267
remove subclause ", if StbMTimeLeapFuture/PastThreshold is exceeded"

In Description field of [SWS_StbM_00273]
replace "StbM_SetTimer" by "StbM_StartTimer"

In "Parameters (in):" field of [SWS_StbM_00273]
replace
"Actual time value captured, when callback is called by StbM"
by
"Difference time value when callback is called by StbM"

In [SWS_StbM_00271] replace
"actualTime" by "deviationTime"
–Last change on issue 78023 comment 20–

BW-C-Level:

Application	Specification	Bus
4	4	1

1.17 Specification Item SWS_StbM_00273

Trace References:

SRS_StbM_20056, SRS_BSW_00457, SRS_BSW_00360, SRS_BSW_00333

Content:

Service name:	<Customer>_TimeNotificationCallback<TimeBase><Customer>_TimeNotification Callback<TimeBase>
Syntax:	Std_ReturnType <Customer>_TimeNotificationCallback<TimeBase>(StbM_TimeDiffType deviationTime)
Service ID[hex]:	0x18
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant

Parameters (in):	deviationTime<Customer>_TimeNotificationCallback<TimeBase>.deviationTime	Actual time value captured, Difference time value when callback is called by StbM.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: successful E_NOT_OK: failed
Description:	This callback notifies the <Customer>, when a Time Base reaches the time value set by StbM_SetStartTimer for the <TimeBase>	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #78023: [StbM]: Clarification required on return value of the API 'StbM_GetTimeLeap()'

  Problem description:

  As per the requirement [SWS_StbM_00267] StbM_GetTimeLeap() returns value of Time Leap, if StbMTimeLeapFuture/PastThreshold is exceeded.

  The value of Time Leap returned in 'timeJump' in API StbM_GetTimeLeap() would be the value which is calculated in API StbM_BusSetGlobalTime() only?

  And if StbMTimeLeapFuture/PastThreshold is not exceeded then, we are assuming that StbM_GetTimeLeap() shall return E_NOT_OK.

  Agreed solution:

  Add a new requirement in ch. 7 for clarification after SWS_StbM_00305:

  SWS_StbM_00xxx:
  For Time Slaves and Time Gateways of Synchronized Time Bases StbM_GetTimeLeap() shall return the time difference between the newly received and the current Time Base value, i.e. TG - TL, which is calculated upon each, except the very first, valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

  With
  - TL = Current value of the local instance of the Time Base (before newly received Time Base value is applied)
  - TG = Newly received Time Base value

  For Time Slaves and Time Gateways of Offset Time Bases StbM_GetTimeLeap() shall return the time difference between the newly received and the current Time Base offset value, i.e. TOG - TOL, which is calculated upon each, except the very first, valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

With
- TOL = Current offset value of the local instance of the Time Base (before newly received Time Base offset value is applied)
- TOG = Newly received Time Base offset value

If the calculated time difference exceeds the value range of the timeJump parameter of StbM_GetTimeLeap() the returned time difference shall be limited to either the maximum negative or the maximum positive value of the type of timeJump [SWS_StbM_00300].

StbM_GetTimeLeap() shall return E_NOT_OK until the second valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

In Comments Field for GetTimeLeap Operation in SWS_StbM_00247
remove subclause ", if StbMTimeLeapFuture/PastThreshold has been exceeded"

In Description Field for API StbM_GetTimeLeap in SWS_StbM_00267
remove subclause ", if StbMTimeLeapFuture/PastThreshold is exceeded"

In Description field of [SWS_StbM_00273]
replace "StbM_SetTimer" by "StbM_StartTimer"

In "Parameters (in):" field of [SWS_StbM_00273]
replace
"Actual time value captured, when callback is called by StbM"
by
"Difference time value when callback is called by StbM"

In [SWS_StbM_00271] replace
"actualTime" by "deviationTime"
–Last change on issue 78023 comment 20–

BW-C-Level:

Application	Specification	Bus
4	4	1

1.18 Specification Item SWS_StbM_00275

Trace References:

SRS_StbM_20010, SRS_StbM_20056

Content:

Name	TimeNotificationTimeNotification	
Comment	Notification, which indicates, that the timer has expired, which has been set by StbM_StartTimer	
IsService	true	
Variation	{ecuc(StbM/StbMSynchronizedTimeBase/StbMSynchronizedTimeBaseIdentifier)} < 128	
Possible Errors	0	E_OK
	1	E_NOT_OK

Operations:

NotifyTimeTimeNotification.NotifyTime			
Comments	Notification, which indicates, that the timer has expired, which has been set by StbM_StartTimer		
Variation	–		
Parameters	deviationTimeTime Notification.Notify Time.deviationTime	Comment	–
		Type	StbM_TimeDiffType
		Variation	–
		Direction	IN
Possible Errors	E_OK	Operation successful	
	E_NOT_OK	Operation failed	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76895: Naming of ServiceInterfaces

  Problem description:

  The names of service interfaces in chapter 8.2.3 and 8.2.4 is inconsistent. In some cases the prefix StbM is used:

  * StbM_StartTimer
  * StbM_MeasurementNotification_TB_Name

  and in some cases it is not used:

  * StatusNotification
  * GlobalTime_Master_Name

* GlobalTime_Slave_Name
* TimeNotification

In my understanding the prefix should be used consistently, i.e. the prefix should **not** exist in the shortName of any service interface.

Agreed solution:

remove StbM_ prefix from interface names
- StbM_StartTimer
- in SWS_StbM_00409 ("Name" field)
- in SWS_StbM_00275 ("Comments" fields)
- in SWS_StbM_91004 ("Interface" field)

and
- StbM_MeasurementNotification_TB_Name
- in SWS_StbM_00339 ("Name" field)
- in SWS_StbM_00387 ("Name" field + "Interface" field)
–Last change on issue 76895 comment 2–

BW-C-Level:

Application	Specification	Bus
4	4	1

1.19 Specification Item SWS_StbM_00277

Trace References:

SRS_StbM_20001, SRS_StbM_20054

Content:

For Synchronizedand Offset , Offset and Pure Local Time Bases the StbM shall notify Status Notification Customers of a Time Base about status related events by the callback StbM_StatusNotificationCallback, which shall to be set via configuration parameter Stb MStatusNotificationCallback (ECUC_StbM_00046 :).

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77872: [StbM] StbM_Clarification_On_Pure_Local_Timebases

  Problem description:

  1. [SWS_StbM_00277] mentions that StatusNotificationCallback should be called for for Synchronized and Offset Time

Bases , but in [SWS_StbM_00286] the variation is mentioned as ecuc(StbM/StbMSynchronizedTimeBase/StbMSynchronizedTimeBaseIdentifier)
< 128
Question - StatusNotificationCallback should be called for pure local time bases also?

2. For pure local time bases, Event notification bits should be set, when StbM_SetGlobalTime()/StbM_UpdateGlobalTime() API are called (GLOBAL_TIME_BASE bit is set when StbM_SetGlobalTime()/StbM_UpdateGlobalTime is called) ?

Agreed solution:

Change SWS_StbM_00277 from
For Synchronized and Offset Time Bases the StbM
to
For Synchronized, Offset and Pure Local Time Bases the StbM
–Last change on issue 77872 comment 2–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.20 Specification Item SWS_StbM_00286

Trace References:

SRS_StbM_20010, SRS_StbM_20054

Content:

Name	StatusNotificationStatusNotification	
Comment	Callback, which is called, if Notification about a Time Base related status event occurs change	
IsService	false	
Variation	{ecuc(StbM/StbMSynchronizedTimeBase/StbMSynchronizedTimeBaseIdentifier)} < 128	
Data Elements	eventNotificationStatusNotification.eventNotification	
	Type	StbM_TimeBaseNotificationType
	Variation	–

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76868: SWS_StbM_00286: Description does not make sense.

Problem description:

The description of the SenderReceiverInterface StbM_StatusNotification says:

"Callback, which is called, if a Time Base related status event occurs"

A callback is a callback, and an event is an event. But an event is not a callback.

The description should be fixed.
–Last change on issue 76868 comment 12–

Agreed solution:

Change comment in SWS_StbM_00286 from "Callback, which is called, if a Time Base related status event occurs" to "Notification about a Time Base related status change"
–Last change on issue 76868 comment 5–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.21 Specification Item SWS_StbM_00287

Trace References:

SRS_StbM_20010, SRS_StbM_20054

Content:

Name	StbM_TimeBaseNotificationTypeStbM_TimeBaseNotificationType
Kind	Bitfield
Derived from	uint32

Name	StbM_TimeBaseNotificationTypeStbM_TimeBaseNotificationType			
Elements	Kind	Name	Mask	Description
	bit	EV_GLOBAL_TIMEStbM_TimeBaseNotificationType.EV_GLOBAL_TIME	0x01	Bit 0 (LSB): 0: synchronization to global time master not changed 1: GLOBAL_TIME_BASE in StbM_TimeBaseStatusType has changed from 0 to 1
	bit	EV_TIMEOUT_OCCURREDStbM_TimeBaseNotificationType.EV_TIMEOUT_OCCURRED	0x01 0x02	Bit 1: 1: TIMEOUT bit in timeBaseStatus has changed from 0 to 1 0: otherwise
	bit	EV_TIMEOUT_REMOVEDStbM_TimeBaseNotificationType.EV_TIMEOUT_REMOVED	0x04	Bit 2 1: TIMEOUT bit in timeBaseStatus has changed from 1 to 0 0: otherwise
	bit	EV_TIMELEAP_FUTUREStbM_TimeBaseNotificationType.EV_TIMELEAP_FUTURE	0x08	Bit 3 1: TIMELEAP_FUTURE bit in timeBaseStatus has changed from 0 to 1 0: otherwise
	bit	EV_TIMELEAP_FUTURE_REMOVEDStbM_TimeBaseNotificationType.EV_TIMELEAP_FUTURE_REMOVED	0x10	Bit 4 1: TIMELEAP_FUTURE bit in timeBaseStatus has changed from 1 to 0 0: otherwise
	bit	EV_TIMELEAP_PASTStbM_TimeBaseNotificationType.EV_TIMELEAP_PAST	0x20	Bit 5 1: TIMELEAP_PAST bit in timeBaseStatus has changed from 0 to 1 0: otherwise
	bit	EV_TIMELEAP_PAST_REMOVEDStbM_TimeBaseNotificationType.EV_TIMELEAP_PAST_REMOVED	0x40	Bit 6 1: TIMELEAP_PAST bit in timeBaseStatus has changed from 1 to 0 0: otherwise
	bit	EV_SYNC_TO_SUBDOMAINStbM_TimeBaseNotificationType.EV_SYNC_TO_SUBDOMAIN	0x80	Bit 7 1: SYNC_TO_GATEWAY bit in timeBaseStatus has changed from 0 to 1 0: otherwise
	bit	EV_SYNC_TO_GLOBAL_MASTERStbM_TimeBaseNotificationType.EV_SYNC_TO_GLOBAL_MASTER	0x100	Bit 8 1: SYNC_TO_GATEWAY bit of Time Domain changes from 1 to 0 0: otherwise
	bit	EV_RESYNCStbM_TimeBaseNotificationType.EV_RESYNC	0x0200	Bit 9: 1: A synchronization of the local time to the valid Global Time value has occured 0: No resynchronization event occured
	bit	EV_RATECORRECTIONStbM_TimeBaseNotificationType.EV_RATECORRECTION	0x0400	Bit 10 1: a valid rate correction has been calculated (not beyond limits) 0: No rate correction calculated

Name	StbM_TimeBaseNotificationTypeStbM_TimeBaseNotificationType
Description	The StbM_TimeBaseNotificationType type defines a number of global time related events. The type definition is used for storing the events in the status variable NotificationEvents and for setting the mask variable NotificationMask which defines a subset of events for which an interrupt request shall be raised.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #78023: [StbM]: Clarification required on return value of the API 'StbM_GetTimeLeap()'

  Problem description:

  As per the requirement [SWS_StbM_00267] StbM_GetTimeLeap() returns value of Time Leap, if StbMTimeLeapFuture/PastThreshold is exceeded.

  The value of Time Leap returned in 'timeJump' in API StbM_GetTimeLeap() would be the value which is calculated in API StbM_BusSetGlobalTime() only?

  And if StbMTimeLeapFuture/PastThreshold is not exceeded then, we are assuming that StbM_GetTimeLeap() shall return E_NOT_OK.

  Agreed solution:

  Add a new requirement in ch. 7 for clarification after SWS_StbM_00305:

  SWS_StbM_00xxx:
  For Time Slaves and Time Gateways of Synchronized Time Bases StbM_GetTimeLeap() shall return the time difference between the newly received and the current Time Base value, i.e. TG - TL, which is calculated upon each, except the very first, valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

  With
  - TL = Current value of the local instance of the Time Base (before newly received Time Base value is applied)
  - TG = Newly received Time Base value

  For Time Slaves and Time Gateways of Offset Time Bases StbM_GetTimeLeap() shall return the time difference between the newly received and the current Time Base offset value, i.e. TOG - TOL, which is calculated upon each, except the very first, valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

  With
  - TOL = Current offset value of the local instance of the Time Base (before newly

received Time Base offset value is applied)
- TOG = Newly received Time Base offset value

If the calculated time difference exceeds the value range of the timeJump parameter of StbM_GetTimeLeap() the returned time difference shall be limited to either the maximum negative or the maximum positive value of the type of timeJump [SWS_StbM_00300].

StbM_GetTimeLeap() shall return E_NOT_OK until the second valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

In Comments Field for GetTimeLeap Operation in SWS_StbM_00247
remove subclause ", if StbMTimeLeapFuture/PastThreshold has been exceeded"

In Description Field for API StbM_GetTimeLeap in SWS_StbM_00267
remove subclause ", if StbMTimeLeapFuture/PastThreshold is exceeded"

In Description field of [SWS_StbM_00273]
replace "StbM_SetTimer" by "StbM_StartTimer"

In "Parameters (in):" field of [SWS_StbM_00273]
replace
"Actual time value captured, when callback is called by StbM"
by
"Difference time value when callback is called by StbM"

In [SWS_StbM_00271] replace
"actualTime" by "deviationTime"
–Last change on issue 78023 comment 20–

BW-C-Level:

Application	Specification	Bus
4	4	1

1.22 Specification Item SWS_StbM_00290

Trace References:

SRS_StbM_20010, SRS_StbM_20054

Content:

Name	GlobalTime_StatusEvent_{TBName}StbM.GlobalTime_StatusEvent		
Kind	RequiredProvidedPort	Interface	StatusNotification
Description	–		
Variation	({ecuc(StbM/StbMSynchronizedTimeBase/StbMStatusNotificationCallback)} != NULL) TBName = {ecuc(StbM/StbMSynchronizedTimeBase.SHORT-NAME)}		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76897: Required Ports described in C8.2.2 are actually provided Ports

  Problem description:

  The PortPrototypes described in chapter 8.2.2 represent information that is passed from the StbM to an application software-component, right?

  If this assumption is correct then all these PortPrototypes need to be PPortPrototypes because, according to the SWS STbM, these PortPrototypes are typed by a SenderReceiverInterface.

  In other words, a PortPrototype typed by a Sender ReceiverInterface that is supposed to pass information out of the StbM can only be a PPortPrototype.

  Agreed solution:

  Move SWS_StbM_00290 GlobalTime_StatusEvent to section 8.2.1 "Provided Ports" and change attribute "Kind" from Required Port to Provided Port.
  –Last change on issue 76897 comment 2–

  BW-C-Level:

Application	Specification	Bus
4	4	1

1.23 Specification Item SWS_StbM_00301

Trace References:

SRS_StbM_20056

Content:

Name	StbM_RateDeviationTypeStbM_RateDeviationType
Kind	Type

Name	StbM_RateDeviationTypeStbM_RateDeviationType		
Derived from	sint32 sint16		
Description	Variables of this type are used to express a rate deviation in ppm.		
Range	-32000..32000		parts per million (-32000..32000)
Variation	–		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77020: [StbM]: Clarification required on the 'type' of the element 'RateDeviation' and the Implementation Data Type 'StbM_RateDeviationType'

  Problem description:

  As per the requirement Id [SWS_StbM_00301]: 'StbM_RateDeviationType' is of type 'sint32'
  But the range mentioned for the 'StbM_RateDeviationType' is (-32000..32000).
  For the range of (-32000..32000) 'sint16' type can also be used.

  In requirement [SWS_StbM_00332]: In 'StbM_SyncRecordTableBlockType' the element 'RateDeviation' is of type 'sint16'

  And in requirement [SWS_StbM_00308]: In Block elements, the element 'RateDeviation' is of type 'sint16' and the number of 'Bytes' assigned is 2

  Hence we need clarification with respect to the 'type' of the element 'RateDeviation' and the 'StbM_RateDeviationType'.

  Agreed solution:

  In SWS_StbM_00301 change field DerivedFrom from sint32 to sint16.

  In SWS_StbM_00332 and SWS_StbM_00308 change the type of element 'RateDeviation' from 'sint16' to StbM_RateDeviationType
  –Last change on issue 77020 comment 3–

  BW-C-Level:

Application	Specification	Bus
4	4	1

1.24 Specification Item SWS_StbM_00304

Trace References:

SRS_StbM_20028

Content:

On invocation of StbM_SetGlobalTime() or StbM_UpdateGlobalTime() for Time Domains 16 to 31 the StbM shall check the GLOBAL_TIME_BASE bit within timeBaseStatus of the underlying Synchronized Time Base and shall return E_NOT_OK if is not set.

If the GLOBAL_TIME_BASE bit is set, the StbM:

- shall calculate the Offset Time by obtaining the actual Time Base value of the underlying Synchronized Time Base and subtract that from the Absolute Time value which is passed by StbM_SetGlobalTime() or StbM_UpdateGlobalTime()
- shall call StbM_SetOffset () update the corresponding Offset Time Base with the calculated Offset Time value and the User Data that was passed by StbM_SetGlobalTime() or StbM_UpdateGlobalTime().

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77939: [StbM]: Correction required in 'timeBaseUpdateCounter' increment for Pure Local Time Base

  Problem description:

  As per the requirement [SWS_StbM_00350], StbM shall increment the 'timeBaseUpdateCounter' of the corresponding Time Base by 1 (one).

  But as per requirement [SWS_StbM_00347] and [SWS_StbM_00348], the API StbM_GetTimeBaseUpdateCounter() shall report to DET the development error STBM_E_PARAM, if called with a parameter 'timeBaseID', which refers to a Pure Local Time Base.

  And in section '2.2.12 Time Master' as a 'Special Case Pure Local Time Master' it is mentioned that, "A Pure Local Time Master is an entity which is the master of a Pure Local Time Base and which does therefore not propagate this time base to any Time Slave."

  So the requirement [SWS_StbM_00350] should also mention that, StbM shall not increment the 'timeBaseUpdateCounter' of the corresponding Time Base by 1 (one) if the 'timeBaseID' refers to Pure Local Time Base.

  Is the above understanding is correct?

  Agreed solution:

Change [SWS_StbM_00351] by adding the following subclause at the beginning of those requirements:
"For Synchronized and Offset Time Bases, ..."

Change SWS_StbM_00350 to (to also consider StbM_SetOffset) from:
"On a valid invocation of StbM_SetGlobalTime(), StbM_BusSetGlobalTime() or StbM_TriggerTimeTransmission() the StbM shall increment the timeBaseUpdateCounter of the corresponding Time Base by 1 (one). At 255 it shall wrap around to 0."
to
" For Synchronized and Offset Time Bases on a valid invocation of StbM_SetGlobalTime(), StbM_BusSetGlobalTime(), or StbM_TriggerTimeTransmission() and
for Offset Time Bases on a valid invocation of StbM_SetOffset()
the StbM shall increment the timeBaseUpdateCounter of the corresponding Time Base by 1 (one).
At 255 the timeBaseUpdateCounter shall wrap around to 0."

Add a note after SWS_StbM_00350:
Note: For Offset Time Bases the term corresponding Time Base refers to the Offset Time Base only and not to the underlying Synchronized Time Base.

Change SWS_StbM_00304 from
" shall call StbM_SetOffset() with the calculated Offset Time value and the User Data passed by StbM_SetGlobalTime() or StbM_UpdateGlobalTime()."
to
" shall update the corresponding Offset Time Base with the calculated Offset Time value and the User Data that was passed by StbM_SetGlobalTime() or StbM_UpdateGlobalTime()."

Change [SWS_StbM_00181] from
On a valid invocation of StbM_SetGlobalTime()or StbM_UpdateGlobalTime() the StbM shall set the GLOBAL_TIME_BASE bit within timeBaseStatus of the corresponding Time Domain and shall clear all other bits.
to
On a valid invocation of StbM_SetGlobalTime(), StbM_UpdateGlobalTime(), or StbM_SetOffset() the StbM shall set the GLOBAL_TIME_BASE bit within timeBaseStatus of the corresponding Time Base and shall clear all other bits.

Change the sentence in the introduction of ch. 7.3.6
"Nevertheless, it might be necessary, that the StbM provides an interface,..."
to

"Nevertheless it is necessary, that the StbM provides an interface, ..."
–Last change on issue 77939 comment 9–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.25 Specification Item SWS_StbM_00305

Trace References:

SRS_StbM_20025

Content:

For each Time Domain where a Time Slave or a Time Gateway Slave Port belongs to, an invocation of StbM_BusSetGlobalTime() shall check, if the time difference between the current and the updated Time Base value exceeds the configured threshold of StbMTimeLeapPastThreshold (ECUC_StbM_00042 :), i.e. TLSync - TG > StbMTimeLeapPastThreshold, if at least one Time Base value has been successfully received before.

With:

- TLSync = Value of the local instance of the Time Base before the new value of the Global Time is applied
- TG = Received value of the Global Time

In case the threshold is exceeded the StbM shall set the TIMELEAP_PAST bit within timeBaseStatus of the Time Base.

If the next StbMClearTimeleapCount updates are within the threshold of StbMTimeLeapPastThreshold the StbM shall clear the TIMELEAP_PAST bit within timeBaseStatus of the Time Base.

A threshold of 0 shall deactivate this check.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77346: Difference calculation for StbMTimeLeapFutureThreshold

  Problem description:

  In [SWS_StbM_00182] it is stated that
  "if the time difference between the current and the updated Time Base value exceeds the configured threshold of StbMTimeLeapFutureThreshold" the "TIMELEAP_FUTURE bit within timeBaseStatus of the Time Base" shall be set.

The statement "time difference between the current and the updated Time Base value" suggests that this means subtracting the updated Time Base value from the current value.
For a time leap into the future this will result in a negative value.
The threshold value StbMTimeLeapFutureThreshold however has been defined in [ECUC_StbM_00041] as "This represents the maximum allowed positive difference between the current Local Time Base value and a newly received Global Time Base value [unit: seconds]."

Agreed solution:

Change SWS_StbM_00182
from:
"..., if the time difference between the current and the updated Time Base value exceeds the configured threshold of StbMTimeLeapFutureThreshold (ECUC_StbM_00041 :),
if at least one Time Base value has been successfully received before..."
to
"..., if the time difference between the updated Time Base and the current value exceeds the configured threshold of StbMTimeLeapFutureThreshold (ECUC_StbM_00041 :),
i.e. TG - TLSync > StbMTimeLeapFutureThreshold, if at least one Time Base value has been successfully received before.

(where:
- TLSync: Value of the local instance of the Time Base before the new value of the Global Time is applied
- TG: Received value of the Global Time)

..."

change SWS_StbM_00305
from:
"... if the time difference between the current and the updated Time Base value exceeds the configured threshold of StbMTimeLeapPastThreshold (ECUC_StbM_00042 :),
if at least one Time Base value has been successfully received before..."
to
"..., if the time difference between the current and the updated Time Base value exceeds the configured threshold of StbMTimeLeapPastThreshold (ECUC_StbM_00042 :),
i.e. TLSync - TG > StbMTimeLeapPastThreshold, if at least one Time Base value has been successfully received before.

(where:
- TLSync = Value of the local instance of the Time Base before the new value of the Global Time is applied
- TG = Received value of the Global Time)

Change Description in ECUC_StbM_00041
from "This represents the maximum allowed positive difference between the current Local Time Base value and a newly received Global Time Base value [unit: seconds]."
to
"This represents the maximum allowed positive difference between a newly received Global Time Base value and the current Local Time Base value [unit: seconds]."
–Last change on issue 77346 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.26 Specification Item SWS_StbM_00308

Trace References:

SRS_StbM_20057

Content:

If StbMTimeRecordingSupport (ECUC_StbM_00038 :) is set to TRUE, the StbM establishes a table to record values depending on the Synchronized Time Base with the following structure:

	Record Table Element	Multi-plicity	Range	Bytes	Type / Unit
Header		1		9	
	Synchronized TimeDomain	1	0..15	1	uint8
	HWfrequency	1	0..4294967295	4	uint32 / Hz
	HWprescaler	1	0..4294967295	4	uint32
Block 0		1		27	
	GlbSeconds	1	0..4294967295	4	StbM_TimeStamp Type. seconds
	GlbNanoSeconds	1	0..999999999	4	StbM_TimeStamp Type. nanoseconds

	Record Table Element	Multi-plicity	Range	Bytes	Type / Unit
	TimeBaseStatus	1	0..255	1	StbM_TimeStamp Type. StbM_TimeBase StatusType
	HWcounter	1	0..4294967295	4	uint32 / nanoseconds
	RateDeviation	1	0..+-32000	2	sint16 StbM_Rate DeviationType / ppm
	LocSeconds	1	0..4294967295	4	StbM_TimeStamp Type. seconds
	LocNanoSeconds	1	0..999999999	4	StbM_TimeStamp Type. nanoseconds
	PathDelay	1	0..4294967295	4	uint32 / nanoseconds
Block 1	...				
...					
Block (Block-Count-1)	...				

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77020: [StbM]: Clarification required on the ’type’ of the element ’RateDeviation’ and the Implementation Data Type ’StbM_RateDeviationType’

  Problem description:

  As per the requirement Id [SWS_StbM_00301]: ’StbM_RateDeviationType’ is of type ’sint32’
  But the range mentioned for the ’StbM_RateDeviationType’ is (-32000..32000).
  For the range of (-32000..32000) ’sint16’ type can also be used.

  In requirement [SWS_StbM_00332]: In ’StbM_SyncRecordTableBlockType’ the element ’RateDeviation’ is of type ’sint16’

  And in requirement [SWS_StbM_00308]: In Block elements, the element ’RateDeviation’ is of type ’sint16’ and the number of ’Bytes’ assigned is 2

  Hence we need clarification with respect to the ’type’ of the element ’RateDeviation’ and the ’StbM_RateDeviationType’.

  Agreed solution:

In SWS_StbM_00301 change field DerivedFrom from sint32 to sint16.

In SWS_StbM_00332 and SWS_StbM_00308 change the type of element 'RateDeviation' from 'sint16' to StbM_RateDeviationType
–Last change on issue 77020 comment 3–

BW-C-Level:

Application	Specification	Bus
4	4	1

1.27 Specification Item SWS_StbM_00332

Trace References:

SRS_StbM_20057

Content:

Name	StbM_SyncRecordTableBlockTypeStbM_SyncRecordTableBlockType		
Kind	Structure		
Elements	GlbSecondsStbM_SyncRecordTableBlockType.GlbSeconds	uint32	Seconds of the Local Time Base directly after synchronization with the Global Time Base
	GlbNanoSecondsStbM_SyncRecordTableBlockType.GlbNanoSeconds	uint32	Nanoseconds of the Local Time Base directly after synchronization with the Global Time Base
	TimeBaseStatusStbM_SyncRecordTableBlockType.TimeBaseStatus	StbM_TimeBaseStatusType	Time Base Status of the Local Time Base directly after synchronization with the Global Time Base
	HWcounterStbM_SyncRecordTableBlockType.HWcounter	uint32	HW counter reference value directly after synchronization with the Global Time Base
	RateDeviationStbM_SyncRecordTableBlockType.RateDeviation	sint16 StbM_RateDeviationType	Calculated Rate Deviation directly after rate deviation measurement
	LocSecondsStbM_SyncRecordTableBlockType.LocSeconds	uint32	Seconds of the Local Time Base directly before synchronization with the Global Time Base
	LocNanoSecondsStbM_SyncRecordTableBlockType.LocNanoSeconds	uint32	Nanoseconds of the Local Time Base directly before synchronization with the Global Time Base
	PathDelayStbM_SyncRecordTableBlockType.PathDelay	uint32	Current propagation delay in nanoseconds
Description	Synchronized Time Base Record Table Block		
Variation	–		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77020: [StbM]: Clarification required on the ’type’ of the element ’RateDeviation’ and the Implementation Data Type ’StbM_RateDeviationType’

  Problem description:

  As per the requirement Id [SWS_StbM_00301]: ’StbM_RateDeviationType’ is of type ’sint32’
  But the range mentioned for the ’StbM_RateDeviationType’ is (-32000..32000).
  For the range of (-32000..32000) ’sint16’ type can also be used.

  In requirement [SWS_StbM_00332]: In ’StbM_SyncRecordTableBlockType’ the element ’RateDeviation’ is of type ’sint16’

  And in requirement [SWS_StbM_00308]: In Block elements, the element ’RateDeviation’ is of type ’sint16’ and the number of ’Bytes’ assigned is 2

  Hence we need clarification with respect to the ’type’ of the element ’RateDeviation’ and the ’StbM_RateDeviationType’.

  Agreed solution:

  In SWS_StbM_00301 change field DerivedFrom from sint32 to sint16.

  In SWS_StbM_00332 and SWS_StbM_00308 change the type of element ’RateDeviation’ from ’sint16’ to StbM_RateDeviationType
  –Last change on issue 77020 comment 3–

  BW-C-Level:

Application	Specification	Bus
4	4	1

1.28 Specification Item SWS_StbM_00339

Trace References:

SRS_StbM_20057

Content:

Name	StbM_MeasurementNotification_{TB_Name}MeasurementNotification
Comment	Notifies about the availability of a new recorded measurement data block belonging to the Time Base.
IsService	true

Name	StbM_MeasurementNotification_{TB_Name}MeasurementNotification	
Variation	(ecuc(StbM/StbMGeneral/StbMTimeRecordingSupport)} == True) &&({ecuc(StbM/StbMSynchronizedTimeBase/StbMSynchronizedTimeBaseIdentifier)} < 32) TBName={ecuc(StbM/StbMSynchronizedTimeBase.SHORT-NAME)}	
Possible Errors	0	E_OK
	1	E_NOT_OK

Operations:

SetOffsetTimeRecordTableMeasurementNotification.SetOffsetTimeRecordTable			
Comments	Provides to the recorded snapshot data Block of the table belonging to the Offset Time Base.		
Variation	{ecuc(StbM/StbMSynchronizedTimeBase/StbMSynchronizedTimeBaseIdentifier)} > 15 &&{ecuc(StbM/StbMSynchronizedTimeBase/StbMSynchronizedTimeBaseIdentifier)} < 32		
Parameters	offsetRecordTableBlock MeasurementNotification.SetOffsetTimeRecordTable.offsetRecordTableBlock	Comment	Header of the table
		Type	StbM_OffsetRecordTableBlockType
		Variation	–
		Direction	IN
Possible Errors	E_OK	Measurement data access completed successfully	
	E_NOT_OK	Measurement data access failed	

SetSyncTimeRecordTableMeasurementNotification.SetSyncTimeRecordTable			
Comments	Provides the recorded snapshot data Block of the table belonging to the Synchronized Time Base.		
Variation	{ecuc(StbM/StbMSynchronizedTimeBase/StbMSynchronizedTimeBaseIdentifier)} < 16		
Parameters	syncRecordTableBlock MeasurementNotification.SetSyncTimeRecordTable.syncRecordTableBlock	Comment	Block of the table
		Type	StbM_SyncRecordTableBlockType
		Variation	–
		Direction	IN
Possible Errors	E_OK	Measurement data access completed successfully	
	E_NOT_OK	Measurement data access failed	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76895: Naming of ServiceInterfaces

  Problem description:

  The names of service interfaces in chapter 8.2.3 and 8.2.4 is inconsistent. In some cases the prefix StbM is used:

  * StbM_StartTimer
  * StbM_MeasurementNotification_TB_Name

and in some cases it is not used:

* StatusNotification
* GlobalTime_Master_Name
* GlobalTime_Slave_Name
* TimeNotification

In my understanding the prefix should be used consistently, i.e. the prefix should **not** exist in the shortName of any service interface.

Agreed solution:

remove StbM_ prefix from interface names
- StbM_StartTimer
- in SWS_StbM_00409 ("Name" field)
- in SWS_StbM_00275 ("Comments" fields)
- in SWS_StbM_91004 ("Interface" field)

and
- StbM_MeasurementNotification_TB_Name
- in SWS_StbM_00339 ("Name" field)
- in SWS_StbM_00387 ("Name" field + "Interface" field)
–Last change on issue 76895 comment 2–

BW-C-Level:

Application	Specification	Bus
4	4	1

1.29 Specification Item SWS_StbM_00347

Trace References:

SRS_StbM_20064

Content:

Service name:	StbM_GetTimeBaseUpdateCounterStbM_GetTimeBaseUpdateCounter
Syntax:	uint8* uint8 StbM_GetTimeBaseUpdateCounter(StbM_SynchronizedTimeBaseType timeBaseId)
Service ID[hex]:	0x1b
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant

Parameters (in):	timeBaseIdStbM_GetTimeBaseUpdateCounter.timeBaseId	Time Base reference
Parameters (inout):	None	
Parameters (out):	None	
Return value:	uint8* uint8	Counter value belonging to the Time Base, that indicates a Time Base update to the Timesync Modules
Description:	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 #76901: StbM_GetTimeBaseUpdateCounter shall return uint8

  Problem description:

  The API of StbM_GetTimeBaseUpdateCounter in
  section 8.1.3.21 is erroneous since the return value
  should of course not be of type uint8*
  but just a plain uint8

  Agreed solution:

  Change prototype from
  uint8* StbM_GetTimeBaseUpdateCounter(StbM_SynchronizedTimeBaseType timeBaseId)
  to
  uint8 StbM_GetTimeBaseUpdateCounter(StbM_SynchronizedTimeBaseType timeBaseId)
  –Last change on issue 76901 comment 4–

  BW-C-Level:

Application	Specification	Bus
1	4	1

1.30 Specification Item SWS_StbM_00350

Trace References:

SRS_StbM_20064

Content:

On

- For Synchronized and Offset Time Bases on a valid invocation of StbM_SetGlobalTime(), StbM_BusSetGlobalTime(), or StbM_TriggerTimeTransmission() and
- for Offset Time Bases on a valid invocation of StbM_SetOffset(),

the StbM shall increment the timeBaseUpdateCounter of the corresponding Time Base by 1 (one).

At 255 it the timeBaseUpdateCounter shall wrap around to 0.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77939: [StbM]: Correction required in ’timeBaseUpdateCounter’ increment for Pure Local Time Base

  Problem description:

  As per the requirement [SWS_StbM_00350], StbM shall increment the ’timeBaseUpdateCounter’ of the corresponding Time Base by 1 (one).

  But as per requirement [SWS_StbM_00347] and [SWS_StbM_00348], the API StbM_GetTimeBaseUpdateCounter() shall report to DET the development error STBM_E_PARAM, if called with a parameter ’timeBaseID’, which refers to a Pure Local Time Base.

  And in section ’2.2.12 Time Master’ as a ’Special Case Pure Local Time Master’ it is mentioned that, "A Pure Local Time Master is an entity which is the master of a Pure Local Time Base and which does therefore not propagate this time base to any Time Slave."

  So the requirement [SWS_StbM_00350] should also mention that, StbM shall not increment the ’timeBaseUpdateCounter’ of the corresponding Time Base by 1 (one) if the ’timeBaseID’ refers to Pure Local Time Base.

  Is the above understanding is correct?

  Agreed solution:

  Change [SWS_StbM_00351] by adding the following subclause at the beginning of those requirements:
  "For Synchronized and Offset Time Bases, ..."

  Change SWS_StbM_00350 to (to also consider StbM_SetOffset) from:
  "On a valid invocation of StbM_SetGlobalTime(), StbM_BusSetGlobalTime() or StbM_TriggerTimeTransmission() the StbM shall increment the timeBaseUpdateCounter of the corresponding Time Base by 1 (one). At 255 it shall wrap around to

0."
to
" For Synchronized and Offset Time Bases on a valid invocation of StbM_SetGlobalTime(), StbM_BusSetGlobalTime(), or StbM_TriggerTimeTransmission() and
for Offset Time Bases on a valid invocation of StbM_SetOffset()
the StbM shall increment the timeBaseUpdateCounter of the corresponding Time Base by 1 (one).
At 255 the timeBaseUpdateCounter shall wrap around to 0."

Add a note after SWS_StbM_00350:
Note: For Offset Time Bases the term corresponding Time Base refers to the Offset Time Base only and not to the underlying Synchronized Time Base.

Change SWS_StbM_00304 from
" shall call StbM_SetOffset() with the calculated Offset Time value and the User Data passed by StbM_SetGlobalTime() or StbM_UpdateGlobalTime()."
to
" shall update the corresponding Offset Time Base with the calculated Offset Time value and the User Data that was passed by StbM_SetGlobalTime() or StbM_UpdateGlobalTime()."

Change [SWS_StbM_00181] from
On a valid invocation of StbM_SetGlobalTime()or StbM_UpdateGlobalTime() the StbM shall set the GLOBAL_TIME_BASE bit within timeBaseStatus of the corresponding Time Domain and shall clear all other bits.
to
On a valid invocation of StbM_SetGlobalTime(), StbM_UpdateGlobalTime(), or StbM_SetOffset() the StbM shall set the GLOBAL_TIME_BASE bit within timeBaseStatus of the corresponding Time Base and shall clear all other bits.

Change the sentence in the introduction of ch. 7.3.6
"Nevertheless, it might be necessary, that the StbM provides an interface,..."
to
"Nevertheless it is necessary, that the StbM provides an interface, ..."
–Last change on issue 77939 comment 9–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.31 Specification Item SWS_StbM_00351

Trace References:

SRS_StbM_20064

Content:

The For Synchronized and Offset Time Bases, the timeBaseUpdateCounter of a Time Base shall have the value range 0 to 255.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77939: [StbM]: Correction required in ’timeBaseUpdateCounter’ increment for Pure Local Time Base

  Problem description:

  As per the requirement [SWS_StbM_00350], StbM shall increment the ’timeBaseUpdateCounter’ of the corresponding Time Base by 1 (one).

  But as per requirement [SWS_StbM_00347] and [SWS_StbM_00348], the API StbM_GetTimeBaseUpdateCounter() shall report to DET the development error STBM_E_PARAM, if called with a parameter ’timeBaseID’, which refers to a Pure Local Time Base.

  And in section ’2.2.12 Time Master’ as a ’Special Case Pure Local Time Master’ it is mentioned that, "A Pure Local Time Master is an entity which is the master of a Pure Local Time Base and which does therefore not propagate this time base to any Time Slave."

  So the requirement [SWS_StbM_00350] should also mention that, StbM shall not increment the ’timeBaseUpdateCounter’ of the corresponding Time Base by 1 (one) if the ’timeBaseID’ refers to Pure Local Time Base.

  Is the above understanding is correct?

  Agreed solution:

  Change [SWS_StbM_00351] by adding the following subclause at the beginning of those requirements:
  "For Synchronized and Offset Time Bases, ..."

  Change SWS_StbM_00350 to (to also consider StbM_SetOffset) from:
  "On a valid invocation of StbM_SetGlobalTime(), StbM_BusSetGlobalTime() or StbM_TriggerTimeTransmission() the StbM shall increment the timeBaseUpdate-

Counter of the corresponding Time Base by 1 (one). At 255 it shall wrap around to 0."
to
" For Synchronized and Offset Time Bases on a valid invocation of StbM_SetGlobalTime(), StbM_BusSetGlobalTime(), or StbM_TriggerTimeTransmission() and
for Offset Time Bases on a valid invocation of StbM_SetOffset()
the StbM shall increment the timeBaseUpdateCounter of the corresponding Time Base by 1 (one).
At 255 the timeBaseUpdateCounter shall wrap around to 0."

Add a note after SWS_StbM_00350:
Note: For Offset Time Bases the term corresponding Time Base refers to the Offset Time Base only and not to the underlying Synchronized Time Base.

Change SWS_StbM_00304 from
" shall call StbM_SetOffset() with the calculated Offset Time value and the User Data passed by StbM_SetGlobalTime() or StbM_UpdateGlobalTime()."
to
" shall update the corresponding Offset Time Base with the calculated Offset Time value and the User Data that was passed by StbM_SetGlobalTime() or StbM_UpdateGlobalTime()."

Change [SWS_StbM_00181] from
On a valid invocation of StbM_SetGlobalTime()or StbM_UpdateGlobalTime() the StbM shall set the GLOBAL_TIME_BASE bit within timeBaseStatus of the corresponding Time Domain and shall clear all other bits.
to
On a valid invocation of StbM_SetGlobalTime(), StbM_UpdateGlobalTime(), or StbM_SetOffset() the StbM shall set the GLOBAL_TIME_BASE bit within timeBaseStatus of the corresponding Time Base and shall clear all other bits.

Change the sentence in the introduction of ch. 7.3.6
"Nevertheless, it might be necessary, that the StbM provides an interface,..."
to
"Nevertheless it is necessary, that the StbM provides an interface, ..."
–Last change on issue 77939 comment 9–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.32 Specification Item SWS_StbM_00353

Trace References:

SRS_StbM_20065

Content:

If the absolute time offset between the Global Time Base and the local instance of the Time Base (abs(TG - TLTLSync)) is smaller than StbMOffsetCorrectionJumpThreshold (ECUC_StbM_00056 :), the StbM shall calculate the corrected time (TL) of its local instance of the Time Base after the period of StbMOffsetCorrectionAdaptionInterval (ECUC_StbM_00057 :) as specified in [SWS_StbM_00355].

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77379: Correct Offset Correction

  Problem description:

  The description in 7.3.8.2 Offset Correction (for Time Slaves)
  has some flaws regarding the offset correction by additional rate correction.

  The formulas in [SWS_StbM_00356] and [SWS_StbM_00354] shall
  work for the two corner cases:
  1.) start of the offset correction interval
  2.) end of the offset correction interval (i.e., after StbMOffsetCorrectionAdaptionInterval)

  The current formulas fail for the second corner case where the
  requirement is that TL as calculated in [SWS_StbM_00354] matches TL as calculated in [SWS_StbM_00355] for TV = TVSync + TCorrInt

  This RfC addresses some issues that were previously mentioned in # 77247 and are now moved to here.

  Agreed solution:

  Change in [SWS_StbM_00353]
  from
  "If the absolute time offset between the Global Time Base and the local instance of the Time Base (abs(TG - TL)) is smaller than"
  to
  "If the absolute time offset between the Global Time Base and the local instance of the Time Base (abs(TG - TLSync)) is smaller than"

Change in [SWS_StbM_00354]
from
"- TLSync = Corrected current value of the local instance of the Time Base"
to
"- TLSync = Value of the local instance of the Time Base before the new value of the Global Time is applied"

Change in [SWS_StbM_00354]
from
"TL = TLSync + (rrc * (TV - TVSync) * roc)"
to
"TL = TLSync + (TV - TVSync) * (rrc + roc)"

Change in [SWS_StbM_00356]
from
"roc = (TG TLSync) / (TCorrInt) +1"
to
"roc = (TG TLSync) / (TCorrInt)"
–Last change on issue 77379 comment 1–

BW-C-Level:

Application	Specification	Bus
4	4	1

1.33 Specification Item SWS_StbM_00354

Trace References:

SRS_StbM_20065, SRS_StbM_20067

Content:

If the absolute time offset between Global Time Base and local instance of the Time Base (abs(TG - TLSync)) is smaller than StbMOffsetCorrectionJumpThreshold (ECUC_StbM_00056 :), the StbM shall calculate the corrected time (TL) of its local instance of the Time Base within the period of StbMOffsetCorrectionAdaptionInterval (ECUC_StbM_00057 :) as shown:

TL = TLSync + (rrc * (TV - TVSync) * (rrc + roc)

(WhereWith:

- TLSync = Corrected current value Value of the local instance of the Time Base before the new value of the Global Time is applied

- TV = Current value of the Virtual Local Time of the Time Base
- TVSync = Value of the Virtual Local Time as defined in [SWS_StbM_00359]
- rrc = Actual rate for correcting the local instance of the Time Base
- roc = Rate for time offset elimination via Rate Adaption

This correction shall be done whenever the time is read in the scope of these functions:

- StbM_GetCurrentTime()
- StbM_GetCurrentTimeExtended()

This correction shall also be done when the StbM needs to determine the time of the local instance of the Time Base.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77379: Correct Offset Correction

  Problem description:

  The description in 7.3.8.2 Offset Correction (for Time Slaves)
  has some flaws regarding the offset correction by additional rate correction.

  The formulas in [SWS_StbM_00356] and [SWS_StbM_00354] shall
  work for the two corner cases:
  1.) start of the offset correction interval
  2.) end of the offset correction interval (i.e., after StbMOffsetCorrectionAdaptionInterval)

  The current formulas fail for the second corner case where the
  requirement is that TL as calculated in [SWS_StbM_00354] matches TL as calculated in [SWS_StbM_00355] for TV = TVSync + TCorrInt

  This RfC addresses some issues that were previously mentioned in # 77247 and are now moved to here.

  Agreed solution:

  Change in [SWS_StbM_00353]
  from
  "If the absolute time offset between the Global Time Base and the local instance of the Time Base (abs(TG - TL)) is smaller than"
  to
  "If the absolute time offset between the Global Time Base and the local instance of the Time Base (abs(TG - TLSync)) is smaller than"

Change in [SWS_StbM_00354]
from
"- TLSync = Corrected current value of the local instance of the Time Base"
to
"- TLSync = Value of the local instance of the Time Base before the new value of the Global Time is applied"

Change in [SWS_StbM_00354]
from
"TL = TLSync + (rrc * (TV - TVSync) * roc)"
to
"TL = TLSync + (TV - TVSync) * (rrc + roc)"

Change in [SWS_StbM_00356]
from
"roc = (TG TLSync) / (TCorrInt) +1"
to
"roc = (TG TLSync) / (TCorrInt)"
–Last change on issue 77379 comment 1–

BW-C-Level:

Application	Specification	Bus
4	4	1

1.34 Specification Item SWS_StbM_00356

Trace References:

SRS_StbM_20065, SRS_StbM_20067

Content:

The StbM shall correct absolute time offsets between the Global Time Base and the local instance of the Time Base (abs(TG - TLSync)), which are smaller than the value given by StbMOffsetCorrectionJumpThreshold (ECUC_StbM_00056 :) by temporarily applying an additional rate (roc) to rrc. This rate shall be used for the duration defined by parameter StbMOffsetCorrectionAdaptionInterval (ECUC_StbM_00057 :). roc is calculated as shown:

roc = (TG - TLSync) / (TCorrInt) +1

(WhereWith:

- TCorrInt = StbMOffsetCorrectionAdaptionInterval

- TLSync = Value of the local instance of the Time Base before the new value of the Global Time is applied
- TG = Received value of the Global Time

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77379: Correct Offset Correction

  Problem description:

  The description in 7.3.8.2 Offset Correction (for Time Slaves)
  has some flaws regarding the offset correction by additional rate correction.

  The formulas in [SWS_StbM_00356] and [SWS_StbM_00354] shall
  work for the two corner cases:
  1.) start of the offset correction interval
  2.) end of the offset correction interval (i.e., after StbMOffsetCorrectionAdaptionInterval)

  The current formulas fail for the second corner case where the
  requirement is that TL as calculated in [SWS_StbM_00354] matches TL as calculated in [SWS_StbM_00355] for TV = TVSync + TCorrInt

  This RfC addresses some issues that were previously mentioned in # 77247 and are now moved to here.

  Agreed solution:

  Change in [SWS_StbM_00353]
  from
  "If the absolute time offset between the Global Time Base and the local instance of the Time Base (abs(TG - TL)) is smaller than"
  to
  "If the absolute time offset between the Global Time Base and the local instance of the Time Base (abs(TG - TLSync)) is smaller than"

  Change in [SWS_StbM_00354]
  from
  "- TLSync = Corrected current value of the local instance of the Time Base"
  to
  "- TLSync = Value of the local instance of the Time Base before the new value of the Global Time is applied"

  Change in [SWS_StbM_00354]

from
"TL = TLSync + (rrc * (TV - TVSync) * roc)"
to
"TL = TLSync + (TV - TVSync) * (rrc + roc)"

Change in [SWS_StbM_00356]
from
"roc = (TG TLSync) / (TCorrInt) +1"
to
"roc = (TG TLSync) / (TCorrInt)"
–Last change on issue 77379 comment 1–

BW-C-Level:

Application	Specification	Bus
4	4	1

1.35 Specification Item SWS_StbM_00360

Trace References:

SRS_StbM_20065

Content:

At the end of a Rate Correction measurement, the StbM shall calculate the rate (rorc) for Offset Time Bases as shown:

rorc = ((TSStop - TSStart) + (TOStop - TOStart)) / (TSStop TVStop - TSStartTVStart)

With:

- rorc = Rate deviation correction value of the Offset Time Base in regards to the associated Synchronized Time Base
- TSStop TVStop - Current corrected time provided by the local instance of the associated time of the Virtual Local Time of the related Synchronized Time Base
- TOStop - Current Offset value of the Offset Time Base
- TSStart TVStart - Corrected time provided by the local instance of the associated Time Base Value of the Virtual Local Time Base of the related Synchronized Time Base at the start of the measurement
- TOStart - Offset value of the Offset Time Base at the start of the measurement

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77376: [StbM]: Clarification required on how to apply Rate Correction

  Problem description:

  As per the requirement [SWS_StbM_00412]:
  The StbM shall apply Rate Correction to a Synchronized Time Base, if a valid correction rate (rrc) has been calculated.
  The StbM shall apply Rate Correction to an Offset Time Base, if a valid correction rate (rorc) has been calculated.

  Is given below understanding is correct?
  Rate correction of Synchronized and Offset time bases has to be done continuously by multiplying with corresponding Rate Deviation as a correction factor to determine the Rate Corrected Local Instance of time.

  Agreed solution:

  Rate Correction of Offset Time Bases for Time Slaves part (refer to comment # 4):

  [SWS_StbM_00xxx]
  For Offset Time Bases the StbM shall calculate the rate-corrected offset value of the local instance of the Time Base as:

  TOL = TOG + ((TV TVSync) * rorc

  Where:
  - rorc = Rate correction value of the Offset Time Base
  - TOL = Current rate corrected offset value of the local instance of the Offset Time Base.
  - TOG = newly received Offset Time Base value
  - TV = Current value of the Virtual Local Time
  - TVSync = Value of the Virtual Local Time when Offset Time Base value is newly received from master

  This correction shall be done whenever the time is read in the scope of these functions:
  - StbM_GetCurrentTime()
  - StbM_GetCurrentTimeExtended()
  This correction shall also be done when the StbM needs to determine the time of the local instance of the Time Base.
  (SRS_StbM_20065)

  Change [SWS_StbM_00360] from

  At the end of a Rate Correction measurement, the StbM shall calculate the rate (rorc) for Offset Time Bases as shown:

rorc = ((TSStop - TSStart) + (TOStop - TOStart)) / (TSStop - TSStart)

With:
- rorc = Rate deviation of the Offset Time Base in regards to the associated Synchronized Time Base
- TSStop Current corrected time provided by the local instance of the associated Time Base
- TOStop Current Offset value of the Offset Time Base
- TSStart Corrected time provided by the local instance of the associated Time Base
- TOStart Offset value of the Offset Time Base
to
At the end of a Rate Correction measurement, the StbM shall calculate the rate (rorc) for Offset Time Bases as shown:

rorc = (TOStop - TOStart)) / (TVStop - TVStart)

With:
- rorc = Rate correction value of the Offset Time Base
- TVStop Current time of the Virtual Local Time of the related Synchronized Time Base
- TOStop Current Offset value of the Offset Time Base
- TVStart Value of the Virtual Local Time Base of the related Synchronized Time Base at the start of the measurement
- TOStart Offset value of the Offset Time Base at the start of the measurement

Add note below [SWS_StbM_00360]:
Note: To determine the resulting rate deviation the value 1 has to be subtracted from rorc.

Change [SWS_StbM_00363]from

At the end of the Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TSStop Current corrected time provided by the local instance of the associated Time Base
- TOStop Current Offset of the Offset Time Base given as function parameter

to

At the end of the Rate Correction measurement of Offset Time Bases, the

StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TVStop Current time of the Virtual Local Time of the related Synchronized Time Base
- TOStop Current Offset value of the Offset Time Base given as function parameter

Change [SWS_StbM_00365] from

At the start of a Rate correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TSStart Current corrected time provided by the local instance of the associated Time Base
- TOStart Current Offset of the Offset Time Base given as function parameter

to

At the start of a Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TVStart Value of the Virtual Local Time Base of the related Synchronized Time Base at the start of the measurement
- TOStart Current Offset value of the Offset Time Base given as function parameter

Change [SWS_StbM_00412]from
The StbM shall apply Rate Correction to a Synchronized Time Base, if a valid correction rate (rrc) has been calculated.
The StbM shall apply Rate Correction to an Offset Time Base, if a valid correction rate (rorc) has been calculated.

to
The StbM shall use rrc = 1 for a Synchronized Time Base, if a valid correction rate (rrc) has not yet been calculated or is not being calculated (refer [SWS_StbM_00377]) but shall be applied (refer [SWS_StbM_00355], [SWS_StbM_00354]).
The StbM shall use rorc = 1 for an Offset Time Base, if a valid correction rate (rorc) has not yet been calculated or is not being calculated (refer [SWS_StbM_00377]) but shall be applied.

Rate Correction for Global Time Masters part (refer to comment # 5):

Add a new requirement to chapter 7.3.8.3 after SWS_StbM_00396

[SWS_StbM_00xxx]
The StbM shall calculate the rate corrected time (TL) of its local instance of the Time Base as:

TL = TGSync + (TV - TVSync) * rrc

(Where:
- TV = Current value of the Virtual Local Time
- TVSync = Value of the Virtual Local Time at the synchronization event
- TGSync = Value of the Global Time at the synchronization event
- rrc = Rate for correcting the Time Base
(SRS_StbM_20065)

Note: Synchronization events for determining TVSync and TGSync are invocations of StbM_SetRateCorrection(), StbM_SetGlobalTime(), StbM_UpdateGlobalTime(), and the initialization of the StbM. StbM_SetOffset() is an additional synchronization event for Offset Time Bases. In case of StbM_SetRateCorrection() TGSync is calculated as TL based on the previous TGSync value. Additional events might need to be considered for synchronization (e.g., overflow of underlying HW timers). Those should however occur not too often to avoid worsening the precision, e.g., by rounding effects.
–Last change on issue 77376 comment 18–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.36 Specification Item SWS_StbM_00363

Trace References:

SRS_StbM_20065

Content:

At the end of the Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():

- TSStop TVStop - Current corrected time provided by the local instance of the associated time of the Virtual Local Time of the related Synchronized Time Base

- TOStop - Current Offset value of the Offset Time Base given as function parameter

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77376: [StbM]: Clarification required on how to apply Rate Correction

  Problem description:

  As per the requirement [SWS_StbM_00412]:
  The StbM shall apply Rate Correction to a Synchronized Time Base, if a valid correction rate (rrc) has been calculated.
  The StbM shall apply Rate Correction to an Offset Time Base, if a valid correction rate (rorc) has been calculated.

  Is given below understanding is correct?
  Rate correction of Synchronized and Offset time bases has to be done continuously by multiplying with corresponding Rate Deviation as a correction factor to determine the Rate Corrected Local Instance of time.

  Agreed solution:

  Rate Correction of Offset Time Bases for Time Slaves part (refer to comment # 4):

  [SWS_StbM_00xxx]
  For Offset Time Bases the StbM shall calculate the rate-corrected offset value of the local instance of the Time Base as:

  $TOL = TOG + ((TV\ TVSync) * rorc$

  Where:
  - rorc = Rate correction value of the Offset Time Base
  - TOL = Current rate corrected offset value of the local instance of the Offset Time Base.
  - TOG = newly received Offset Time Base value
  - TV = Current value of the Virtual Local Time
  - TVSync = Value of the Virtual Local Time when Offset Time Base value is newly received from master

  This correction shall be done whenever the time is read in the scope of these functions:
  - StbM_GetCurrentTime()
  - StbM_GetCurrentTimeExtended()
  This correction shall also be done when the StbM needs to determine the time of the local instance of the Time Base.
  (SRS_StbM_20065)

Change [SWS_StbM_00360] from

At the end of a Rate Correction measurement, the StbM shall calculate the rate (rorc) for Offset Time Bases as shown:

rorc = ((TSStop - TSStart) + (TOStop - TOStart)) / (TSStop - TSStart)

With:
- rorc = Rate deviation of the Offset Time Base in regards to the associated Synchronized Time Base
- TSStop Current corrected time provided by the local instance of the associated Time Base
- TOStop Current Offset value of the Offset Time Base
- TSStart Corrected time provided by the local instance of the associated Time Base
- TOStart Offset value of the Offset Time Base
to
At the end of a Rate Correction measurement, the StbM shall calculate the rate (rorc) for Offset Time Bases as shown:

rorc = (TOStop - TOStart)) / (TVStop - TVStart)

With:
- rorc = Rate correction value of the Offset Time Base
- TVStop Current time of the Virtual Local Time of the related Synchronized Time Base
- TOStop Current Offset value of the Offset Time Base
- TVStart Value of the Virtual Local Time Base of the related Synchronized Time Base at the start of the measurement
- TOStart Offset value of the Offset Time Base at the start of the measurement

Add note below [SWS_StbM_00360]:
Note: To determine the resulting rate deviation the value 1 has to be subtracted from rorc.

Change [SWS_StbM_00363]from

At the end of the Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TSStop Current corrected time provided by the local instance of the associated Time Base
- TOStop Current Offset of the Offset Time Base given as function parameter

to

At the end of the Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TVStop Current time of the Virtual Local Time of the related Synchronized Time Base
- TOStop Current Offset value of the Offset Time Base given as function parameter

Change [SWS_StbM_00365] from

At the start of a Rate correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TSStart Current corrected time provided by the local instance of the associated Time Base
- TOStart Current Offset of the Offset Time Base given as function parameter

to

At the start of a Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TVStart Value of the Virtual Local Time Base of the related Synchronized Time Base at the start of the measurement
- TOStart Current Offset value of the Offset Time Base given as function parameter

Change [SWS_StbM_00412]from
The StbM shall apply Rate Correction to a Synchronized Time Base, if a valid correction rate (rrc) has been calculated.
The StbM shall apply Rate Correction to an Offset Time Base, if a valid correction rate (rorc) has been calculated.

to
The StbM shall use rrc = 1 for a Synchronized Time Base, if a valid correction rate (rrc) has not yet been calculated or is not being calculated (refer [SWS_StbM_00377]) but shall be applied (refer [SWS_StbM_00355], [SWS_StbM_00354]).
The StbM shall use rorc = 1 for an Offset Time Base, if a valid correction rate (rorc) has not yet been calculated or is not being calculated (refer [SWS_StbM_00377])

but shall be applied.

Rate Correction for Global Time Masters part (refer to comment # 5):

Add a new requirement to chapter 7.3.8.3 after SWS_StbM_00396

[SWS_StbM_00xxx]
The StbM shall calculate the rate corrected time (TL) of its local instance of the Time Base as:

TL = TGSync + (TV - TVSync) * rrc

(Where:
- TV = Current value of the Virtual Local Time
- TVSync = Value of the Virtual Local Time at the synchronization event
- TGSync = Value of the Global Time at the synchronization event
- rrc = Rate for correcting the Time Base
(SRS_StbM_20065)

Note: Synchronization events for determining TVSync and TGSync are invocations of StbM_SetRateCorrection(), StbM_SetGlobalTime(), StbM_UpdateGlobalTime(), and the initialization of the StbM. StbM_SetOffset() is an additional synchronization event for Offset Time Bases. In case of StbM_SetRateCorrection() TGSync is calculated as TL based on the previous TGSync value. Additional events might need to be considered for synchronization (e.g., overflow of underlying HW timers). Those should however occur not too often to avoid worsening the precision, e.g., by rounding effects.
–Last change on issue 77376 comment 18–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.37 Specification Item SWS_StbM_00365

Trace References:

SRS_StbM_20065

Content:

At the start of a Rate correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():

- TSStart TVStart - Current corrected time provided by the local instance of the associated Time Base Value of the Virtual Local Time Base of the related Synchronized Time Base at the start of the measurement
- TOStart - Current Offset value of the Offset Time Base given as function parameter

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77376: [StbM]: Clarification required on how to apply Rate Correction

  Problem description:

  As per the requirement [SWS_StbM_00412]:
  The StbM shall apply Rate Correction to a Synchronized Time Base, if a valid correction rate (rrc) has been calculated.
  The StbM shall apply Rate Correction to an Offset Time Base, if a valid correction rate (rorc) has been calculated.

  Is given below understanding is correct?
  Rate correction of Synchronized and Offset time bases has to be done continuously by multiplying with corresponding Rate Deviation as a correction factor to determine the Rate Corrected Local Instance of time.

  Agreed solution:

  Rate Correction of Offset Time Bases for Time Slaves part (refer to comment # 4):

  [SWS_StbM_00xxx]
  For Offset Time Bases the StbM shall calculate the rate-corrected offset value of the local instance of the Time Base as:

  TOL = TOG + ((TV TVSync) * rorc

  Where:
  - rorc = Rate correction value of the Offset Time Base
  - TOL = Current rate corrected offset value of the local instance of the Offset Time Base.
  - TOG = newly received Offset Time Base value
  - TV = Current value of the Virtual Local Time
  - TVSync = Value of the Virtual Local Time when Offset Time Base value is newly received from master

  This correction shall be done whenever the time is read in the scope of these

functions:
- StbM_GetCurrentTime()
- StbM_GetCurrentTimeExtended()
This correction shall also be done when the StbM needs to determine the time of the local instance of the Time Base.
(SRS_StbM_20065)

Change [SWS_StbM_00360] from

At the end of a Rate Correction measurement, the StbM shall calculate the rate (rorc) for Offset Time Bases as shown:

rorc = ((TSStop - TSStart) + (TOStop - TOStart)) / (TSStop - TSStart)

With:
- rorc = Rate deviation of the Offset Time Base in regards to the associated Synchronized Time Base
- TSStop Current corrected time provided by the local instance of the associated Time Base
- TOStop Current Offset value of the Offset Time Base
- TSStart Corrected time provided by the local instance of the associated Time Base
- TOStart Offset value of the Offset Time Base
to
At the end of a Rate Correction measurement, the StbM shall calculate the rate (rorc) for Offset Time Bases as shown:

rorc = (TOStop - TOStart)) / (TVStop - TVStart)

With:
- rorc = Rate correction value of the Offset Time Base
- TVStop Current time of the Virtual Local Time of the related Synchronized Time Base
- TOStop Current Offset value of the Offset Time Base
- TVStart Value of the Virtual Local Time Base of the related Synchronized Time Base at the start of the measurement
- TOStart Offset value of the Offset Time Base at the start of the measurement

Add note below [SWS_StbM_00360]:
Note: To determine the resulting rate deviation the value 1 has to be subtracted from rorc.

Change [SWS_StbM_00363]from

At the end of the Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TSStop Current corrected time provided by the local instance of the associated Time Base
- TOStop Current Offset of the Offset Time Base given as function parameter

to

At the end of the Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TVStop Current time of the Virtual Local Time of the related Synchronized Time Base
- TOStop Current Offset value of the Offset Time Base given as function parameter

Change [SWS_StbM_00365] from

At the start of a Rate correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TSStart Current corrected time provided by the local instance of the associated Time Base
- TOStart Current Offset of the Offset Time Base given as function parameter

to

At the start of a Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TVStart Value of the Virtual Local Time Base of the related Synchronized Time Base at the start of the measurement
- TOStart Current Offset value of the Offset Time Base given as function parameter

Change [SWS_StbM_00412]from
The StbM shall apply Rate Correction to a Synchronized Time Base, if a valid correction rate (rrc) has been calculated.
The StbM shall apply Rate Correction to an Offset Time Base, if a valid correction rate (rorc) has been calculated.

to

The StbM shall use rrc = 1 for a Synchronized Time Base, if a valid correction rate (rrc) has not yet been calculated or is not being calculated (refer [SWS_StbM_00377]) but shall be applied (refer [SWS_StbM_00355], [SWS_StbM_00354]).
The StbM shall use rorc = 1 for an Offset Time Base, if a valid correction rate (rorc) has not yet been calculated or is not being calculated (refer [SWS_StbM_00377]) but shall be applied.

Rate Correction for Global Time Masters part (refer to comment # 5):

Add a new requirement to chapter 7.3.8.3 after SWS_StbM_00396

[SWS_StbM_00xxx]
The StbM shall calculate the rate corrected time (TL) of its local instance of the Time Base as:

TL = TGSync + (TV - TVSync) * rrc

(Where:
- TV = Current value of the Virtual Local Time
- TVSync = Value of the Virtual Local Time at the synchronization event
- TGSync = Value of the Global Time at the synchronization event
- rrc = Rate for correcting the Time Base
(SRS_StbM_20065)

Note: Synchronization events for determining TVSync and TGSync are invocations of StbM_SetRateCorrection(), StbM_SetGlobalTime(), StbM_UpdateGlobalTime(), and the initialization of the StbM. StbM_SetOffset() is an additional synchronization event for Offset Time Bases. In case of StbM_SetRateCorrection() TGSync is calculated as TL based on the previous TGSync value. Additional events might need to be considered for synchronization (e.g., overflow of underlying HW timers). Those should however occur not too often to avoid worsening the precision, e.g., by rounding effects.
–Last change on issue 77376 comment 18–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.38 Specification Item SWS_StbM_00374

Trace References:

SRS_StbM_20065

Content:

For Rate Correction measurements, the StbM shall evaluate state changes of the TIMEOUT flag. The StbM shall discard measurements when the state changes during a measurementthe measurement, if the flag equals "Set".

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77671: Various Time Base Status clarifications

  Problem description:

  1.) Both, [SWS_StbM_00183] and [SWS_StbM_00187]
  specify that "If the timeout occurs, the StbM shall set the TIMEOUT bit within timeBaseStatus of the Time Base."

  Since the specified timeout time expires asynchronous
  to any StbM invocation it should be defined when the TIMEOUT bit shall be set.
  Since there is no requirement to set up a dedicated timer outside the StbM which shall call the StbM on expiration there are in principle two occasions where the TIMEOUT bit can be set:
  1.) Inside the cyclically called StbM_MainFunction
  2.) Within every function that returns the Time Base Status, e.g. StbM_GetTimeBaseStatus(), StbM_GetCurrentTime().

  Doing it only in a function that returns the Time Base Status is not sufficient if EV_TIMEOUT_OCCURED was configured as a Status Event.
  Doing it only inside the StbM_MainFunction leads to a slightly deferred TIMEOUT detection (and the points in time at which multiple ECUs will detect a timeout will differ more from each other).

  It is also undefined which time shall be used to determine the timeout (either the Virtual Local Time or the Local Instance of the Synchronized Time Base itself).

  2.) [SWS_StbM_00374] specifies that "For Rate Correction measurements, the StbM shall evaluate state changes of the TIMEOUT flag. The StbM shall discard measurements when the state changes during a measurement."

This requirement is overspecified and thus confusing since it is sufficient to only discard the measurement if the TIMEOUT flag gets set.
If a timeout occurs and measurement is stopped, a new measurement is only started by "reception of time values for Synchronized or Offset Time Bases." ([SWS_StbM_00371]), which will then reset the TIMEOUT flag.

3.) [SWS_StbM_00376] specifies that "the StbM shall evaluate state changes of the TIMELEAP_FUTURE and TIMELEAP_PAST flags during measurements. The StbM shall discard the measurement if the flag state changes."

This requirement is overspecified and thus confusing since [SWS_StbM_00373] requires that the StbM shall not start a Rate Correction measurement when either of the flags is set.

Therefore it is sufficient to only discard the measurement if any of the TIMELEAP flags gets set.

Agreed solution:

For 1)
In SWS_StbM_00183 and SWS_StbM_00187 replace the sentence:
The timeout StbMSyncLossTimeout (ECUC_StbM_00028 :) shall be measured from last invocation of StbM_BusSetGlobalTime()
by
The timeout StbMSyncLossTimeout (ECUC_StbM_00028 :) shall be measured based on the Virtual Local Time from last invocation of StbM_BusSetGlobalTime()

Add a new requirement + note after SWS_StbM_00187

SWS_StbM_xxxx: The StbM shall check for a timeout condition of a Time Base within StbM_MainFunction() and all API functions, which return the Time Base Status (e.g. StbM_GetTimeBaseStatus() or StbM_GetCurrentTime())

Note: Since a Status Notification is triggered inside StbM_MainFunction(), the other functions like e.g StbM_GetTimeBaseStatus() might detect a timeout condition sooner than the corresponding Status Notification is actually triggered. Such a delayed Status Notification is considered acceptable.

For 2)
Change SWS_StbM_00374 from:
"..., the StbM shall evaluate state changes of the TIMEOUT flag.
The StbM shall discard measurements when the state changes during a measure-

ment."
to:
"..., the StbM shall evaluate the TIMEOUT flag.
The StbM shall discard the measurement if the flag equals Set."

For 3)
Change SWS_StbM_00376
from:
"..., the StbM shall evaluate state changes of the TIMELEAP_FUTURE and TIMELEAP_PAST flags during measurements.
The StbM shall discard the measurement if the flag state changes."
to:
"..., the StbM shall evaluate the TIMELEAP_FUTURE and TIMELEAP_PAST flags during measurements.
The StbM shall discard the measurement if any of the flags equals Set."

Add a note after SWS_StbM_00362:
"A newly calculated Rate Correction rrc is only applied to following time calculations."
–Last change on issue 77671 comment 14–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.39 Specification Item SWS_StbM_00376

Trace References:

SRS_StbM_20065

Content:

For Rate Correction measurements, the StbM shall evaluate state changes of the TIMELEAP_FUTURE and TIMELEAP_PAST flags during measurements. The StbM shall discard the measurementif the flag state changes, if any of the flags equals "Set".

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77671: Various Time Base Status clarifications

  Problem description:

1.) Both, [SWS_StbM_00183] and [SWS_StbM_00187]
specify that "If the timeout occurs, the StbM shall set the TIMEOUT bit within timeBaseStatus of the Time Base."

Since the specified timeout time expires asynchronous
to any StbM invocation it should be defined when the TIMEOUT bit shall be set.
Since there is no requirement to set up a dedicated timer outside the StbM which shall call the StbM on expiration there are in principle two occasions where the TIMEOUT bit can be set:
1.) Inside the cyclically called StbM_MainFunction
2.) Within every function that returns the Time Base Status, e.g. StbM_GetTimeBaseStatus(), StbM_GetCurrentTime().

Doing it only in a function that returns the Time Base Status is not sufficient if EV_TIMEOUT_OCCURED was configured as a Status Event.
Doing it only inside the StbM_MainFunction leads to a slightly deferred TIMEOUT detection (and the points in time at which multiple ECUs will detect a timeout will differ more from each other).

It is also undefined which time shall be used to determine the timeout (either the Virtual Local Time or the Local Instance of the Synchronized Time Base itself).

2.) [SWS_StbM_00374] specifies that "For Rate Correction measurements, the StbM shall evaluate state changes of the TIMEOUT flag. The StbM shall discard measurements when the state changes during a measurement."

This requirement is overspecified and thus confusing since it is sufficient to only discard the measurement if the TIMEOUT flag gets set.
If a timeout occurs and measurement is stopped, a new measurement is only started by "reception of time values for Synchronized or Offset Time Bases." ([SWS_StbM_00371]), which will then reset the TIMEOUT flag.

3.) [SWS_StbM_00376] specifies that "the StbM shall evaluate state changes of the TIMELEAP_FUTURE and TIMELEAP_PAST flags during measurements. The StbM shall discard the measurement if the flag state changes."

This requirement is overspecified and thus confusing since [SWS_StbM_00373] requires that the StbM shall not start a Rate Correction measurement when either of the flags is set.

Therefore it is sufficient to only discard the measurement if any of the TIMELEAP flags gets set.

Agreed solution:

For 1)
In SWS_StbM_00183 and SWS_StbM_00187 replace the sentence:
The timeout StbMSyncLossTimeout (ECUC_StbM_00028 :) shall be measured from last invocation of StbM_BusSetGlobalTime()
by
The timeout StbMSyncLossTimeout (ECUC_StbM_00028 :) shall be measured based on the Virtual Local Time from last invocation of StbM_BusSetGlobalTime()

Add a new requirement + note after SWS_StbM_00187

SWS_StbM_xxxx: The StbM shall check for a timeout condition of a Time Base within StbM_MainFunction() and all API functions, which return the Time Base Status (e.g. StbM_GetTimeBaseStatus() or StbM_GetCurrentTime())

Note: Since a Status Notification is triggered inside StbM_MainFunction(), the other functions like e.g StbM_GetTimeBaseStatus() might detect a timeout condition sooner than the corresponding Status Notification is actually triggered. Such a delayed Status Notification is considered acceptable.

For 2)
Change SWS_StbM_00374 from:
"..., the StbM shall evaluate state changes of the TIMEOUT flag.
The StbM shall discard measurements when the state changes during a measurement."
to:
"..., the StbM shall evaluate the TIMEOUT flag.
The StbM shall discard the measurement if the flag equals Set."

For 3)
Change SWS_StbM_00376
from:
"..., the StbM shall evaluate state changes of the TIMELEAP_FUTURE and TIMELEAP_PAST flags during measurements.
The StbM shall discard the measurement if the flag state changes."
to:
"..., the StbM shall evaluate the TIMELEAP_FUTURE and TIMELEAP_PAST flags during measurements.
The StbM shall discard the measurement if any of the flags equals Set."

Add a note after SWS_StbM_00362:
"A newly calculated Rate Correction rrc is only applied to following time calculations."
–Last change on issue 77671 comment 14–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.40 Specification Item SWS_StbM_00379

Trace References:

SRS_BSW_00386, SRS_BSW_00323

Content:

If the switch StbMDevErrorDetect (ECUC_StbM_00012 :) is set to TRUE, StbM_GetRateDeviation() shall report to DET the development error STBM_E_PARAM, if called with a parameter timeBaseIDId, which

- is not configured or
- refers to an Offset Time Base or
- is within the reserved value range.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77938: [StbM]: Correction required in DET error reporting for the range of 'timeBaseID' in API 'StbM_GetRateDeviation()'

  Problem description:

  As per the requirements [SWS_StbM_00378] and [SWS_StbM_00379], StbM_GetRateDeviation() shall report to DET the development error STBM_E_PARAM, if called with a parameter 'timeBaseID', which refers to an Offset Time Base.

  But as per the requirements [SWS_StbM_00390] and [SWS_StbM_00391], StbM_SetRateCorrection() shall report to DET the development error STBM_E_PARAM, if called with a parameter 'timeBaseID', which
  -is not configured or
  -is within the reserved value range.

The use case of both the APIs (StbM_GetRateDeviation() and StbM_SetRateCorrection()) as mentioned in 'Note' in requirement [SWS_StbM_00396], is applicable for 'Offset Time Base' also.

So the API StbM_GetRateDeviation() shouldn't report to DET the development error STBM_E_PARAM, if called with a parameter 'timeBaseID', which refers to an Offset Time Base as mentioned in requirement [SWS_StbM_00379].

Agreed solution:

Change,
From:
[SWS_StbM_00379]
If the switch StbMDevErrorDetect (ECUC_StbM_00012 :) is set to TRUE, StbM_GetRateDeviation() shall report to DET the development error STBM_E_PARAM, if called with a parameter timeBaseID, which
-is not configured or
-refers to an Offset Time Base or
-is within the reserved value range.

To:
[SWS_StbM_00379]
If the switch StbMDevErrorDetect (ECUC_StbM_00012 :) is set to TRUE, StbM_GetRateDeviation() shall report to DET the development error STBM_E_PARAM, if called with a parameter timeBaseID, which
-is not configured or
-is within the reserved value range.

Change [SWS_StbM_00397] to
For Time Bases with StbMSynchronizedTimeBaseIdentifier 0 to 31 (ECUC_StbM_00021) and StbMIsSystemWideGlobalTimeMaster = False (ECUC_StbM_00036), the StbM shall return on invocation of StbM_GetRateDeviation() the rate deviation, which has been calculated for that Time Base (i.e., rrc -1 for Synchronized Time Bases or rorc - 1 for Offset Time Bases).

If no rate deviation has been calculated, StbM_GetRateDeviation() shall return E_NOT_OK.

Add new requirement [SWS_StbM_00xxxx] in chapter 7.3.8.3 after SWS_StbM_00396
- For Time Bases with StbMSynchronizedTimeBaseIdentifier 32 to 127 (ECUC_StbM_00021) and

- for Time Bases with StbMSynchronizedTimeBaseIdentifier 0 to 31 and StbMIsSystemWideGlobalTimeMaster = True (ECUC_StbM_00036),
the StbM shall return on invocation of StbM_GetRateDeviation() the rate deviation that has been set by StbM_SetRateCorrection() for that Time Base.

If no rate deviation has been set, StbM_GetRateDeviation() shall return E_NOT_OK.
–Last change on issue 77938 comment 9–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.41 Specification Item SWS_StbM_00387

Trace References:

SRS_StbM_20057

Content:

Name	StbM_MeasurementNotification_{TBName}StbM.GlobalTime_Measurement		
Kind	RequiredPort	Interface	StbM_Measurement Notification_{TB_Name}
Description	–		
Variation	({ecuc(StbM/StbMGeneral/StbMTimeRecordingSupport)} == True) &&({ecuc(StbM/StbMSynchronizedTimeBase/StbMSynchronizedTimeBaseIdentifier)} < 32) TBName={ecuc(StbM/StbMSynchronizedTimeBase.SHORT-NAME)}		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76895: Naming of ServiceInterfaces

  Problem description:

  The names of service interfaces in chapter 8.2.3 and 8.2.4 is inconsistent. In some cases the prefix StbM is used:

  * StbM_StartTimer
  * StbM_MeasurementNotification_TB_Name

  and in some cases it is not used:

  * StatusNotification

* GlobalTime_Master_Name
* GlobalTime_Slave_Name
* TimeNotification

In my understanding the prefix should be used consistently, i.e. the prefix should **not** exist in the shortName of any service interface.

Agreed solution:

remove StbM_ prefix from interface names
- StbM_StartTimer
- in SWS_StbM_00409 ("Name" field)
- in SWS_StbM_00275 ("Comments" fields)
- in SWS_StbM_91004 ("Interface" field)

and
- StbM_MeasurementNotification_TB_Name
- in SWS_StbM_00339 ("Name" field)
- in SWS_StbM_00387 ("Name" field + "Interface" field)

–Last change on issue 76895 comment 2–

BW-C-Level:

Application	Specification	Bus
4	4	1

1.42 Specification Item SWS_StbM_00390

Trace References:

SRS_StbM_20065

Content:

Service name:	StbM_SetRateCorrectionStbM_SetRateCorrection	
Syntax:	Std_ReturnType StbM_SetRateCorrection(StbM_SynchronizedTimeBaseType timeBaseId, const StbM_RateDeviationType* Type rateDeviation)	
Service ID[hex]:	0x12	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	timeBaseIdStbM_SetRate Correction.timeBaseId	Time Base reference
	rateDeviationStbM_SetRate Correction.rateDeviation	Value of the applied rate deviation

Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: successful E_NOT_OK: failed
Description:	Allows to set the rate of a Synchronized Time Base (being either a Pure Local Time Base or not).	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76863: StbM_RateDeviationType within StbM_SetRateCorrection() shall be not a pointer.

  Problem description:

  For primitive data types the RTE expects always non pointer types for function calls which using input data.
  StbM_RateDeviationType within StbM_SetRateCorrection() is currently specified as a pointer, but a primitive data type (sint32). This leads to a compile issue and could only be solved, if the it will be corrected from

  Std_ReturnType StbM_SetRateCorrection(
  StbM_SynchronizedTimeBaseType timeBaseId,
  const StbM_RateDeviationType* rateDeviation
)

  to

  Std_ReturnType StbM_SetRateCorrection(
  StbM_SynchronizedTimeBaseType timeBaseId,
  const StbM_RateDeviationType rateDeviation
)

  Agreed solution:

  Make IN parameter rateDeviation in SWS_StbM_00390 (API StbM_SetRateCorrection) a non pointer:

  Std_ReturnType StbM_SetRateCorrection(
  StbM_SynchronizedTimeBaseType timeBaseId,
  StbM_RateDeviationType rateDeviation
)
  –Last change on issue 76863 comment 4–

  BW-C-Level:

Application	Specification	Bus
4	4	1

1.43 Specification Item SWS_StbM_00393

Trace References:

SRS_StbM_20014, SRS_StbM_20025

Content:

Each For Time Domains 16 to 31 each invocation of StbM_BusSetGlobalTime() shall update the corresponding Offset Time Base and set the User Data and the Time Base Status accordingly.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76975: [StbM] Clarification on updating Offset Time Base

  Problem description:

  The timebaseId received in the API StbM_BusSetGlobalTime() can be 0..15 for synchronized time base and 16..31 for
  offset time base. And based on the timebaseId, either the synchronized time base or the offset time base
  will be set in the API StbM_BusSetGlobalTime(). Is this understanding is correct?

  As per the requirement:
  [SWS_StbM_00393]
  Each invocation of StbM_BusSetGlobalTime() shall update the corresponding Offset Time Base and set the User Data
  and the Time Base Status accordingly.

  The offset time base need to be also updated even when the In parameter 'timebaseId' is referring to synchronized
  time base?

  Agreed solution:

  Change [SWS_StbM_00393]
  from "Each invocation of StbM_BusSetGlobalTime() shall update the corresponding Offset Time Base and set the User Data and the Time Base Status accordingly."
  to "For Time Domains 16 to 31 each invocation of StbM_BusSetGlobalTime() shall update the corresponding Offset Time Base and set the User Data and the Time Base Status accordingly."

Change[SWS_StbM_00179]
from "Each invocation of StbM_BusSetGlobalTime() shall update the corresponding Synchronized Time Base and set the User Data and the Time Base Status accordingly."
to "For Time Domains 0 to 15 each invocation of StbM_BusSetGlobalTime() shall update the corresponding Synchronized Time Base and set the User Data and the Time Base Status accordingly."
–Last change on issue 76975 comment 2–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.44 Specification Item SWS_StbM_00397

Trace References:

SRS_StbM_20065

Content:

On invocation of StbMFor Time Bases with StbMSynchronizedTimeBaseIdentifier 0 to 31 (ECUC_GetRateDeviation() StbM_00021 :) and StbMIsSystemWideGlobalTimeMaster = False (ECUC_StbM_00036 :), the StbM shall return the calculated rate deviationon invocation of StbM_GetRateDeviation() the rate deviation, which has been calculated for that Time Base (i.e., rrc -1 for Synchronized Time Bases or rorc - 1 for Offset Time Bases).

If no rate deviation has yet been calculated, StbM_GetRateDeviation() shall return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77938: [StbM]: Correction required in DET error reporting for the range of ’timeBaseID’ in API ’StbM_GetRateDeviation()’

  Problem description:

  As per the requirements [SWS_StbM_00378] and [SWS_StbM_00379], StbM_GetRateDeviation() shall report to DET the development error STBM_E_PARAM, if called with a parameter ’timeBaseID’, which refers to an Offset Time Base.

  But as per the requirements [SWS_StbM_00390] and [SWS_StbM_00391],

StbM_SetRateCorrection() shall report to DET the development error STBM_E_PARAM, if called with a parameter ’timeBaseID’, which
-is not configured or
-is within the reserved value range.

The use case of both the APIs (StbM_GetRateDeviation() and StbM_SetRateCorrection()) as mentioned in ’Note’ in requirement [SWS_StbM_00396], is applicable for ’Offset Time Base’ also.

So the API StbM_GetRateDeviation() shouldn’t report to DET the development error STBM_E_PARAM, if called with a parameter ’timeBaseID’, which refers to an Offset Time Base as mentioned in requirement [SWS_StbM_00379].

Agreed solution:

Change,
From:
[SWS_StbM_00379]
If the switch StbMDevErrorDetect (ECUC_StbM_00012 :) is set to TRUE, StbM_GetRateDeviation() shall report to DET the development error STBM_E_PARAM, if called with a parameter timeBaseID, which
-is not configured or
-refers to an Offset Time Base or
-is within the reserved value range.

To:
[SWS_StbM_00379]
If the switch StbMDevErrorDetect (ECUC_StbM_00012 :) is set to TRUE, StbM_GetRateDeviation() shall report to DET the development error STBM_E_PARAM, if called with a parameter timeBaseID, which
-is not configured or
-is within the reserved value range.

Change [SWS_StbM_00397] to
For Time Bases with StbMSynchronizedTimeBaseIdentifier 0 to 31 (ECUC_StbM_00021) and StbMIsSystemWideGlobalTimeMaster = False (ECUC_StbM_00036), the StbM shall return on invocation of StbM_GetRateDeviation() the rate deviation, which has been calculated for that Time Base (i.e., rrc -1 for Synchronized Time Bases or rorc - 1 for Offset Time Bases).

If no rate deviation has been calculated, StbM_GetRateDeviation() shall return E_NOT_OK.

Add new requirement [SWS_StbM_00xxxx] in chapter 7.3.8.3 after SWS_StbM_00396
- For Time Bases with StbMSynchronizedTimeBaseIdentifier 32 to 127 (ECUC_StbM_00021) and
- for Time Bases with StbMSynchronizedTimeBaseIdentifier 0 to 31 and StbMIsSystemWideGlobalTimeMaster = True (ECUC_StbM_00036),
the StbM shall return on invocation of StbM_GetRateDeviation() the rate deviation that has been set by StbM_SetRateCorrection() for that Time Base.

If no rate deviation has been set, StbM_GetRateDeviation() shall return E_NOT_OK.
–Last change on issue 77938 comment 9–

BW-C-Level:

Application	**Specification**	**Bus**
1	4	1

1.45 Specification Item SWS_StbM_00409

Trace References:

SRS_StbM_20056

Content:

Name	StbM_StartTimerStbM_StartTimer	
Comment	Interface, which starts a timer for a Time Base	
IsService	true	
Variation	{ecuc(StbM/StbMSynchronizedTimeBase/StbMSynchronizedTimeBaseIdentifier)} < 128	
Possible Errors	0	E_OK
	1	E_NOT_OK

Operations:

StartTimerStbM_StartTimer.StartTimer			
Comments	Starts a StbM internal timer, which expires at the given expireTime and which triggers a time notification callback.		
Variation	–		
Parameters	expireTimeStbM_StartTimer.StartTimer.expireTime	Comment	–
		Type	StbM_TimeStampType
		Variation	–
		Direction	IN

StartTimerStbM_StartTimer.StartTimer		
Possible Errors	E_OK	Operation successful
	E_NOT_OK	Operation failed

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76895: Naming of ServiceInterfaces

  Problem description:

  The names of service interfaces in chapter 8.2.3 and 8.2.4 is inconsistent. In some cases the prefix StbM is used:

  * StbM_StartTimer
  * StbM_MeasurementNotification_TB_Name

  and in some cases it is not used:

  * StatusNotification
  * GlobalTime_Master_Name
  * GlobalTime_Slave_Name
  * TimeNotification

  In my understanding the prefix should be used consistently, i.e. the prefix should **not** exist in the shortName of any service interface.

  Agreed solution:

  remove StbM_ prefix from interface names
  - StbM_StartTimer
  - in SWS_StbM_00409 ("Name" field)
  - in SWS_StbM_00275 ("Comments" fields)
  - in SWS_StbM_91004 ("Interface" field)

  and
  - StbM_MeasurementNotification_TB_Name
  - in SWS_StbM_00339 ("Name" field)
  - in SWS_StbM_00387 ("Name" field + "Interface" field)
  –Last change on issue 76895 comment 2–

  BW-C-Level:

Application	Specification	Bus
4	4	1

1.46 Specification Item SWS_StbM_00412

Trace References:

SRS_StbM_20065

Content:

The StbM shall apply Rate Correction to For a Synchronized Time Base the StbM shall use rrc = 1, if a valid correction rate (rrc) has been calculated.

The not yet been calculated or is not being calculated (refer [SWS_StbMshall apply Rate Correction to _00377]) but shall be applied (refer [SWS_StbM_00355] and [SWS_StbM_00354]).

For an Offset Time Base the StbM shall use rorc = 1, if a valid correction rate (rorc) has been calculatednot yet been calculated or is not being calculated (refer [SWS_StbM_00377]) but shall be applied.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77376: [StbM]: Clarification required on how to apply Rate Correction

  Problem description:

  As per the requirement [SWS_StbM_00412]:
  The StbM shall apply Rate Correction to a Synchronized Time Base, if a valid correction rate (rrc) has been calculated.
  The StbM shall apply Rate Correction to an Offset Time Base, if a valid correction rate (rorc) has been calculated.

  Is given below understanding is correct?
  Rate correction of Synchronized and Offset time bases has to be done continuously by multiplying with corresponding Rate Deviation as a correction factor to determine the Rate Corrected Local Instance of time.

  Agreed solution:

  Rate Correction of Offset Time Bases for Time Slaves part (refer to comment # 4):

  [SWS_StbM_00xxx]
  For Offset Time Bases the StbM shall calculate the rate-corrected offset value of the local instance of the Time Base as:

  TOL = TOG + ((TV TVSync) * rorc

  Where:

- rorc = Rate correction value of the Offset Time Base
- TOL = Current rate corrected offset value of the local instance of the Offset Time Base.
- TOG = newly received Offset Time Base value
- TV = Current value of the Virtual Local Time
- TVSync = Value of the Virtual Local Time when Offset Time Base value is newly received from master

This correction shall be done whenever the time is read in the scope of these functions:
- StbM_GetCurrentTime()
- StbM_GetCurrentTimeExtended()
This correction shall also be done when the StbM needs to determine the time of the local instance of the Time Base.
(SRS_StbM_20065)

Change [SWS_StbM_00360] from

At the end of a Rate Correction measurement, the StbM shall calculate the rate (rorc) for Offset Time Bases as shown:

rorc = ((TSStop - TSStart) + (TOStop - TOStart)) / (TSStop - TSStart)

With:
- rorc = Rate deviation of the Offset Time Base in regards to the associated Synchronized Time Base
- TSStop Current corrected time provided by the local instance of the associated Time Base
- TOStop Current Offset value of the Offset Time Base
- TSStart Corrected time provided by the local instance of the associated Time Base
- TOStart Offset value of the Offset Time Base
to
At the end of a Rate Correction measurement, the StbM shall calculate the rate (rorc) for Offset Time Bases as shown:

rorc = (TOStop - TOStart)) / (TVStop - TVStart)

With:
- rorc = Rate correction value of the Offset Time Base
- TVStop Current time of the Virtual Local Time of the related Synchronized Time Base
- TOStop Current Offset value of the Offset Time Base
- TVStart Value of the Virtual Local Time Base of the related Synchronized Time

Base at the start of the measurement
- TOStart Offset value of the Offset Time Base at the start of the measurement

Add note below [SWS_StbM_00360]:
Note: To determine the resulting rate deviation the value 1 has to be subtracted from rorc.

Change [SWS_StbM_00363]from

At the end of the Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TSStop Current corrected time provided by the local instance of the associated Time Base
- TOStop Current Offset of the Offset Time Base given as function parameter

to

At the end of the Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TVStop Current time of the Virtual Local Time of the related Synchronized Time Base
- TOStop Current Offset value of the Offset Time Base given as function parameter

Change [SWS_StbM_00365] from

At the start of a Rate correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TSStart Current corrected time provided by the local instance of the associated Time Base
- TOStart Current Offset of the Offset Time Base given as function parameter

to

At the start of a Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TVStart Value of the Virtual Local Time Base of the related Synchronized Time Base at the start of the measurement
- TOStart Current Offset value of the Offset Time Base given as function parameter

Change [SWS_StbM_00412]from
The StbM shall apply Rate Correction to a Synchronized Time Base, if a valid correction rate (rrc) has been calculated.
The StbM shall apply Rate Correction to an Offset Time Base, if a valid correction rate (rorc) has been calculated.

to
The StbM shall use rrc = 1 for a Synchronized Time Base, if a valid correction rate (rrc) has not yet been calculated or is not being calculated (refer [SWS_StbM_00377]) but shall be applied (refer [SWS_StbM_00355], [SWS_StbM_00354]).
The StbM shall use rorc = 1 for an Offset Time Base, if a valid correction rate (rorc) has not yet been calculated or is not being calculated (refer [SWS_StbM_00377]) but shall be applied.

Rate Correction for Global Time Masters part (refer to comment # 5):

Add a new requirement to chapter 7.3.8.3 after SWS_StbM_00396

[SWS_StbM_00xxx]
The StbM shall calculate the rate corrected time (TL) of its local instance of the Time Base as:

TL = TGSync + (TV - TVSync) * rrc

(Where:
- TV = Current value of the Virtual Local Time
- TVSync = Value of the Virtual Local Time at the synchronization event
- TGSync = Value of the Global Time at the synchronization event
- rrc = Rate for correcting the Time Base
(SRS_StbM_20065)

Note: Synchronization events for determining TVSync and TGSync are invocations of StbM_SetRateCorrection(), StbM_SetGlobalTime(), StbM_UpdateGlobalTime(), and the initialization of the StbM. StbM_SetOffset() is an additional synchronization event for Offset Time Bases. In case of StbM_SetRateCorrection() TGSync is calculated as TL based on the previous TGSync value. Additional events might need to be considered for synchronization (e.g., overflow of underlying HW timers). Those should however occur not too often to avoid worsening the precision, e.g., by

rounding effects.
–Last change on issue 77376 comment 18–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.47 Specification Item SWS_StbM_00414

Trace References:

SRS_StbM_20064

Content:

StbM_GetTimeBaseUpdateCounter() shall return the value of the timeBaseUpdateCounter of the corresponding Time Base.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77053: StbM_GetTimeBaseUpdateCounter return value undefined for invalid timeBaseID

  Problem description:

  The API function StbM_GetTimeBaseUpdateCounter is only defined in chapter 8 ([SWS_StbM_00347]).
  The function shall return the value of timeBaseUpdateCounter which has the value range 0 to 255 ([SWS_StbM_00351]).

  [SWS_StbM_00348] states that a DET error shall be raised if the function StbM_GetTimeBaseUpdateCounter is called with an invalid value of timeBaseID.

  However, it is undefined which value shall be returned in that case.

  Since the return value type is uint8 and since the range is from 0 to 255 there are no values available that can be exclusively used for that purpose.

  Since the default value is 0, I recommend to use the value 0 as well, which is better than just leaving it unspecified and up to the implementer.

  Besides there should be a SWS item in chapter 7 describing the function and its return value in case of an error.

Remark: this issue was identified by RB.

Agreed solution:

Add the following immediately before SWS_StbM_00351
"..
StbM_GetTimeBaseUpdateCounter allows the Timesync Modules to detect whether a Time Base should be transmitted immediately in the subsequent <Bus>TSyn_MainFunction() cycle.

[SWS_StbM_00xxx]
StbM_GetTimeBaseUpdateCounter shall return the value of the timeBaseUpdateCounter of the corresponding Time Base.
(SRS_StbM_20064)
.."
–Last change on issue 77053 comment 2–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.48 Specification Item SWS_StbM_00415

Trace References:

SRS_BSW_00386, SRS_BSW_00323

Content:

If the switch StbMDevErrorDetect (ECUC_StbM_00012 :) is set to TRUE, StbM_GetMasterConfig() shall report to DET the development error STBM_E_PARAM, if called with a parameter timeBaseId, which

- is not configured or
- is within the reserved value range.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77940: [StbM]: Addition of a requirement is needed to check the range of 'timeBaseId' in API 'StbM_GetMasterConfig()'

  Problem description:

As per the requirement [SWS_StbM_00408] StbM_GetMasterConfig() shall return the value of the configuration parameter 'StbMIsSystemWideGlobalTimeMaster' for the Time Base 'timeBaseId'.

And [SWS_StbM_91002] StbM_GetMasterConfig() doesn't specify the valid range of 'timeBaseId'.

Agreed solution:

Add a new requirement below the requirement [SWS_StbM_91002] as:
[SWS_StbM_00xxx]
If the switch StbMDevErrorDetect (ECUC_StbM_00012 :) is set to TRUE, StbM_GetMasterConfig() shall report to DET the development error STBM_E_PARAM, if called with a parameter timeBaseID, which
-is not configured or
-is within the reserved value range.

[SWS_StbM_00xxx]
If the switch StbMDevErrorDetect (ECUC_StbM_00012 :) is set to TRUE, StbM_GetMasterConfig() shall report to DET the development error STBM_E_PARAM_POINTER, if called with a NULL pointer for parameter masterConfig.
–Last change on issue 77940 comment 4–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.49 Specification Item SWS_StbM_00416

Trace References:

SRS_BSW_00386, SRS_BSW_00323

Content:

If the switch StbMDevErrorDetect (ECUC_StbM_00012 :) is set to TRUE, StbM_GetMasterConfig() shall report to DET the development error STBM_E_PARAM_POINTER, if called with a NULL pointer for parameter masterConfig.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77940: [StbM]: Addition of a requirement is needed to check the range of 'timeBaseId' in API 'StbM_GetMasterConfig()'

Problem description:

As per the requirement [SWS_StbM_00408] StbM_GetMasterConfig() shall return the value of the configuration parameter 'StbMIsSystemWideGlobalTimeMaster' for the Time Base 'timeBaseId'.

And [SWS_StbM_91002] StbM_GetMasterConfig() doesn't specify the valid range of 'timeBaseId'.

Agreed solution:

Add a new requirement below the requirement [SWS_StbM_91002] as:
[SWS_StbM_00xxx]
If the switch StbMDevErrorDetect (ECUC_StbM_00012 :) is set to TRUE, StbM_GetMasterConfig() shall report to DET the development error STBM_E_PARAM, if called with a parameter timeBaseID, which
-is not configured or
-is within the reserved value range.

[SWS_StbM_00xxx]
If the switch StbMDevErrorDetect (ECUC_StbM_00012 :) is set to TRUE, StbM_GetMasterConfig() shall report to DET the development error STBM_E_PARAM_POINTER, if called with a NULL pointer for parameter masterConfig.
–Last change on issue 77940 comment 4–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.50 Specification Item SWS_StbM_00417

Trace References:

SRS_BSW_00386, SRS_BSW_00323

Content:

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.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- 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

1.51 Specification Item SWS_StbM_00418

Trace References:

SRS_BSW_00386, SRS_BSW_00323

Content:

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.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- 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

1.52 Specification Item SWS_StbM_00419

Trace References:

SRS_StbM_20007, SRS_StbM_20025

Content:

The StbM shall check for a timeout condition of a Time Base within StbM_MainFunction() and all API functions, which return the Time Base Status (e.g. StbM_GetTimeBaseStatus() or StbM_GetCurrentTime())

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77671: Various Time Base Status clarifications

  Problem description:

  1.) Both, [SWS_StbM_00183] and [SWS_StbM_00187]
  specify that "If the timeout occurs, the StbM shall set the TIMEOUT bit within timeBaseStatus of the Time Base."

  Since the specified timeout time expires asynchronous
  to any StbM invocation it should be defined when the TIMEOUT bit shall be set.
  Since there is no requirement to set up a dedicated timer outside the StbM which shall call the StbM on expiration there are in principle two occasions where the TIMEOUT bit can be set:
  1.) Inside the cyclically called StbM_MainFunction
  2.) Within every function that returns the Time Base Status, e.g. StbM_GetTimeBaseStatus(), StbM_GetCurrentTime().

  Doing it only in a function that returns the Time Base Status is not sufficient if EV_TIMEOUT_OCCURED was configured as a Status Event.
  Doing it only inside the StbM_MainFunction leads to a slightly deferred TIMEOUT detection (and the points in time at which multiple ECUs will detect a timeout will differ more from each other).

  It is also undefined which time shall be used to determine the timeout (either the Virtual Local Time or the Local Instance of the Synchronized Time Base itself).

  2.) [SWS_StbM_00374] specifies that "For Rate Correction measurements, the StbM shall evaluate state changes of the TIMEOUT flag. The StbM shall discard measurements when the state changes during a measurement."

This requirement is overspecified and thus confusing since it is sufficient to only discard the measurement if the TIMEOUT flag gets set.
If a timeout occurs and measurement is stopped, a new measurement is only started by "reception of time values for Synchronized or Offset Time Bases." ([SWS_StbM_00371]), which will then reset the TIMEOUT flag.

3.) [SWS_StbM_00376] specifies that "the StbM shall evaluate state changes of the TIMELEAP_FUTURE and TIMELEAP_PAST flags during measurements. The StbM shall discard the measurement if the flag state changes."

This requirement is overspecified and thus confusing since [SWS_StbM_00373] requires that the StbM shall not start a Rate Correction measurement when either of the flags is set.

Therefore it is sufficient to only discard the measurement if any of the TIMELEAP flags gets set.

Agreed solution:

For 1)
In SWS_StbM_00183 and SWS_StbM_00187 replace the sentence:
The timeout StbMSyncLossTimeout (ECUC_StbM_00028 :) shall be measured from last invocation of StbM_BusSetGlobalTime()
by
The timeout StbMSyncLossTimeout (ECUC_StbM_00028 :) shall be measured based on the Virtual Local Time from last invocation of StbM_BusSetGlobalTime()

Add a new requirement + note after SWS_StbM_00187

SWS_StbM_xxxx: The StbM shall check for a timeout condition of a Time Base within StbM_MainFunction() and all API functions, which return the Time Base Status (e.g. StbM_GetTimeBaseStatus() or StbM_GetCurrentTime())

Note: Since a Status Notification is triggered inside StbM_MainFunction(), the other functions like e.g StbM_GetTimeBaseStatus() might detect a timeout condition sooner than the corresponding Status Notification is actually triggered. Such a delayed Status Notification is considered acceptable.

For 2)
Change SWS_StbM_00374 from:
"..., the StbM shall evaluate state changes of the TIMEOUT flag.
The StbM shall discard measurements when the state changes during a measure-

ment."
to:
"..., the StbM shall evaluate the TIMEOUT flag.
The StbM shall discard the measurement if the flag equals Set."

For 3)
Change SWS_StbM_00376
from:
"..., the StbM shall evaluate state changes of the TIMELEAP_FUTURE and TIMELEAP_PAST flags during measurements.
The StbM shall discard the measurement if the flag state changes."
to:
"..., the StbM shall evaluate the TIMELEAP_FUTURE and TIMELEAP_PAST flags during measurements.
The StbM shall discard the measurement if any of the flags equals Set."

Add a note after SWS_StbM_00362:
"A newly calculated Rate Correction rrc is only applied to following time calculations."
–Last change on issue 77671 comment 14–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.53 Specification Item SWS_StbM_00422

Trace References:

SRS_StbM_20065

Content:

- For Time Bases with StbMSynchronizedTimeBaseIdentifier 32 to 127 (ECUC_StbM_00021 :) and
- for Time Bases with StbMSynchronizedTimeBaseIdentifier 0 to 31 and StbMIsSystemWideGlobalTimeMaster equals True (ECUC_StbM_00036 :),

the StbM shall return on invocation of StbM_GetRateDeviation() the rate deviation that has been set by StbM_SetRateCorrection() for that Time Base.

If no rate deviation has been set, StbM_GetRateDeviation() shall return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77938: [StbM]: Correction required in DET error reporting for the range of 'timeBaseID' in API 'StbM_GetRateDeviation()'

  Problem description:

  As per the requirements [SWS_StbM_00378] and [SWS_StbM_00379], StbM_GetRateDeviation() shall report to DET the development error STBM_E_PARAM, if called with a parameter 'timeBaseID', which refers to an Offset Time Base.

  But as per the requirements [SWS_StbM_00390] and [SWS_StbM_00391], StbM_SetRateCorrection() shall report to DET the development error STBM_E_PARAM, if called with a parameter 'timeBaseID', which
  -is not configured or
  -is within the reserved value range.

  The use case of both the APIs (StbM_GetRateDeviation() and StbM_SetRateCorrection()) as mentioned in 'Note' in requirement [SWS_StbM_00396], is applicable for 'Offset Time Base' also.

  So the API StbM_GetRateDeviation() shouldn't report to DET the development error STBM_E_PARAM, if called with a parameter 'timeBaseID', which refers to an Offset Time Base as mentioned in requirement [SWS_StbM_00379].

  Agreed solution:

  Change,
  From:
  [SWS_StbM_00379]
  If the switch StbMDevErrorDetect (ECUC_StbM_00012 :) is set to TRUE, StbM_GetRateDeviation() shall report to DET the development error STBM_E_PARAM, if called with a parameter timeBaseID, which
  -is not configured or
  -refers to an Offset Time Base or
  -is within the reserved value range.

  To:
  [SWS_StbM_00379]
  If the switch StbMDevErrorDetect (ECUC_StbM_00012 :) is set to TRUE, StbM_GetRateDeviation() shall report to DET the development error STBM_E_PARAM, if called with a parameter timeBaseID, which
  -is not configured or
  -is within the reserved value range.

Change [SWS_StbM_00397] to
For Time Bases with StbMSynchronizedTimeBaseIdentifier 0 to 31 (ECUC_StbM_00021) and StbMIsSystemWideGlobalTimeMaster = False (ECUC_StbM_00036), the StbM shall return on invocation of StbM_GetRateDeviation() the rate deviation, which has been calculated for that Time Base (i.e., rrc -1 for Synchronized Time Bases or rorc - 1 for Offset Time Bases).

If no rate deviation has been calculated, StbM_GetRateDeviation() shall return E_NOT_OK.

Add new requirement [SWS_StbM_00xxxx] in chapter 7.3.8.3 after SWS_StbM_00396
- For Time Bases with StbMSynchronizedTimeBaseIdentifier 32 to 127 (ECUC_StbM_00021) and
- for Time Bases with StbMSynchronizedTimeBaseIdentifier 0 to 31 and StbMIsSystemWideGlobalTimeMaster = True (ECUC_StbM_00036),
the StbM shall return on invocation of StbM_GetRateDeviation() the rate deviation that has been set by StbM_SetRateCorrection() for that Time Base.

If no rate deviation has been set, StbM_GetRateDeviation() shall return E_NOT_OK.
–Last change on issue 77938 comment 9–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.54 Specification Item SWS_StbM_00423

Trace References:

SRS_StbM_20065

Content:

For Offset Time Bases the StbM shall calculate the rate-corrected offset value of the local instance of the Time Base as:

TOL = TOG + (TV - TVSync) * rorc

With:

- rorc = Rate correction value of the Offset Time Base

- TOL = Current rate corrected offset value of the local instance of the Offset Time Base.
- TOG = newly received Offset Time Base value
- TV = Current value of the Virtual Local Time
- TVSync = Value of the Virtual Local Time when Offset Time Base value is newly received from master

This correction shall be done whenever the time is read in the scope of these functions:

- StbM_GetCurrentTime()
- StbM_GetCurrentTimeExtended()

This correction shall also be done when the StbM needs to determine the time of the local instance of the Time Base.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77376: [StbM]: Clarification required on how to apply Rate Correction

  Problem description:

  As per the requirement [SWS_StbM_00412]:
  The StbM shall apply Rate Correction to a Synchronized Time Base, if a valid correction rate (rrc) has been calculated.
  The StbM shall apply Rate Correction to an Offset Time Base, if a valid correction rate (rorc) has been calculated.

  Is given below understanding is correct?
  Rate correction of Synchronized and Offset time bases has to be done continuously by multiplying with corresponding Rate Deviation as a correction factor to determine the Rate Corrected Local Instance of time.

  Agreed solution:

  Rate Correction of Offset Time Bases for Time Slaves part (refer to comment # 4):

  [SWS_StbM_00xxx]
  For Offset Time Bases the StbM shall calculate the rate-corrected offset value of the local instance of the Time Base as:

  TOL = TOG + ((TV TVSync) * rorc

  Where:
  - rorc = Rate correction value of the Offset Time Base

- TOL = Current rate corrected offset value of the local instance of the Offset Time Base.
- TOG = newly received Offset Time Base value
- TV = Current value of the Virtual Local Time
- TVSync = Value of the Virtual Local Time when Offset Time Base value is newly received from master

This correction shall be done whenever the time is read in the scope of these functions:
- StbM_GetCurrentTime()
- StbM_GetCurrentTimeExtended()
This correction shall also be done when the StbM needs to determine the time of the local instance of the Time Base.
(SRS_StbM_20065)

Change [SWS_StbM_00360] from

At the end of a Rate Correction measurement, the StbM shall calculate the rate (rorc) for Offset Time Bases as shown:

rorc = ((TSStop - TSStart) + (TOStop - TOStart)) / (TSStop - TSStart)

With:
- rorc = Rate deviation of the Offset Time Base in regards to the associated Synchronized Time Base
- TSStop Current corrected time provided by the local instance of the associated Time Base
- TOStop Current Offset value of the Offset Time Base
- TSStart Corrected time provided by the local instance of the associated Time Base
- TOStart Offset value of the Offset Time Base
to
At the end of a Rate Correction measurement, the StbM shall calculate the rate (rorc) for Offset Time Bases as shown:

rorc = (TOStop - TOStart)) / (TVStop - TVStart)

With:
- rorc = Rate correction value of the Offset Time Base
- TVStop Current time of the Virtual Local Time of the related Synchronized Time Base
- TOStop Current Offset value of the Offset Time Base
- TVStart Value of the Virtual Local Time Base of the related Synchronized Time Base at the start of the measurement

- TOStart Offset value of the Offset Time Base at the start of the measurement

Add note below [SWS_StbM_00360]:
Note: To determine the resulting rate deviation the value 1 has to be subtracted from rorc.

Change [SWS_StbM_00363]from

At the end of the Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TSStop Current corrected time provided by the local instance of the associated Time Base
- TOStop Current Offset of the Offset Time Base given as function parameter

to

At the end of the Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TVStop Current time of the Virtual Local Time of the related Synchronized Time Base
- TOStop Current Offset value of the Offset Time Base given as function parameter

Change [SWS_StbM_00365] from

At the start of a Rate correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TSStart Current corrected time provided by the local instance of the associated Time Base
- TOStart Current Offset of the Offset Time Base given as function parameter

to

At the start of a Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TVStart Value of the Virtual Local Time Base of the related Synchronized Time Base at the start of the measurement
- TOStart Current Offset value of the Offset Time Base given as function parameter

Change [SWS_StbM_00412]from
The StbM shall apply Rate Correction to a Synchronized Time Base, if a valid correction rate (rrc) has been calculated.
The StbM shall apply Rate Correction to an Offset Time Base, if a valid correction rate (rorc) has been calculated.

to
The StbM shall use rrc = 1 for a Synchronized Time Base, if a valid correction rate (rrc) has not yet been calculated or is not being calculated (refer [SWS_StbM_00377]) but shall be applied (refer [SWS_StbM_00355], [SWS_StbM_00354]).
The StbM shall use rorc = 1 for an Offset Time Base, if a valid correction rate (rorc) has not yet been calculated or is not being calculated (refer [SWS_StbM_00377]) but shall be applied.

Rate Correction for Global Time Masters part (refer to comment # 5):

Add a new requirement to chapter 7.3.8.3 after SWS_StbM_00396

[SWS_StbM_00xxx]
The StbM shall calculate the rate corrected time (TL) of its local instance of the Time Base as:

TL = TGSync + (TV - TVSync) * rrc

(Where:
- TV = Current value of the Virtual Local Time
- TVSync = Value of the Virtual Local Time at the synchronization event
- TGSync = Value of the Global Time at the synchronization event
- rrc = Rate for correcting the Time Base
(SRS_StbM_20065)

Note: Synchronization events for determining TVSync and TGSync are invocations of StbM_SetRateCorrection(), StbM_SetGlobalTime(), StbM_UpdateGlobalTime(), and the initialization of the StbM. StbM_SetOffset() is an additional synchronization event for Offset Time Bases. In case of StbM_SetRateCorrection() TGSync is calculated as TL based on the previous TGSync value. Additional events might need to be considered for synchronization (e.g., overflow of underlying HW timers). Those should however occur not too often to avoid worsening the precision, e.g., by rounding effects.
–Last change on issue 77376 comment 18–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.55 Specification Item SWS_StbM_00424

Trace References:

SRS_StbM_20065

Content:

The StbM shall calculate the rate corrected time (TL) of its local instance of the Time Base as:

TL = TGSync + (TV - TVSync) * rrc

With:

- TV = Current value of the Virtual Local Time
- TVSync = Value of the Virtual Local Time at the synchronization event
- TGSync = Value of the Global Time at the synchronization event
- rrc = Rate for correcting the Time Base

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77376: [StbM]: Clarification required on how to apply Rate Correction

  Problem description:

  As per the requirement [SWS_StbM_00412]:
  The StbM shall apply Rate Correction to a Synchronized Time Base, if a valid correction rate (rrc) has been calculated.
  The StbM shall apply Rate Correction to an Offset Time Base, if a valid correction rate (rorc) has been calculated.

  Is given below understanding is correct?
  Rate correction of Synchronized and Offset time bases has to be done continuously by multiplying with corresponding Rate Deviation as a correction factor to determine the Rate Corrected Local Instance of time.

  Agreed solution:

Rate Correction of Offset Time Bases for Time Slaves part (refer to comment # 4):

[SWS_StbM_00xxx]
For Offset Time Bases the StbM shall calculate the rate-corrected offset value of the local instance of the Time Base as:

TOL = TOG + ((TV TVSync) * rorc

Where:
- rorc = Rate correction value of the Offset Time Base
- TOL = Current rate corrected offset value of the local instance of the Offset Time Base.
- TOG = newly received Offset Time Base value
- TV = Current value of the Virtual Local Time
- TVSync = Value of the Virtual Local Time when Offset Time Base value is newly received from master

This correction shall be done whenever the time is read in the scope of these functions:
- StbM_GetCurrentTime()
- StbM_GetCurrentTimeExtended()
This correction shall also be done when the StbM needs to determine the time of the local instance of the Time Base.
(SRS_StbM_20065)

Change [SWS_StbM_00360] from

At the end of a Rate Correction measurement, the StbM shall calculate the rate (rorc) for Offset Time Bases as shown:

rorc = ((TSStop - TSStart) + (TOStop - TOStart)) / (TSStop - TSStart)

With:
- rorc = Rate deviation of the Offset Time Base in regards to the associated Synchronized Time Base
- TSStop Current corrected time provided by the local instance of the associated Time Base
- TOStop Current Offset value of the Offset Time Base
- TSStart Corrected time provided by the local instance of the associated Time Base
- TOStart Offset value of the Offset Time Base
to
At the end of a Rate Correction measurement, the StbM shall calculate the rate (rorc) for Offset Time Bases as shown:

rorc = (TOStop - TOStart)) / (TVStop - TVStart)

With:
- rorc = Rate correction value of the Offset Time Base
- TVStop Current time of the Virtual Local Time of the related Synchronized Time Base
- TOStop Current Offset value of the Offset Time Base
- TVStart Value of the Virtual Local Time Base of the related Synchronized Time Base at the start of the measurement
- TOStart Offset value of the Offset Time Base at the start of the measurement

Add note below [SWS_StbM_00360]:
Note: To determine the resulting rate deviation the value 1 has to be subtracted from rorc.

Change [SWS_StbM_00363]from

At the end of the Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TSStop Current corrected time provided by the local instance of the associated Time Base
- TOStop Current Offset of the Offset Time Base given as function parameter

to

At the end of the Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TVStop Current time of the Virtual Local Time of the related Synchronized Time Base
- TOStop Current Offset value of the Offset Time Base given as function parameter

Change [SWS_StbM_00365] from

At the start of a Rate correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TSStart Current corrected time provided by the local instance of the associated Time Base
- TOStart Current Offset of the Offset Time Base given as function parameter

to

At the start of a Rate Correction measurement of Offset Time Bases, the StbM shall take the following time-snapshots in the scope of the function StbM_BusSetGlobalTime():
- TVStart Value of the Virtual Local Time Base of the related Synchronized Time Base at the start of the measurement
- TOStart Current Offset value of the Offset Time Base given as function parameter

Change [SWS_StbM_00412]from
The StbM shall apply Rate Correction to a Synchronized Time Base, if a valid correction rate (rrc) has been calculated.
The StbM shall apply Rate Correction to an Offset Time Base, if a valid correction rate (rorc) has been calculated.

to
The StbM shall use rrc = 1 for a Synchronized Time Base, if a valid correction rate (rrc) has not yet been calculated or is not being calculated (refer [SWS_StbM_00377]) but shall be applied (refer [SWS_StbM_00355], [SWS_StbM_00354]).
The StbM shall use rorc = 1 for an Offset Time Base, if a valid correction rate (rorc) has not yet been calculated or is not being calculated (refer [SWS_StbM_00377]) but shall be applied.

Rate Correction for Global Time Masters part (refer to comment # 5):

Add a new requirement to chapter 7.3.8.3 after SWS_StbM_00396

[SWS_StbM_00xxx]
The StbM shall calculate the rate corrected time (TL) of its local instance of the Time Base as:

TL = TGSync + (TV - TVSync) * rrc

(Where:
- TV = Current value of the Virtual Local Time
- TVSync = Value of the Virtual Local Time at the synchronization event
- TGSync = Value of the Global Time at the synchronization event
- rrc = Rate for correcting the Time Base
(SRS_StbM_20065)

Note: Synchronization events for determining TVSync and TGSync are invocations of StbM_SetRateCorrection(), StbM_SetGlobalTime(), StbM_UpdateGlobalTime(), and the initialization of the StbM. StbM_SetOffset() is an additional synchronization event for Offset Time Bases. In case of StbM_SetRateCorrection() TGSync is calculated as TL based on the previous TGSync value. Additional events might need to be considered for synchronization (e.g., overflow of underlying HW timers). Those should however occur not too often to avoid worsening the precision, e.g., by rounding effects.
–Last change on issue 77376 comment 18–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.56 Specification Item SWS_StbM_00425

Trace References:

SRS_StbM_20025

Content:

For Time Slaves and Time Gateways of Synchronized Time Bases StbM_GetTimeLeap() shall return the time difference between the newly received and the current Time Base value, i.e. TG - TL, which is calculated upon each, except the very first, valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

With

- TL = Current value of the local instance of the Time Base (before newly received Time Base value is applied)
- TG = Newly received Time Base value

For Time Slaves and Time Gateways of Offset Time Bases StbM_GetTimeLeap() shall return the time difference between the newly received and the current Time Base offset value, i.e. TOG - TOL, which is calculated upon each, except the very first, valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

With

- TOL = Current offset value of the local instance of the Time Base (before newly received Time Base offset value is applied)
- TOG = Newly received Time Base offset value

If the calculated time difference exceeds the value range of the timeJump parameter of StbM_GetTimeLeap() the returned time difference shall be limited to

either the maximum negative or the maximum positive value of the type of timeJump (refer to StbM_TimeDiffType).

StbM_GetTimeLeap() shall return E_NOT_OK until the second valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #78023: [StbM]: Clarification required on return value of the API 'StbM_GetTimeLeap()'

  Problem description:

  As per the requirement [SWS_StbM_00267] StbM_GetTimeLeap() returns value of Time Leap, if StbMTimeLeapFuture/PastThreshold is exceeded.

  The value of Time Leap returned in 'timeJump' in API StbM_GetTimeLeap() would be the value which is calculated in API StbM_BusSetGlobalTime() only?

  And if StbMTimeLeapFuture/PastThreshold is not exceeded then, we are assuming that StbM_GetTimeLeap() shall return E_NOT_OK.

  Agreed solution:

  Add a new requirement in ch. 7 for clarification after SWS_StbM_00305:

  SWS_StbM_00xxx:
  For Time Slaves and Time Gateways of Synchronized Time Bases StbM_GetTimeLeap() shall return the time difference between the newly received and the current Time Base value, i.e. TG - TL, which is calculated upon each, except the very first, valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

  With
  - TL = Current value of the local instance of the Time Base (before newly received Time Base value is applied)
  - TG = Newly received Time Base value

  For Time Slaves and Time Gateways of Offset Time Bases StbM_GetTimeLeap() shall return the time difference between the newly received and the current Time Base offset value, i.e. TOG - TOL, which is calculated upon each, except the very first, valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

With
- TOL = Current offset value of the local instance of the Time Base (before newly received Time Base offset value is applied)
- TOG = Newly received Time Base offset value

If the calculated time difference exceeds the value range of the timeJump parameter of StbM_GetTimeLeap() the returned time difference shall be limited to either the maximum negative or the maximum positive value of the type of timeJump [SWS_StbM_00300].

StbM_GetTimeLeap() shall return E_NOT_OK until the second valid invocation of StbM_BusSetGlobalTime() for the corresponding Time Base.

In Comments Field for GetTimeLeap Operation in SWS_StbM_00247
remove subclause ", if StbMTimeLeapFuture/PastThreshold has been exceeded"

In Description Field for API StbM_GetTimeLeap in SWS_StbM_00267
remove subclause ", if StbMTimeLeapFuture/PastThreshold is exceeded"

In Description field of [SWS_StbM_00273]
replace "StbM_SetTimer" by "StbM_StartTimer"

In "Parameters (in):" field of [SWS_StbM_00273]
replace
"Actual time value captured, when callback is called by StbM"
by
"Difference time value when callback is called by StbM"

In [SWS_StbM_00271] replace
"actualTime" by "deviationTime"
–Last change on issue 78023 comment 20–

BW-C-Level:

Application	Specification	Bus
4	4	1

1.57 Specification Item SWS_StbM_91001

Trace References:

SRS_StbM_20023

Content:

Name	StbM_MasterConfigTypeStbM_MasterConfigType		
Kind	Enumeration Type		
Derived from	uint8		
Description	This type indicates if an ECU is configured for a system wide master for a given Time Base is available or not.		
Range	STBM_ SYSTEM_WIDE_MASTER_DISABLEDStbM_MasterConfigType.STBM_SYSTEM_WIDE_MASTER_DISABLED	0x00	not configured as System Wide Master
	STBM_SYSTEM_WIDE_MASTER_ENABLEDStbM_MasterConfig Type.STBM_SYSTEM_WIDE_MASTER_ENABLED	0x01	configured as System Wide Master
	Description		
Variation	–		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77871: [StbM] StbM_MasterConfigType should not be of kind Enumeration since Enumeration is not supported by RTE

  Problem description:

  As RTE does not support Enumeration data type , the kind of StbM_MasterConfigType should not be Enumeration.

  Agreed solution:

  SWS_StbM_91001: Change the kind of StbM_MasterConfigType from "Enumeration" to uint8.
  –Last change on issue 77871 comment 2–

  BW-C-Level:

Application	Specification	Bus
1	4	1

1.58 Specification Item SWS_StbM_91004

Trace References:

SRS_StbM_20056

Content:

Name	StartTimer_{TimeBase}_{Customer}StbM.StbM_StartTimer		
Kind	ProvidedPort	Interface	StbM_StartTimer
Description	–		
Port Defined Argument Value(s)	Type	StbM_SynchronizedTimeBaseType	
	Value	{ecuc(StbM/StbMSynchronizedTimeBase/StbMSynchronizedTimeBaseIdentifier.value)}	
	Type	StbM_CustomerIdType	
	Value	{ecuc(StbM/StbMSynchronizedTimeBase/StbMNotificationCustomer/StbMNotificationCustomerId.value)}	

Variation	{ecuc(StbM/StbMSynchronizedTimeBase/StbMSynchronizedTimeBaseIdentifier)} < 128
	TimeBase = {ecuc(StbM/StbMSynchronizedTimeBase.SHORT-NAME)}
	Customer = {ecuc(StbM/StbMSynchronizedTimeBase/StbMNotificationCustomer.SHORT-NAME)}		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76895: Naming of ServiceInterfaces

  Problem description:

  The names of service interfaces in chapter 8.2.3 and 8.2.4 is inconsistent. In some cases the prefix StbM is used:

  * StbM_StartTimer
  * StbM_MeasurementNotification_TB_Name

  and in some cases it is not used:

  * StatusNotification
  * GlobalTime_Master_Name
  * GlobalTime_Slave_Name
  * TimeNotification

  In my understanding the prefix should be used consistently, i.e. the prefix should **not** exist in the shortName of any service interface.

  Agreed solution:

  remove StbM_ prefix from interface names
  - StbM_StartTimer
  - in SWS_StbM_00409 ("Name" field)
  - in SWS_StbM_00275 ("Comments" fields)
  - in SWS_StbM_91004 ("Interface" field)

and
- StbM_MeasurementNotification_TB_Name
- in SWS_StbM_00339 ("Name" field)
- in SWS_StbM_00387 ("Name" field + "Interface" field)
–Last change on issue 76895 comment 2–

BW-C-Level:

Application	Specification	Bus
4	4	1