Disclaimer

This specification and the material contained in it, as released by AUTOSAR is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the specification.

The material contained in this specification is protected by copyright and other types of Intellectual Property Rights. The commercial exploitation of the material contained in this specification requires a license to such Intellectual Property Rights.

This specification may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the specification may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The AUTOSAR specifications have been developed for automotive applications only. They have neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Advice for users

AUTOSAR Specification Documents may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the Specification Documents for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such Specification Documents, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.
# Table of Contents

1  Acronyms and abbreviations................................................................. 6  
2  Related Documentation........................................................................... 7  
   2.1  Input documents.................................................................................. 7  
3  Scope......................................................................................................... 8  
4  RS_RBF_01660 - Global Time Synchronization over CAN............................ 9  
   4.1  General Test Objective and Approach................................................. 9  
   4.1.1  Test System...................................................................................... 10  
   4.1.2  Test Case Design............................................................................. 13  
4.2  Configuration requirements.................................................................... 14  
   4.2.1  Test Configuration.......................................................................... 14  
   4.2.2  Static Configuration......................................................................... 14  
4.3  Re-usable Test Steps............................................................................. 21  
4.4  Test cases............................................................................................... 21  
   4.4.1  [ATS_GTS_01228] Global Time Master: Setting of Global Time base  
  by Active Customer and sending of SYNC frames (CRC_SUPPORTED) over CAN......................................................................................... 21  
   4.4.2  [ATS_GTS_01266] Global Time Master: Handling of SYNC message  
  Confirmation Failures.................................................................................. 25  
   4.4.3  [ATS_GTS_01264] Global Time Master: Setting of Global Time base  
  by Active Customer and sending of offset frames (CRC_NOT_SUPPORTED) over CAN.......................................................... 27  
   4.4.4  [ATS_GTS_01268] Global Time Master: Handling of Offset message  
  Confirmation Failures................................................................................ 29  
   4.4.5  [ATS_GTS_01238] Global Time Master: Handling of time base using  
  NvM (Storage and Retrieve)...................................................................... 31  
   4.4.6  [ATS_GTS_01242] Time Slave: Reception of SYNC frames (CRC_  
  IGNORED) over CAN, Synchronize Local Time Base and share the current  
  time to active customers........................................................................... 33  
   4.4.7  [ATS_GTS_01271] Time Slave: Handling of SYNC message reception  
  timeout (CanTSynGlobalTimeFollowUpTimeout)...................................... 35  
   4.4.8  [ATS_GTS_01273] Time Slave: Handling of SYNC message  
  Sequence Mismatch failures..................................................................... 38  
   4.4.9  [ATS_GTS_01286] Time Slave: Reception of Offset frames (CRC_  
  VALIDATED) over CAN, Synchronize Local Time Base and share the current  
  time to active customers........................................................................... 43  
   4.4.10 [ATS_GTS_01244] Time Slave: Reception of Offset frames (CRC_  
  NOT_VALIDATED) over CAN, Synchronize Local Time Base and share the  
  current time to active customer.................................................................. 45  
   4.4.11 [ATS_GTS_01272] Time Slave: Handling of Offset message reception  
  timeout (CanTSynGlobalTimeFollowUpTimeout)...................................... 46  
   4.4.12 [ATS_GTS_01274] Time Slave: Handling of Offset message  
  Sequence Mismatch failure....................................................................... 49  
   4.4.13 [ATS_GTS_01270] Time Slave: Synchronize Runnable entities to time  
  base........................................................................................................... 53
5 RS_BRF_01660 - Global Time Synchronization over FlexRay ........................... 56
5.1 General Test Objective and Approach .................................................. 56
5.1.1 Test System ..................................................................................... 56
5.1.2 Test Case Design ............................................................................ 59
5.2 Configuration requirements ................................................................. 60
5.2.1 Test Configuration ......................................................................... 60
5.2.2 Static Configuration ....................................................................... 60
5.3 Re-usable Test Steps ........................................................................ 67
5.4 Test Cases ......................................................................................... 67
  5.4.1 [ATS_GTS_01275] Global Time Master: Setting of Global Time base
  and user data by Active Customer and sending of SYNC frames
  (CRC_NOT_SUPPORTED) over FlexRay .................................................. 67
  5.4.2 [ATS_GTS_01287] Global Time Master: Setting of Offset Time base
  by Active Customer and sending Offset frames (CRC_SUPPORTED)
  over FlexRay .................................................................................... 70
  5.4.3 [ATS_GTS_01249] Time Slave: Reception of SYNC frames
  (CRC_VALIDATED) over FlexRay, Synchronize Local Time Base and
  share the current time to active customers ........................................ 72
  5.4.4 [ATS_GTS_01250] Time Slave: Reception of SYNC frames
  (CRC_IGNORERD) over FlexRay, Synchronize Local Time Base and
  share the current time to active customers ......................................... 75
  5.4.5 [ATS_GTS_01256] Time Slave: Handling of SYNC message
  Sequence Mismatch failures ............................................................... 77
  5.4.6 [ATS_GTS_01277] Time Slave: Reception of Offset frames
  (CRC_NOT_VALIDATED) over FlexRay, Synchronize Local Time
  Base and share the current time to active customers ......................... 80
  5.4.7 [ATS_GTS_01280] Time Slave: Handling of Offset message
  Sequence Mismatch failures ............................................................... 82
6 RS_BRF_01660 - Global Time Synchronization over Multiple Bus .................. 85
6.1 General Test Objective and Approach .................................................. 85
6.1.1 Test System ..................................................................................... 86
6.1.2 Test Case Design ............................................................................ 91
6.2 Configuration requirements ................................................................. 93
6.2.1 Test Configuration ......................................................................... 93
6.2.2 Static Configuration ....................................................................... 94
6.3 Re-usable Test Steps ........................................................................ 101
6.4 Test Cases ......................................................................................... 101
  6.4.1 [ATS_GTS_01281] Global Time Master over Multiple Bus(Single Time
  Domain): Setting of Global Time base and user data and sending of
  SYNC frame over CAN and FlexRay .................................................. 101
  6.4.2 [ATS_GTS_01257] Global Time Master over Multiple
  Bus(MultipleTime Domain): Setting of Global Time base and user data
  and sending of SYNC frame over CAN and FlexRay ......................... 106
  6.4.3 [ATS_GTS_01258] Time Gateway- Time Slave on FlexRay and Time
  Master on CAN .................................................................................. 111
6.4.4 [ATS_GTS_01259] Time Gateway - Time Slave on CAN and Time Master on FlexRay
6.4.5 [ATS_GTS_01260] Time Gateway - Time Slave on CAN (Network 1) and Time Master on CAN (Network 2)
# Acronyms and abbreviations

<table>
<thead>
<tr>
<th>Abbreviation / Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT</td>
<td>Acceptance Test</td>
</tr>
<tr>
<td>CAN</td>
<td>Controller Area Network</td>
</tr>
<tr>
<td>ECU</td>
<td>Electronic Control Unit</td>
</tr>
<tr>
<td>LT</td>
<td>Lower Tester</td>
</tr>
<tr>
<td>PCO</td>
<td>Point of Control and Observation</td>
</tr>
<tr>
<td>Rx</td>
<td>Reception</td>
</tr>
<tr>
<td>SUT</td>
<td>System Under Test</td>
</tr>
<tr>
<td>SWC</td>
<td>Software Component</td>
</tr>
<tr>
<td>TCP</td>
<td>Test Coordination Procedures</td>
</tr>
<tr>
<td>Tx</td>
<td>Transmission</td>
</tr>
<tr>
<td>UT</td>
<td>Upper Tester</td>
</tr>
</tbody>
</table>
2 Related Documentation

2.1 Input documents

AUTOSAR_SWS_SynchronizedTimeBaseManager.pdf

AUTOSAR_SWS_TimeSyncOverCAN.pdf

AUTOSAR_SWS_TimeSyncOverFlexRay.pdf

AUTOSAR_SWS_OS.pdf

[5] Specification of Basic Software Mode Manager
AUTOSAR_SWS_BSWModeManager.pdf

AUTOSAR_SWS_CANInterface.pdf

AUTOSAR_SWS_FlexRayInterface.pdf

[8] Specification of RTE
AUTOSAR_SWS_RTE.pdf

[9] Requirements on Synchronized Time-Base Manager
AUTOSAR_SRS_SynchronizedTimeBaseManager.pdf

[10] Requirements on Acceptance Tests
AUTOSAR_ATR_Requirements.pdf

AUTOSAR_RS_Features.pdf

[12] System Template
AUTOSAR_TPS_SystemTemplate.pdf
3 Scope

The following test cases are used to verify the correct behavior of all the Global Time Synchronization features.

Each test case documents for which releases of the AUTOSAR software specification it can be used:

- When test cases are known to be applicable for a release, this is mentioned in the “AUTOSAR Releases” field of the test case specifications.
  You can find a summary of the applicability of all test cases to the software specification releases in the “AUTOSAR_TR_ATSReleaseApplicability” document.
- When test cases are known to require adaptations (in their configuration requirements or test sequences), this is mentioned in the “Needed Adaptation to other Releases” field of the test case specifications.
4 RS_BRF_01660 - Global Time Synchronization over CAN

4.1 General Test Objective and Approach

This Test Specification intends to cover the Global Time Synchronization feature of StbM and CanTSyn as described in the AUTOSAR Feature [RS_BRF_01660].

The tests use a test bench environment and Embedded Software Components that use the feature.

This test case document has been established to cover the following features:

Below Features are not tested in this test suite:

- SGW(Sync to Global Time) Handling [SWS_CanTSyn_00030] and STBM_SYNC_TO_GATEWAY [SWS_StbM_00184]: Feature are tested in the test suite 'Global Time Synchronization over Multiple Bus'.
- Use Ethernet Controller [SWS_StbM_00173]: Testing over Ethernet bus is out of scope of this ATS.

This specification gives the description of required tests environments (test bench, uses case, configuration files) and detailed tests cases for executing tests.
4.1.1 Test System

4.1.1.1 Overview on Architecture

In order to cover the required features / sub-features coverage, the environment has been separated in several use cases.

Test Cases are derived based on below use cases

4.1.1.1.1 UC 01.01: Global Time Master over CAN

SUT acts as Global time Master and sets time base, offset time base and Trigger for transmission of Synchronization over CAN bus.

As shown in below figure, Functionalities of Global Time master of a time domain are tested over CAN in the use case 01.01
4.1.1.1.2 UC 01.02: Time Slave over CAN

SUT acts as Time Slave and Gets time base, offset time base and Synchronizes Local time to Global Time base.

As shown in below figure, Functionalities of Time Slave of a time domain are tested over CAN in the use case 01.02
4.1.1.1.3 UC 01.03: Synchronization of Runnable entities to Global Time

SUT Synchronizes Runnable entities to Global Time base using OS Scheduler (Triggered Customer).

4.1.1.1.4 UC 01.04: Initialization of time base from value stored in Non-volatile Memory

During initialization, SUT updates time base from Non-volatile Memory.

4.1.1.2 Specific Requirements

Not Applicable.

4.1.1.3 Test Coordination Requirements

UC 01.01: Global Time Master over CAN

- Test System (LT <CAN>) shall read the CanTSyn CAN Frames and decode the same as per Frame Format provided in AUTOSAR_SWS_TimeSyncOverCAN.
UC 01.02: Time Slave over CAN

- Test System (LT <CAN>) shall encode the CanTSyn CAN Frames as per Frame Format provided in AUTOSAR_SWS_TimeSyncOverCAN and transmit over bus.

Requirements for CRC Calculation

- Test System (LT <CAN>) shall use the Crc_CalculateCRC8H2F() (Refer AUTOSAR Specification of CRC Routines AUTOSAR_SWS_CRCLibrary.pdf) to calculate the CRC of the Frame. Below are the parameters used for CRC calculation:
  - The CRC start value shall be 0xFF.
  - The CRC final XOR-value shall be 0xFF.
  - The CRC polynomial shall be 0x2F.
  - The DataIDList shall be same as provided in CanTSyn Static Configuration.

4.1.2 Test Case Design

Below diagrams explain test design for different use cases:

Fig: UC 01.01 - Global Time Master over CAN Test Design
4.2 Configuration requirements

The configuration can be divided into two separate parts. The test configuration describes variables used to parameterize the test case. The static configuration describes the necessary settings of the DUT in order to allow a test case to perform.

4.2.1 Test Configuration

Communication data base for CanTSyn is depicted below

<table>
<thead>
<tr>
<th>test configuration parameters</th>
<th>I-Pdu</th>
<th>CAN ID</th>
<th>Tx ECU</th>
<th>Rx ECU</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT_101_Ipdu</td>
<td>101</td>
<td>SUT</td>
<td>Test Bench</td>
<td></td>
</tr>
<tr>
<td>AT_102_Ipdu</td>
<td>102</td>
<td>Test Bench</td>
<td>SUT</td>
<td></td>
</tr>
</tbody>
</table>

4.2.2 Static Configuration

4.2.2.1 Static Configuration Groups

<table>
<thead>
<tr>
<th>SCG_ATS_GlobalTimeSync_Time Master</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Description Parameters</td>
</tr>
<tr>
<td>StbM</td>
</tr>
<tr>
<td>SystemTemplate::GlobalTime::GlobalTimeMaster.isSystemWideGlobalTimeMaster</td>
</tr>
<tr>
<td>CanTSyn</td>
</tr>
<tr>
<td>SystemTemplate::GlobalTime::GlobalTimeMaster.syncPeriod</td>
</tr>
<tr>
<td>SystemTemplate::GlobalTime::CAN::GlobalTimeCanMaster.syncConfirmationTimeout</td>
</tr>
<tr>
<td>SystemTemplate::GlobalTime::GlobalTimeDomain.globalTimePdu</td>
</tr>
<tr>
<td>ECU Configuration Parameters</td>
</tr>
<tr>
<td>StbM</td>
</tr>
<tr>
<td>StbM.StbMGeneral.StbMMainFunctionPeriod</td>
</tr>
</tbody>
</table>
**StbM.StbMSynchronizedTimeBase.StbMLocalTimeRef**
Ref. to OSCounter

**CanTSyn**

**CanTSyn.CanTSynGeneral.CanTSynMainFunctionPeriod** 5ms

**CanTSyn.CanTSynGlobalTimeDomain.**
**CanTSynGlobalTimeDomainId** 1

**CanTSyn.CanTSynGlobalTimeDomain.**
**CanTSynSynchronizedTimeBaseRef** Ref. to StbM time base

**CanTSyn.CanTSynGlobalTimeDomain.**
**CanTSynGlobalTimeMaster.**
**CanTSynGlobalTimeTxFollowUpOffset** 100ms

**CanTSyn.CanTSynGlobalTimeDomain.**
**CanTSynGlobalTimeMaster.**
**CanTSynGlobalTimeMasterPdu.**
**CanTSynGlobalTimeMasterConfirmationHandleId** 0

**Use Cases**

**UC 01.01**

---

**SCG_ATS_GlobalTimeSync_Time Slave**

**System Description Parameters**

**StbM**

**SystemTemplate::GlobalTime::GlobalTimeMaster.isSystemWideGlobalTimeMaster** FALSE

**CanTSyn**

**SystemTemplate::GlobalTime::GlobalTimeMaster.syncPeriod** 2000ms

**SystemTemplate::GlobalTime::GlobalTimeDomain.followUpTimeoutValue** 300ms

**ECU Configuration Parameters**

**StbM**

**StbM.StbMGeneral.StbMMainFunctionPeriod** 5ms

**StbM.StbMSynchronizedTimeBase.StbMLocalTimeRef** Ref. to OSCounter

**CanTSyn**

**CanTSyn.CanTSynGeneral.CanTSynMainFunctionPeriod** 5ms

**CanTSyn.CanTSynGlobalTimeDomain.**
**CanTSynGlobalTimeDomainId** 1

**CanTSyn.CanTSynGlobalTimeDomain.**
**CanTSynGlobalTimeSequenceCounterJumpWidth** 1

**CanTSyn.CanTSynGlobalTimeDomain.**
**CanTSynSynchronizedTimeBaseRef** Ref. to StbM time base

**CanTSyn.CanTSynGlobalTimeDomain.**
**CanTSynGlobalTimeSlave.CanTSynGlobalTimeSlavePdu.**
**CanTSynGlobalTimeSlaveConfirmationHandleId** 1

**CanTSyn.CanTSynGlobalTimeDomain.**
**CanTSynGlobalTimeSlave.CanTSynGlobalTimeSlavePdu.**
Ref. to PDU
CanTSynGlobalTimePduRef

Use Cases
UC 01.02

SCG_ATS_GlobalTimeSync_Schedule table synchronization

<table>
<thead>
<tr>
<th>System Description Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM</td>
</tr>
<tr>
<td>SystemTemplate::GlobalTime::GlobalTimeMaster.isSystemWideGlobalTimeMaster</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ECU Configuration Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM</td>
</tr>
<tr>
<td>StbM.StbMGeneral.StbMMainFunctionPeriod</td>
</tr>
<tr>
<td>StbM.StbMTriggeredCustomer.StbMTriggeredCustomerPeriod</td>
</tr>
<tr>
<td>StbM.StbMSynchronizedTimeBase.StbMLocalTimeRef</td>
</tr>
<tr>
<td>StbM.StbMTriggeredCustomer.StbMOSSScheduleTableRef</td>
</tr>
<tr>
<td>StbM.StbMTriggeredCustomer.StbMSynchronizedTimeBaseRef</td>
</tr>
</tbody>
</table>

Test Cases
UC 01.03

SCG_ATS_GlobalTimeSync_NvM

<table>
<thead>
<tr>
<th>System Description Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM</td>
</tr>
<tr>
<td>SystemTemplate::GlobalTime::GlobalTimeMaster.isSystemWideGlobalTimeMaster</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ECU Configuration Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM</td>
</tr>
<tr>
<td>StbM.StbMGeneral.StbMMainFunctionPeriod</td>
</tr>
<tr>
<td>StbM.StbMSynchronizedTimeBase.StbMLocalTimeRef</td>
</tr>
<tr>
<td>StbM.StbMSynchronizedTimeBase.StbMStoreTimebaseNonVolatile</td>
</tr>
</tbody>
</table>

Use Cases
UC 01.04

4.2.2.2 Required System Description
Refer section 3.2.2.1

4.2.2.3 Required ECU Configuration
Refer section 3.2.2.1
4.2.2.4 Required Software Components

Fig: UC 01.01 - Global Time Master over CAN SWC Overview

Fig: UC 01.02 - Time Slave over CAN SWC Overview
### 4.2.2.4.1 SWC Client GlobalTime_Provider

<table>
<thead>
<tr>
<th>SWC Name</th>
<th>GlobalTime_Provider</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Client_SetGlobalTime</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>RPortPrototype</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>GlobalTime_Master_Interface</td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td></td>
</tr>
</tbody>
</table>

#### PORTS

<table>
<thead>
<tr>
<th>Name</th>
<th>Client_SetUserData</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>RPortPrototype</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>GlobalTime_Master_Interface</td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Client_SetOffset</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>RPortPrototype</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>GlobalTime_Master_Interface</td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td></td>
</tr>
</tbody>
</table>

#### RUNNABLE ENTITIES

<table>
<thead>
<tr>
<th>Name</th>
<th>RUN_GlobalTimeProvider</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements</strong></td>
<td>Runnable shall be invoked by TCP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>sscp_GlobalTimeProvider</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>SynchronousServerCallPoint</td>
</tr>
</tbody>
</table>

#### ServerCallPoint

- **Access to**
  - Client_SetGlobalTime (Write operation)
  - Client_SetUserData (Write operation)
  - Client_SetOffset (Write operation)

| **Requirements** |                     |

### 4.2.2.4.2 SWC Client Time_User

<table>
<thead>
<tr>
<th>SWC Name</th>
<th>Time_User</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Client_GetCurrentTime</td>
</tr>
</tbody>
</table>
### PORTS

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Interface</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client_GetCurrentTimeExtended</td>
<td>RPortPrototype</td>
<td>GlobalTime_Slave_Interface</td>
<td></td>
</tr>
</tbody>
</table>

### RUNNABLE ENTITIES

<table>
<thead>
<tr>
<th>Name</th>
<th>Requirements</th>
<th>Access to</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUN_TimeUser</td>
<td>Runnable shall be invoked by TCP</td>
<td>Client_GetCurrentTime (Read operation)</td>
</tr>
<tr>
<td>sscp_TimeUser</td>
<td></td>
<td>Client_GetCurrentTimeExtended (Read operation)</td>
</tr>
</tbody>
</table>

### 4.2.2.4.3 SWC Server StbM

<table>
<thead>
<tr>
<th>SWC Name</th>
<th>StbM</th>
<th>Name</th>
<th>Type</th>
<th>Interface</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Server_SetGlobalTime</td>
<td>PPortPrototype</td>
<td>GlobalTime_Master_Interface</td>
<td></td>
</tr>
</tbody>
</table>

|                                  |                                | Server_SetOffset            | PPortPrototype    | GlobalTime_Master_Interface    |              |

- Document ID 841: AUTOSAR_ATS_GlobalTimeSynchronization
- AUTOSAR Confidential
## PORTS

<table>
<thead>
<tr>
<th>Name</th>
<th>Server_SetUserData</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>PPortPrototype</td>
</tr>
<tr>
<td>Interface</td>
<td>GlobalTime_Master_Interface</td>
</tr>
<tr>
<td>Requirements</td>
<td></td>
</tr>
</tbody>
</table>

## Requirements

<table>
<thead>
<tr>
<th>Name</th>
<th>Server_GetCurrentTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>PPortPrototype</td>
</tr>
<tr>
<td>Interface</td>
<td>GlobalTime_Slave_Interface</td>
</tr>
<tr>
<td>Requirements</td>
<td></td>
</tr>
</tbody>
</table>

## Requirements

<table>
<thead>
<tr>
<th>Name</th>
<th>Server_GetCurrentTimeExtended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>PPortPrototype</td>
</tr>
<tr>
<td>Interface</td>
<td>GlobalTime_Slave_Interface</td>
</tr>
<tr>
<td>Requirements</td>
<td></td>
</tr>
</tbody>
</table>

## RUNNABLE ENTITIES

### StbM_SetGlobalTime

<table>
<thead>
<tr>
<th>Requirements</th>
<th></th>
</tr>
</thead>
</table>

### Started by Event

<table>
<thead>
<tr>
<th>Name</th>
<th>OIE_SetGlobalTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>OperationInvokedEvent</td>
</tr>
<tr>
<td>Port</td>
<td>Server_SetGlobalTime</td>
</tr>
<tr>
<td>Operation</td>
<td>Read</td>
</tr>
<tr>
<td>Requirements</td>
<td></td>
</tr>
</tbody>
</table>

### StbM_SetOffset

<table>
<thead>
<tr>
<th>Requirements</th>
<th></th>
</tr>
</thead>
</table>

### ServerCallPoint

<table>
<thead>
<tr>
<th>Name</th>
<th>OIE_SetOffset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>OperationInvokedEvent</td>
</tr>
<tr>
<td>Port</td>
<td>Server_SetOffset</td>
</tr>
<tr>
<td>Operation</td>
<td>Read</td>
</tr>
<tr>
<td>Requirements</td>
<td></td>
</tr>
</tbody>
</table>

### StbM_SetUserData

<table>
<thead>
<tr>
<th>Requirements</th>
<th></th>
</tr>
</thead>
</table>
### Requirements

#### Started by Event

<table>
<thead>
<tr>
<th>Name</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>OIE_SetUserData</td>
<td></td>
</tr>
</tbody>
</table>

- **Type**: OperationInvokedEvent
- **Port**: Server_SetUserData
- **Operation**: Read

#### Requirements

<table>
<thead>
<tr>
<th>Name</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM_GetCurrentTime</td>
<td></td>
</tr>
</tbody>
</table>

- **Type**: OperationInvokedEvent
- **Port**: Server_GetCurrentTime
- **Operation**: Read

#### Requirements

<table>
<thead>
<tr>
<th>Name</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>OIE_GetCurrentTime</td>
<td></td>
</tr>
</tbody>
</table>

- **Type**: OperationInvokedEvent
- **Port**: Server_GetCurrentTime
- **Operation**: Read

#### Requirements

<table>
<thead>
<tr>
<th>Name</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM_GetCurrentTimeExtended</td>
<td></td>
</tr>
</tbody>
</table>

- **Type**: OperationInvokedEvent
- **Port**: Server_GetCurrentTimeExtended
- **Operation**: Read

### 4.3 Re-usable Test Steps

Not applicable

### 4.4 Test cases

#### 4.4.1 [ATS_GTS_01228] Global Time Master: Setting of Global Time base by Active Customer and sending of SYNC frames (CRC_SUPPORTED) over CAN

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Time Master: Setting of Global Time base by Active Customer and sending of SYNC frames (CRC_SUPPORTED) over CAN</td>
<td>ATS_GTS_01228</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>Releases</td>
</tr>
<tr>
<td>------------------</td>
<td>----------</td>
</tr>
<tr>
<td>StbM, CanTSyn</td>
<td>State</td>
</tr>
<tr>
<td>reviewed</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trace to Requirement on Acceptance Test Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATR: ATR_ATR_00131</td>
</tr>
<tr>
<td>ATR: ATR_ATR_00132</td>
</tr>
<tr>
<td>ATR: ATR_ATR_00133</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trace to SWS Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>SynchronizedTimeBaseManager: SWS_StbM_00240</td>
</tr>
<tr>
<td>TimeSyncOverCAN: SWS_CanTSyn_00011</td>
</tr>
<tr>
<td>TimeSyncOverCAN: SWS_CanTSyn_00028</td>
</tr>
<tr>
<td>TimeSyncOverCAN: SWS_CanTSyn_00031</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Requirements Reference to Test Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case UC01.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Configuration Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM:</td>
</tr>
<tr>
<td>StbMSynchronizedTimeBaseIdentifier = 1</td>
</tr>
<tr>
<td>CanTSyn:</td>
</tr>
<tr>
<td>CanTSynGlobalTimeTxCrcSecured = CRC_SUPPORTED</td>
</tr>
<tr>
<td>CanTSynGlobalTimeSyncDataDLIndex = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aim is to:</td>
</tr>
<tr>
<td>Verify that StbM accepts the global time base from Upper Tester using client-server interface.</td>
</tr>
<tr>
<td>Verify that CanTSyn shall Transmit the global time base to time slave periodically via SYNC and FUP message with CRC secured.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Needed Adaptation to other Releases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pre-conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUT shall be initialized.</td>
</tr>
</tbody>
</table>

### Main Test Execution

#### Test Steps

<table>
<thead>
<tr>
<th>Test Steps</th>
<th>Pass Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td></td>
</tr>
<tr>
<td>[CP]</td>
<td></td>
</tr>
<tr>
<td>Start RUN_GlobalTimeProvider</td>
<td></td>
</tr>
</tbody>
</table>

| Step 2     |               |
| [RUN<RUN_GlobalTimeProvider>]         |               |
| Execute Rte_Call_Client_SetGlobalTime and Rte_Call_Client_SetUserData with below values: |               |
| timeBaseId = 1                          |               |
| StbM_TimeStampType.nanoseconds = 0x00000000 |               |
| StbM_TimeStampType.seconds =            |               |

<p>| [RUN&lt;RUN_GlobalTimeProvider&gt;] |               |
| Rte_Call returns RTE_E_OK |               |</p>
<table>
<thead>
<tr>
<th>Step 3</th>
<th>[CP]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Wait 100ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>[CP]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Start RUN_TimeUser</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Execute Rte_Call_Client_GetCurrentTime</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rte_Call returns RTE_E_OK.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time received from Time user should be as mentioned below:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>StbM_TimeStampType.nanoseconds = 0x00000000 + &lt;TestWaitTime = 100ms&gt;.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>StbM_TimeStampType.seconds = 0x00000E10(3600d)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>StbM_TimeStampType.secondsHi = 0x0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>StbM_UserDataType_User Data Byte 0 = 0xAA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>[LT&lt;CAN&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Receives the SYNC message with CRC validation as per test co-ordination requirement for CRC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>[LT]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Receives SYNC message in the format mentioned below:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>[LT]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Byte 0: Type = 0x20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>[LT]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Byte 1: CRC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>[LT]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Byte 2: D = 0x1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>[LT]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SC = 0x0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>[LT]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Byte 3: User Byte 0 = 0xAA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>[LT]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Byte 4-7: SyncTimeSec = StbM_TimeStampType.seconds</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>[LT&lt;CAN&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Receives the FUP message with CRC validation as per test co-ordination requirement for CRC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>[LT]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Receives FUP message in the format mentioned below:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>[LT]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>StbM_TimeStampType.seconds</td>
</tr>
<tr>
<td>Step 8</td>
<td>[LT&lt;CAN&gt;]</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td>Receives Global time base.</td>
</tr>
<tr>
<td></td>
<td>Store the time as base for next periodic message processing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 9</th>
<th>[LT&lt;CAN&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Receives the next periodic SYNC and FUP message with CRC validation as per test coordination requirement for CRC and calculate the CRC value</td>
</tr>
</tbody>
</table>

|        | [LT] |
|        | Get the time stamp values as below: |
|        | StbM\_TimeStampType.nanoseconds = tA.Nano |
|        | StbM\_TimeStampType.seconds = tA.Sec |
|        | StbM\_TimeStampType.secondsHi = tA.SecHi |

|        | [LT] |
|        | CRC in byte 1 of the frame shall match with Calculated CRC value. |
|        | Time stamp values shall be as mentioned below: |
|        | StbM\_TimeStampType.nanoseconds = tB.Nano |
|        | StbM\_TimeStampType.seconds = tB.Sec |
|        | StbM\_TimeStampType.secondsHi = tB.SecHi |
|        | StbM\_UserDataType.User Data Byte 0 = 0xAA |
|        | The difference between two time base shall be |
|        | tB – tA = CanTSynGlobalTimeTxPeriod (2 second) + CanTSynGlobalTimeTxFollowUpOffset |

Byte 0: Type = 0x28
Byte 1: CRC
Byte 2: D = 0x1
    SC = 0x0
Byte 3: reserved (Bit 7 to Bit 3) = 0
SGW (Bit 2) = 0
OVS = Overflow of seconds (Bit 1 to Bit 0)
Byte 4-7: SyncTimeNSec = StbM\_TimeStampType.nanoseconds
## 4.4.2 [ATS_GTS_01266] Global Time Master: Handling of SYNC message Confirmation Failures

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Global Time Master: Handling of SYNC message Confirmation Failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01266</td>
</tr>
<tr>
<td><strong>AUTOSAR</strong> Releases</td>
<td>4.2.1</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>CanTSyn</td>
</tr>
<tr>
<td><strong>State</strong></td>
<td>reviewed</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00133</td>
</tr>
<tr>
<td>Trace to SWS Item</td>
<td>TimeSyncOverCAN: SWS_CanTSyn_00033</td>
</tr>
<tr>
<td>Requirements Reference to Test Environment</td>
<td>Use Case UC01.01</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>StbM:</td>
</tr>
<tr>
<td></td>
<td>StbMSynchronizedTimeBaseIdentifier = 1</td>
</tr>
<tr>
<td></td>
<td>CanTSyn:</td>
</tr>
<tr>
<td></td>
<td>CanTSynGlobalTimeTxCrcSecured= CRC_NOT_SUPPORTED</td>
</tr>
<tr>
<td></td>
<td>CanIf:</td>
</tr>
<tr>
<td></td>
<td>CanIf.CanIfInitCfg.CanIfTxPduCfg.CanIfTxPduUserTxConfirmationUL = CDD.</td>
</tr>
<tr>
<td>Summary</td>
<td>Aim is to:</td>
</tr>
<tr>
<td></td>
<td>Verify that CanTSyn shall send the SYNC message and on confirmation timeout 'CanTSynMasterConfirmationTimeout', transmission request shall be revoked and no FUP message shall be sent.</td>
</tr>
<tr>
<td>Needed Adaptation to other Releases</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Pre-conditions</td>
<td>SUT shall be initialized.</td>
</tr>
<tr>
<td></td>
<td>StbM shall be initialized with base time:</td>
</tr>
<tr>
<td></td>
<td>StbM._TimeStampType.nanoseconds = 0x00000000</td>
</tr>
<tr>
<td></td>
<td>StbM._TimeStampType.seconds = 0x00000000</td>
</tr>
<tr>
<td></td>
<td>StbM._TimeStampType.secondsHi = 0x0000</td>
</tr>
<tr>
<td>Main Test Execution</td>
<td></td>
</tr>
<tr>
<td>Test Steps</td>
<td>[SUT]</td>
</tr>
<tr>
<td></td>
<td>CanTSyn transmit SYNC message</td>
</tr>
<tr>
<td></td>
<td>[LT&lt;CAN&gt;]</td>
</tr>
<tr>
<td></td>
<td>Receive SYNC message with format</td>
</tr>
<tr>
<td></td>
<td>Byte 0: Type = 0x10</td>
</tr>
<tr>
<td></td>
<td>Byte 1: User Byte 1 = 0x00</td>
</tr>
<tr>
<td>Step 2</td>
<td>[LT&lt;CAN&gt;]</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>Waits for FUP message</td>
</tr>
<tr>
<td></td>
<td>Since parameter CanIfTxPduUserTxConfirmationUL for CanTSyn PDU = CDD, confirmation shall not reach StbM. This causes transmit confirmation timeout at CanTSyn</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>[SUT]</th>
<th>[LT&lt;CAN&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CanTSyn transmit next periodic SYNC message</td>
<td>Receive SYNC message with format</td>
</tr>
<tr>
<td></td>
<td>Byte 0: Type = 0x10</td>
<td>Byte 0: Type = 0x10</td>
</tr>
<tr>
<td></td>
<td>Byte 1: User Byte 1 = 0x00</td>
<td>Byte 1: User Byte 1 = 0x00</td>
</tr>
<tr>
<td></td>
<td>Byte 2: D = 0x1</td>
<td>Byte 2: D = 0x1</td>
</tr>
<tr>
<td></td>
<td>SC = 0x0</td>
<td>SC = 0x0</td>
</tr>
<tr>
<td></td>
<td>Byte 3: User Byte 0 = 0x00</td>
<td>Byte 3: User Byte 0 = 0x00</td>
</tr>
<tr>
<td></td>
<td>Byte 4-7: SyncTimeSec = StbM_TimeStampType.seconds + CanTSynGlobalTimeTxPeriod (2 second) + CanTSynMasterConfirmationTimeout</td>
<td>Byte 4-7: SyncTimeSec = StbM_TimeStampType.seconds + CanTSynGlobalTimeTxPeriod (2 second) + CanTSynMasterConfirmationTimeout</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>[LT&lt;CAN&gt;]</th>
<th>[LT&lt;CAN&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Waits for FUP message</td>
<td>No FUP message received.</td>
</tr>
<tr>
<td></td>
<td>(Since parameter CanIfTxPduUserTxConfirmationUL for CanTSyn PDU = CDD, confirmation shall not reach StbM. This causes transmit confirmation timeout at CanTSyn)</td>
<td>(Since parameter CanIfTxPduUserTxConfirmationUL for CanTSyn PDU = CDD, confirmation shall not reach StbM. This causes transmit confirmation timeout at CanTSyn)</td>
</tr>
</tbody>
</table>

| Post-conditions | None |  |
4.4.3 [ATS_GTS_01264] Global Time Master: Setting of Global Time base by Active Customer and sending of offset frames (CRC_NOT_SUPPORTED) over CAN

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Global Time Master: Setting of Global Time base by Active Customer and sending of offset frames (CRC_NOT_SUPPORTED) over CAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01264</td>
</tr>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>StbM, CanTSyn</td>
</tr>
<tr>
<td>State</td>
<td>reviewed</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00131</td>
</tr>
<tr>
<td></td>
<td>ATR: ATR_ATR_00132</td>
</tr>
<tr>
<td></td>
<td>ATR: ATR_ATR_00133</td>
</tr>
<tr>
<td>Trace to SWS Item</td>
<td>SynchronizedTimeBaseManager: SWS_StbM_00240</td>
</tr>
<tr>
<td></td>
<td>TimeSyncOverCAN: SWS_CanTSyn_00038</td>
</tr>
<tr>
<td></td>
<td>TimeSyncOverCAN: SWS_CanTSyn_00040</td>
</tr>
<tr>
<td></td>
<td>TimeSyncOverCAN: SWS_CanTSyn_00041</td>
</tr>
<tr>
<td>Requirements / Reference to Test Environment</td>
<td>Use Case UC01.01</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>StbM: SbMBsynchronizedTimeBaseIdentifier = 16.</td>
</tr>
<tr>
<td></td>
<td>CanTSyn: CanTSynGlobalTimeTxWithCrcSecured= CRC_NOT_SUPPORTED</td>
</tr>
<tr>
<td>Summary</td>
<td>Aim is to:</td>
</tr>
<tr>
<td></td>
<td>Verify that StbM accepts the global time base from Upper Tester using client-server interface.</td>
</tr>
<tr>
<td></td>
<td>Verify that CanTSyn shall Transmit the offset time base to time slave periodically via OFS and OFNS message.</td>
</tr>
<tr>
<td>Needed Adaptation to other Releases</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Pre-conditions</td>
<td>SUT shall be initialized.</td>
</tr>
<tr>
<td>Main Test Execution</td>
<td></td>
</tr>
<tr>
<td>Test Steps</td>
<td>Pass Criteria</td>
</tr>
<tr>
<td>Step 1</td>
<td>[CP]</td>
</tr>
<tr>
<td></td>
<td>Start RUN_GlobalTimeProvider</td>
</tr>
<tr>
<td>Step 2</td>
<td>[RUN&lt;RUN_GlobalTimeProvider&gt;]</td>
</tr>
<tr>
<td></td>
<td>Execute Rte_Call_Client/SetGlobalTime with below values:</td>
</tr>
<tr>
<td></td>
<td>timeBaseId = 1</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.nanoseconds = 0x00000000</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.seconds =</td>
</tr>
<tr>
<td></td>
<td>[RUN&lt;RUN_GlobalTimeProvider&gt;]</td>
</tr>
<tr>
<td></td>
<td>Rte_Call returns RTE_E_OK</td>
</tr>
<tr>
<td>Step 3</td>
<td>[CP]</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Wait 100ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>[RUN&lt;RUN_GlobalTimeProvider&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Execute Rte_Call_Client_SetOffset with</td>
</tr>
<tr>
<td></td>
<td>below values:</td>
</tr>
<tr>
<td></td>
<td>timeBaseId = 16</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.nanoseconds = 0x00000000</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.seconds = 0x00000064</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.secondsHi = 0x0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>[CP]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Wait 100ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>[CP]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Start RUN_TimeUser</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Execute Rte_Call_Client_GetCurrentTime</td>
</tr>
<tr>
<td></td>
<td>Time base = 16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th>[LT&lt;CAN&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Receives the OFS message</td>
</tr>
</tbody>
</table>

|                       |                                           |
|                       | Receives OFS message with format         |
|                       | Byte 0: Type = 0x30                      |
|                       | Byte 1: reserved = 0x00                  |
|                       | Byte 2: D = 0x0                         |
|                       | SC = 0x0                                 |
### 4.4.4 [ATS_GTS_01268] Global Time Master: Handling of Offset message Confirmation Failures

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Global Time Master: Handling of Offset message Confirmation Failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01268</td>
</tr>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>CanTSyn</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00133</td>
</tr>
<tr>
<td>Trace to SWS Item</td>
<td>TimeSyncOverCAN: SWS_CanTSyn_00042</td>
</tr>
<tr>
<td>Requirements Reference to Test Environment</td>
<td>Use Case UC01.01</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>StbM: StbMSynchronizedTimeBaseIdentifier = 16</td>
</tr>
<tr>
<td></td>
<td>CanTSyn: CanTSynGlobalTimeTxCrcSecured= CRC_NOT_SUPPORTED</td>
</tr>
</tbody>
</table>
## CanIf:
CanIf.CanIfInitCfg.CanIfTxPduCfg.CanIfTxPduUserTxConfirmationUL = CDD.

### Summary
Aim is to:
Verify that CanTSyn shall send the OFS message and on confirmation timeout
'CanTSynMasterConfirmationTimeout', transmission request shall be revoked and
no OFNS message shall be sent.

### Needed Adaptation to other Releases
Not Applicable

### Pre-conditions
SUT shall be initialized.
StbM shall be initialized with base time:
StbM._TimeStampType.nanoseconds = 0x00000000
StbM._TimeStampType.seconds = 0x00000000
StbM._TimeStampType.secondsHi = 0x0000

### Main Test Execution

<table>
<thead>
<tr>
<th>Test Steps</th>
<th>Pass Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> [SUT]</td>
<td><strong>[LT&lt;CAN&gt;]</strong></td>
</tr>
<tr>
<td>CanTSyn transmit OFS message with TimeBaseId = 16 OfsTimeSecLsbLo = 0x00000002</td>
<td>Receive OFS message with format Byte 0: Type = 0x30 Byte 1: reserved = 0x00 Byte 2: D = 0x0 SC = 0x0 Byte 3: OfsTimeSecLsbHi = 8 Bit offset time stamp (LSB) from secondsHi Byte 4-7: OfsTimeSecLsbLo = StbM._TimeStampType.seconds</td>
</tr>
<tr>
<td><strong>Step 2</strong> [LT&lt;CAN&gt;]</td>
<td><strong>[LT&lt;CAN&gt;]</strong></td>
</tr>
<tr>
<td>Waits for OFNS message</td>
<td>No OFNS message received. (Since parameter CanIfTxPduUserTxConfirmationUL for CanTSyn PDU = CDD, Confirmation shall not reach StbM. This cause transmit confirmation timeout at CanTSyn)</td>
</tr>
<tr>
<td><strong>Step 3</strong> [SUT]</td>
<td><strong>[LT&lt;CAN&gt;]</strong></td>
</tr>
<tr>
<td>CanTSyn transmit next periodic OFS message</td>
<td>Receive OFS message with format Byte 0: Type = 0x30 Byte 1: reserved = 0x00 Byte 2: D = 0x0</td>
</tr>
</tbody>
</table>
4.4.5 [ATS_GTS_01238] Global Time Master: Handling of time base using NvM (Storage and Retrieve)

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Global Time Master: Handling of time base using NvM (Storage and Retrieve)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01238</td>
</tr>
<tr>
<td>AUTOSSAR</td>
<td>AUTOSAR Releases 4.2.1</td>
</tr>
<tr>
<td>Affected</td>
<td>StbM</td>
</tr>
<tr>
<td>Modules</td>
<td>State reviewed</td>
</tr>
<tr>
<td>Trace to</td>
<td>ATR: ATR_ATR_00131</td>
</tr>
<tr>
<td>Requirement on</td>
<td>ATR: ATR_ATR_00132</td>
</tr>
<tr>
<td>Acceptance</td>
<td></td>
</tr>
<tr>
<td>Test Document</td>
<td></td>
</tr>
<tr>
<td>Trace to SWS</td>
<td>SynchronizedTimeBaseManager: SWS_StbM_00171</td>
</tr>
<tr>
<td>Item</td>
<td>SynchronizedTimeBaseManager: SWS_StbM_00172</td>
</tr>
<tr>
<td>Requirements/</td>
<td>Use Case UC01.04</td>
</tr>
<tr>
<td>Reference to Test Environment</td>
<td></td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>StbM:</td>
</tr>
<tr>
<td></td>
<td>StbMMainFunctionPeriod 5ms</td>
</tr>
<tr>
<td></td>
<td>StbMlsSystemWideGlobalTimeMaster = TRUE</td>
</tr>
<tr>
<td></td>
<td>StbMStoreTimebaseNonVolatile = STORAGE_AT_SHUTDOWN</td>
</tr>
<tr>
<td>Summary</td>
<td>Aim is to Verify that during Initialization StbM shall load the Time base value from NvM and store the Time base to NvM at shutdown.</td>
</tr>
<tr>
<td>Needed Adaptation to other Releases</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Pre-conditions</td>
<td>SUT shall be initialized.</td>
</tr>
<tr>
<td>Main Test Execution</td>
<td></td>
</tr>
<tr>
<td>Test Steps</td>
<td>Pass Criteria</td>
</tr>
</tbody>
</table>

Step 4

[LT<CAN>] Waits for OFNS message
No OFNS message received.

(Since parameter CanIfTxPduUserTxConfirmationUL for CanTSyn PDU = CDD, Confirmation shall not reach StbM. This cause transmit confirmation timeout at CanTSyn)
<table>
<thead>
<tr>
<th>Step 1</th>
<th>[CP]</th>
<th>Start RUN_GlobalTimeProvider</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>[RUN&lt;RUN_GlobalTimeProvider&gt;]</td>
<td>Execute Rte_Call_Client_SetGlobalTime and Rte_Call_Client_SetUserData with below values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>timeBaseId = 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StbM_TimeStampType.nanoseconds = 0x00000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StbM_TimeStampType.seconds = 0x00000E10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StbM_TimeStampType.secondsHi = 0x0000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StbM_UserDataType.User Data Byte 0 = xAA</td>
</tr>
<tr>
<td></td>
<td>[RUN&lt;RUN_GlobalTimeProvider&gt;]</td>
<td>Rte_Call returns RTE_E_OK</td>
</tr>
<tr>
<td>Step 3</td>
<td>[CP]</td>
<td>Wait 100ms</td>
</tr>
<tr>
<td>Step 4</td>
<td>[CP]</td>
<td>Start RUN_TimeUser</td>
</tr>
<tr>
<td>Step 5</td>
<td>[RUN&lt;RUN_TimeUser&gt;]</td>
<td>Execute Rte_Call_Client_GetCurrentTime</td>
</tr>
<tr>
<td></td>
<td>[RUN&lt;RUN_TimeUser&gt;]</td>
<td>Rte_Call returns RTE_E_OK.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time received from Time user shall be as mentioned below:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StbM_TimeStampType.nanoseconds = 0x00000000 + &lt;TestWaitTime = 100ms&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StbM_TimeStampType.seconds = 0x00000E10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StbM_TimeStampType.secondsHi = 0x0000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StbM_UserDataType.User Data Byte 0 = 0xAA</td>
</tr>
<tr>
<td>Step 6</td>
<td>[SUT]</td>
<td>ECU shutdown</td>
</tr>
<tr>
<td></td>
<td>[SUT]</td>
<td>Time base shall be stored into NvM during shutdown</td>
</tr>
<tr>
<td>Step 7</td>
<td>[SUT]</td>
<td>Restart the ECU</td>
</tr>
<tr>
<td></td>
<td>[SUT]</td>
<td>Time base shall be retrieved from NvM during initialization.</td>
</tr>
</tbody>
</table>
### Step 8

[CP]

Wait 100ms

### Step 9

[CP]

Start RUN_TimeUser

### Step 10

[RUN<RUN_TimeUser>]

Execute Rte_Call_Client_GetCurrentTime

[RUN<RUN_TimeUser>]

Rte_Call returns RTE_E_OK.

Time read at Time user shall be:

\[
\text{StbM\_TimeStampType.nanoseconds} = 0x00000000 + \text{TestWaitTime = 100ms} + \text{TestWaitTime = 100ms} \\
\text{StbM\_TimeStampType.seconds} = 0x00000E10 \\
\text{StbM\_TimeStampType.secondsHi} = 0x0000 \\
\text{StbM\_UserDataType.User Data Byte 0} = 0xAA
\]

### Post-conditions

None

### 4.4.6 [ATS_GTS_01242] Time Slave: Reception of SYNC frames (CRC.IGNORED) over CAN, Synchronize Local Time Base and share the current time to active customers

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Time Slave: Reception of SYNC frames (CRC.IGNORED) over CAN, Synchronize Local Time Base and share the current time to active customers</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01242</td>
</tr>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>StbM, CanTSyn</td>
</tr>
<tr>
<td>State</td>
<td>reviewed</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00131, ATR: ATR_ATR_00132, ATR: ATR_ATR_00133</td>
</tr>
<tr>
<td>Trace to SWS Item</td>
<td>SynchronizedTimeBaseManager: SWS_StbM_00247, TimeSyncOverCAN: SWS_CanTSyn_00011</td>
</tr>
<tr>
<td>Requirements Reference to Test Environment</td>
<td>Use Case UC01.02</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>StbM: StbMSynchronizedTimeBaseIdentifier = 1, CanTSyn: CanTSynRxCrcValidated = CRC.IGNORED</td>
</tr>
</tbody>
</table>
## Summary

Verify that CanTSyn shall call StbM to update global time base on reception of SYNC and FUP message even with invalid CRC when CanTSynRxCrcValidated = CRC_IGNORED.

### Needed Adaptation to other Releases

Not Applicable

### Pre-conditions

SUT shall be initialized.

### Main Test Execution

<table>
<thead>
<tr>
<th>Test Steps</th>
<th>Pass Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>[SUT]</strong></td>
</tr>
<tr>
<td>[LT&lt;CAN&gt;] Transmit SYNC message with timeBaseId = 1</td>
<td>Receives SYNC message ignoring CRC value (if there) in below format</td>
</tr>
<tr>
<td>StbM_TimeStampType.seconds = 0x00000E10</td>
<td>Byte 0: Type = 0x10</td>
</tr>
<tr>
<td>StbM_TimeStampType.secondsHi = 0x0000</td>
<td>Byte 1: User Byte 1 = 0xBB</td>
</tr>
<tr>
<td>StbM_UserDataType.User Data Byte 0 = 0xAA</td>
<td>Byte 2: D = 0x1</td>
</tr>
<tr>
<td>StbM_UserDataType.User Data Byte 1 = 0xBB</td>
<td>Byte 3: User Byte 0 = 0xAA</td>
</tr>
<tr>
<td></td>
<td>Byte 4-7: SyncTimeSec = StbM_TimeStampType.seconds</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>[SUT]</strong></td>
</tr>
<tr>
<td>[LT&lt;CAN&gt;] Transmit FUP message with StbM_TimeStampType.nanoseconds = t4r</td>
<td>Receives FUP message ignoring CRC value (if there) in below format</td>
</tr>
<tr>
<td></td>
<td>Byte 0: Type = 0x18</td>
</tr>
<tr>
<td></td>
<td>Byte 1: User Byte 2 = 0x00</td>
</tr>
<tr>
<td></td>
<td>Byte 2: D = 0x1</td>
</tr>
<tr>
<td></td>
<td>SC = 0x0</td>
</tr>
<tr>
<td></td>
<td>Byte 3: reserved (Bit 7 to Bit 3) = 0</td>
</tr>
<tr>
<td></td>
<td>SGW (Bit 2) = 0</td>
</tr>
<tr>
<td></td>
<td>OVS = Overflow of seconds (Bit 1 to Bit 0)</td>
</tr>
<tr>
<td></td>
<td>Byte 4-7: SyncTimeNSec = StbM_TimeStampType.nanoseconds</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>[SUT]</strong></td>
</tr>
<tr>
<td>StbM updates its time base with values</td>
<td></td>
</tr>
</tbody>
</table>
Step 4 [CP]
Wait 800ms

Step 5 [CP]
Start RUN_TimeUser

Step 6 [RUN<RUN_TimeUser>]
Execute Rte_Call_Client_GetCurrentTime

Post-conditions None

4.4.7 [ATS_GTS_01271] Time Slave: Handling of SYNC message reception timeout (CanTSynGlobalTimeFollowUpTimeout)

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Time Slave: Handling of SYNC message reception timeout (CanTSynGlobalTimeFollowUpTimeout)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01271</td>
</tr>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>StbM, CanTSyn</td>
</tr>
<tr>
<td>State</td>
<td>reviewed</td>
</tr>
</tbody>
</table>
| Trace to Requirement on Acceptance Test Document | ATR: ATR_ATR_00131  
ATR: ATR_ATR_00132  
ATR: ATR_ATR_00133 |
| Trace to SWS Item | SynchronizedTimeBaseManager: SWS_StbM_00183  
SynchronizedTimeBaseManager: SWS_StbM_00247 |
**Requirements Reference to Test Environment**

<table>
<thead>
<tr>
<th>Requirement Parameter</th>
<th>Configuration Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM:</td>
<td>StbM: SynchronizedTimeBaseIdentifier = 1</td>
</tr>
<tr>
<td>CanTSyn:</td>
<td>CanTSynRxCrValidated = CRC_IGNORED</td>
</tr>
</tbody>
</table>

**Summary**

To verify that if FUP messages are not received within 'CanTSynGlobalTimeFollowUpTimeout', CanTSyn shall reset the sequence and wait for new SYNC message.

To verify StbM shall set the bit TIMEOUT in Sync state (when UT requests for current time) if StbM_BusSetGlobalTime is not invoked within StbMSyncLossTimeout and clear the bit on invocation of StbM_BusSetGlobalTime.

**Pre-conditions**

SUT shall be initialized.

**Main Test Execution**

<table>
<thead>
<tr>
<th>Test Steps</th>
<th>Pass Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>[LT&lt;CAN&gt;]</td>
<td>Transmit SYNC message with TimeBaseId = 1</td>
</tr>
<tr>
<td></td>
<td>Receiving SYNC message with format Byte 0: Type = 0x10</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.seconds = 0x00000E10</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.secondsHi = 0x0000</td>
</tr>
<tr>
<td></td>
<td>StbM_UserDataType.User Data Byte 0 = 0xAA</td>
</tr>
<tr>
<td></td>
<td>StbM_UserDataType.User Data Byte 1 = 0xBB</td>
</tr>
<tr>
<td>[SUT]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Receiving format Byte 0: Type = 0x10</td>
</tr>
<tr>
<td></td>
<td>Byte 1: User Byte 1 = 0xBB</td>
</tr>
<tr>
<td></td>
<td>Byte 2: D = 0x1 SC = 0x0</td>
</tr>
<tr>
<td></td>
<td>Byte 3: User Byte 0 = 0xAA</td>
</tr>
<tr>
<td></td>
<td>Byte 4-7: SyncTimeSec = StbM_TimeStampType.seconds</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>[LT&lt;CAN&gt;]</td>
<td>Transmit FUP message with StbM_TimeStampType.nanoseconds = t4r</td>
</tr>
<tr>
<td>[SUT]</td>
<td>Receiving FUP message with format Byte 0: Type = 0x18</td>
</tr>
<tr>
<td></td>
<td>Byte 1: Reserved</td>
</tr>
<tr>
<td></td>
<td>Byte 2: D = 0x1 SC = 0x0</td>
</tr>
<tr>
<td></td>
<td>Byte 3: reserved (Bit 7 to Bit 3) = 0</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>[SUT]</strong></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>[SUT]</td>
<td>CanTSyn shall update the time base of StbM</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.nanoseconds = tA.Nano</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.seconds = tA.Sec</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.secondsHi = tA.SecHi</td>
</tr>
<tr>
<td></td>
<td>Time updated in StbM, tA = tA + ToleranceTime_CanTSyn.</td>
</tr>
<tr>
<td></td>
<td>ToleranceTime_CanTSyn = Max one main function of CanTSyn as reception is asynchronous.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Step 4</strong></th>
<th><strong>[CP]</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>[CP]</td>
<td>Wait 800ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Step 5</strong></th>
<th><strong>[CP]</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>[CP]</td>
<td>Start RUN_TimeUser</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Step 6</strong></th>
<th><strong>[RUN&lt;RUN_TimeUser&gt;]</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>[RUN&lt;RUN_TimeUser&gt;]</td>
<td>Execute Rte_Call_Client_GetCurrentTime</td>
</tr>
<tr>
<td></td>
<td>Rte_Call returns RTE_E_OK.</td>
</tr>
<tr>
<td></td>
<td>Time read at Time user shall be:</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.nanoseconds = tA.Nano + &lt;TestWaitTime = 800ms&gt;</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.seconds = tA.Sec</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.secondsHi = tA.SecHi</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Step 7</strong></th>
<th><strong>[LT&lt;CAN&gt;]</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>[LT&lt;CAN&gt;]</td>
<td>Transmit SYNC message with</td>
</tr>
<tr>
<td></td>
<td>TimeBaseId = 1</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.seconds =</td>
</tr>
<tr>
<td></td>
<td>Receives SYNC message with format</td>
</tr>
<tr>
<td></td>
<td>Byte 0: Type = 0x10</td>
</tr>
<tr>
<td></td>
<td>Byte 1: User Byte 1 = 0xDD</td>
</tr>
</tbody>
</table>
4.4.8 [ATS_GTS_01273] Time Slave: Handling of SYNC message Sequence Mismatch failures

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Time Slave: Handling of SYNC message Sequence Mismatch failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01273</td>
</tr>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>StbM, CanTSyn</td>
</tr>
<tr>
<td>State</td>
<td>reviewed</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th>[LT&lt;CAN&gt;]</th>
<th>[SUT]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not transmit FUP message.</td>
<td>Do not receive FUP message.</td>
<td></td>
</tr>
<tr>
<td>Reception timeout occurs after 300ms (CanTSynGlobalTimeFollowUpTimeout = 300ms) the sequence is reset and waits for a new SYNC message.</td>
<td>StbM_TimeBaseStatusType.TIMEOUT is set to 1.</td>
<td></td>
</tr>
<tr>
<td>CanTSyn shall not update the time base of StbM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 9</th>
<th>[CP]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait 800ms</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 10</th>
<th>[CP]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start RUN_TimeUser</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 11</th>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute Rte_Call_Client_GetCurrentTime</td>
<td>Rte_Call returns RTE_E_OK.</td>
<td></td>
</tr>
<tr>
<td>Time read at Time user shall be:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StbM_TimeStampType.nanoseconds = tA.Nano + &lt;TestWaitTime = 800ms&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StbM_TimeStampType.seconds = tA .Sec + CanTSynGlobalTimeTxPeriod (2 second)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StbM_TimeStampType.secondsHi = tA .SecHi</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Post-conditions | None |

<table>
<thead>
<tr>
<th>0x00001C20</th>
<th>Byte 2: D = 0x1</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM_TimeStampType.secondsHi = 0x0000</td>
<td>SC = 0x1</td>
</tr>
<tr>
<td>StbM_UserDataType.User Data Byte 0 = 0xCC</td>
<td>Byte 3: User Byte 0 = 0xCC</td>
</tr>
<tr>
<td>StbM_UserDataType.User Data Byte 1 = 0xDD</td>
<td>Byte 4-7: SyncTimeSec = StbM_TimeStampType.seconds</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Byte 2: D = 0x1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC = 0x1</td>
</tr>
<tr>
<td>User Data Byte 0 = 0xCC</td>
</tr>
<tr>
<td>User Data Byte 1 = 0xDD</td>
</tr>
<tr>
<td>SyncTimeSec = StbM_TimeStampType.seconds</td>
</tr>
</tbody>
</table>
| Trace to Requirement on Acceptance Test Document | ATR: ATR_ATR_00131  
ATR: ATR_ATR_00132  
ATR: ATR_ATR_00133 |
|-----------------------------------------------|
| Trace to SWS Item  
Requirement s / Reference to Test Environment | SynchronizedTimeBaseManager: SWS_StbM_00247  
TimeSyncOverCAN: SWS_CanTSyn_00076  
TimeSyncOverCAN: SWS_CanTSyn_00078  
Use Case UC01.02 |
| Configuration Parameters | StbM:  
StbMSynchronizedTimeBaseIdentifier = 1  
CanTSyn:  
CanTSynRxCrLfValidated = CRC_IGNORED |
| Summary | To verify that if Sequence Counter of SYNC and FUP message are not matching, CanTSyn shall discard the received SYNC message and shall ignore the received FUP message.  
To verify Sequence Counter Jump Width between two SYNC messages greater than CanTSynGlobalTimeSequenceCounterJumpWidth, the messages will be ignored. |
| Needed Adaptation to other Releases | Not Applicable |
| Preconditions | SUT shall be initialized. |
| Main Test Execution | **Test Steps** | **Pass Criteria** |
| **Step 1** | [LT<CAN>] | [SUT] |
|  | Transmit SYNC message with TimeBaseline = 1  
StbM_TimeStampType.seconds = 0x00000E10  
StbM_TimeStampType.secondsHi = 0x0000  
StbM_UserDataType.User Data Byte 0 = 0xAA  
StbM_UserDataType.User Data Byte 1 = 0xBB | Receives SYNC message with format Byte 0: Type = 0x10  
Byte 1: User Byte 1 = 0xBB  
Byte 2: D = 0x1  
SC = 0x0  
Byte 3: User Byte 0 = 0xAA  
Byte 4-7: SyncTimeSec = StbM_TimeStampType.seconds |
| **Step 2** | [LT<CAN>] | [SUT] |
|  | Transmit FUP message with StbM_TimeStampType.nanosecon | Receives FUP message with format |
ds = t4r

Byte 0: Type = 0x18
Byte 1: Reserved
Byte 2: D = 0x1
    SC = 0x0
Byte 3: reserved (Bit 7 to Bit 3) = 0
SGW (Bit 2) = 0
OVS = Overflow of seconds (Bit 1 to Bit 0)

Byte 4-7: SyncTimeNSec =
StbM_TimeStampType.nanoseconds

Step 3

[SUT]
CanTSyn shall update the time base of StbM
StbM_TimeStampType.nanoseconds = tA.Nano
StbM_TimeStampType.seconds = tA.Sec
StbM_TimeStampType.secondsHi = tA.SecHi
Time updated in StbM, tA = tA +
ToleranceTime_CanTSyn.
ToleranceTime_CanTSyn = Max one main function
of CanTSyn as reception is asynchronous.

Step 4

[CP]
Wait 800ms

Step 5

[CP]
Start RUN_TimeUser

Step 6

[RUN<RUN_TimeUser>]
Execute Rte_Call_Client_GetCurrentTime

[RUN<RUN_TimeUser>]
Rte_Call returns RTE_E_OK.
Time read at Time user shall be:
StbM_TimeStampType.nanoseconds = tA.Nano +
<TestWaitTime = 800ms>.
StbM_TimeStampType.seconds = tA.Sec
StbM_TimeStampType.secondsHi = tA.SecHi
StbM_TimeBaseStatusType.TIMEOUT is set to 0.

Step 7

[LT<CAN>]
Transmit next periodic SYNC

[SUT]
<table>
<thead>
<tr>
<th>Step 8</th>
<th>[LT&lt;CAN&gt;]</th>
<th>[SUT]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmit FUP message with:</td>
<td>Receives FUP message with format</td>
<td></td>
</tr>
<tr>
<td>StbM_TimeStampType.nanoseconds = t9r</td>
<td>Byte 0: Type = 0x18</td>
<td></td>
</tr>
<tr>
<td>Different sequence number</td>
<td>Byte 1: Reserved</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Byte 2: D = 0x1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SC = 0x2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Byte 3: reserved (Bit 7 to Bit 3) = 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SGW (Bit 2) = 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OVS = Overflow of seconds (Bit 1 to Bit 0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Byte 4-7: SyncTimeNSec = StbM_TimeStampType.nanoseconds</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 9</th>
<th>[CP]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait 800ms</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 10</th>
<th>[CP]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start RUN_TimeUser</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 11</th>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute Rte_Call</td>
<td>Rte_Call returns RTE_E_OK.</td>
<td></td>
</tr>
<tr>
<td>Client_GetCurrentTime</td>
<td>Time read at Time user shall be:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.nanoseconds = tA.Nano + &lt;TestWaitTime = 800ms&gt;.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.seconds = tA .Sec +</td>
<td></td>
</tr>
</tbody>
</table>
### Step 12

**[LT<CAN>]**

Transmit next periodic SYNC message with timeBaseId = 1

StbM_TimeStampType.seconds = 0x00002328

StbM_TimeStampType.secondsHi = 0x0000

StbM_UserDataType.User Data Byte 0 = 0xEE

StbM_UserDataType.User Data Byte 1 = 0xFF

Sequence Counter = 3

**[SUT]**

Time base is not updated as per incoming time base as sequence counter value is greater than CanTSynGlobalTimeSequenceCounterJumpWidth.

### Step 13

**[CP]**

Wait 800ms

### Step 14

**[CP]**

Start RUN_TimeUser

### Step 15

**[RUN<RUN_TimeUser>]**

Execute Rte_Call_Client_GetCurrentTime

**[RUN<RUN_TimeUser>]**

Rte_Call returns RTE_E_OK.

Time read at Time user shall be:

StbM_TimeStampType.nanoseconds = tA.Nano + <TestWaitTime = 800ms> + <TestWaitTime = 800ms>.

StbM_TimeStampType.seconds = tA .Sec + CanTSynGlobalTimeTxPeriod (2 second) + CanTSynGlobalTimeTxPeriod (2 second).

StbM_TimeStampType.secondsHi = tA .SecHi

StbM_TimeBaseStatusType.TIMEOUT is set to 1.

**Post-conditions**

None
4.4.9 [ATS_GTS_01286] Time Slave: Reception of Offset frames (CRC_VALIDATED) over CAN, Synchronize Local Time Base and share the current time to active customers.

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Time Slave: Reception of Offset frames (CRC_VALIDATED) over CAN, Synchronize Local Time Base and share the current time to active customers.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01286</td>
</tr>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1/4.2.2</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>StbM, CanTSyn</td>
</tr>
<tr>
<td>State</td>
<td>reviewed</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00131</td>
</tr>
<tr>
<td>Trace to SWS Item</td>
<td>SynchronizedTimeBaseManager: SWS_StbM_00247</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00132</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00133</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>TimeSyncOverCAN: SWS_CanTSyn_00080</td>
</tr>
<tr>
<td>Requirements / Reference to Test Document</td>
<td>Use Case UC01.02</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>StbM:</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>StbM.SynchronizedTimeBaseIdentifier = 17.</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>CanTSyn:</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>CanTSynRxCrCValidated = CRC_VALIDATED</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>CanTSynGlobalTimeOfsDataIDListIndex = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>CanTSynGlobalTimeOfnsDataIDListIndex = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}</td>
</tr>
<tr>
<td>Summary</td>
<td>Aim is to:</td>
</tr>
<tr>
<td>Summary</td>
<td>Verify that CanTSyn shall Receive the offset time periodically via OFS and OFNS message respectively with CRC validation.</td>
</tr>
<tr>
<td>Summary</td>
<td>Verify that StbM shall synchronize its local offset time base on reception of Time Base from CanTSyn Verify that UT shall get the valid current time, offset time, current time in extended format, user data using Client-Server Interface.</td>
</tr>
<tr>
<td>Needed Adaptation to other Releases</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Pre-conditions</td>
<td>SUT shall be initialized.</td>
</tr>
<tr>
<td>Main Test Execution</td>
<td>Test Steps:</td>
</tr>
<tr>
<td>Test Steps</td>
<td>Test Steps</td>
</tr>
<tr>
<td>Test Steps</td>
<td>Step 1</td>
</tr>
<tr>
<td>Test Steps</td>
<td>Step 1</td>
</tr>
<tr>
<td>Test Steps</td>
<td>Step 1</td>
</tr>
<tr>
<td>Test Steps</td>
<td>Step 1</td>
</tr>
<tr>
<td>Test Steps</td>
<td>Pass Criteria</td>
</tr>
<tr>
<td>Test Steps</td>
<td>Pass Criteria</td>
</tr>
<tr>
<td>Test Steps</td>
<td>Pass Criteria</td>
</tr>
</tbody>
</table>
### Step 2

**[CP]**

Wait 800ms

### Step 3

**[LT<CAN>]**

Transmit OFS and OFNS message with:
- TimeBaseId = 17
- StbM_TimeStampType.nanoseconds = 0x00000000
- StbM_TimeStampType.seconds = 0x00000010 (16d)
- StbM_TimeStampType.secondsHi = 0x0000

**[SUT]**

Get the offset time stamp values with CRC validation as below:
- StbM_TimeStampType.nanoseconds = tA.Nano
- StbM_TimeStampType.seconds = tA.Sec
- StbM_TimeStampType.secondsHi = tA.SecHi

### Step 4

**[CP]**

Wait 800ms

### Step 5

**[CP]**

Start RUN_TimeUser

### Step 6

**[RUN<RUN_TimeUser>]**

Execute Rte_Call_Client_GetCurrentTime

**[RUN<RUN_TimeUser>]**

Rte_Call returns RTE_E_OK.

Time read at Time user shall be:
- StbM_TimeStampType.nanoseconds = tA.Nano + <TestWaitTime = 800ms> + <TestWaitTime = 800ms>
- StbM_TimeStampType.seconds = tA.Sec + <offset time = 0x00000010>
- StbM_TimeStampType.secondsHi = tA.SecHi

### Post-conditions

None
4.4.10 [ATS_GTS_01244] Time Slave: Reception of Offset frames (CRC_NOT_VALIDATED) over CAN, Synchronize Local Time Base and share the current time to active customer

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Time Slave: Reception of Offset frames (CRC_NOT_VALIDATED) over CAN, Synchronize Local Time Base and share the current time to active customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01244</td>
</tr>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>StbM, CanTSyn</td>
</tr>
<tr>
<td>State</td>
<td>reviewed</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00131, ATR: ATR_ATR_00132, ATR: ATR_ATR_00133</td>
</tr>
<tr>
<td>Trace to SWS Item</td>
<td>SynchronizedTimeBaseManager: SWS_StbM_00247, TimeSyncOverCAN: SWS_CanTSyn_00072</td>
</tr>
<tr>
<td>Requirements / Reference to Test Environment</td>
<td>Use Case UC01.02</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>StbM: StbMSynchronizedTimeBaseIdentifier = 16</td>
</tr>
<tr>
<td>CanTSyn: CanTSynRxCrcValidated = CRC_NOT_VALIDATED</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Aim is to: Verify that CanTSyn shall Receive the offset time periodically via OFS and OFNS message respectively. Verify that StbM shall synchronize its local offset time base on reception of Time Base from CanTSyn Verify that UT shall shall get the valid current time, offset time, current time in extended format, user data using Client-Server Interface.</td>
</tr>
<tr>
<td>Needed Adaptation to other Releases</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Pre-conditions</td>
<td>SUT shall be initialized.</td>
</tr>
</tbody>
</table>

### Main Test Execution

**Test Steps**

**Step 1**

- [LT<CAN>] Transmit SYNC and FUP message with:
  - TimeBaseId = 1
  - StbM_TimeStampType.nanoseconds = 0x00000000
  - StbM_TimeStampType.seconds = 0x00000E10 (3600d)
  - StbM_TimeStampType.secondsHi = 0x0000

- [SUT] Get the time stamp values as below:
  - StbM_TimeStampType.nanoseconds = tA.Nano
  - StbM_TimeStampType.seconds = tA.Sec
  - StbM_TimeStampType.secondsHi = tA.SecHi

- Time updated in StbM, tA = tA +
### 4.4.11 [ATS_GTS_01272] Time Slave: Handling of Offset message reception timeout (CanTSynGlobalTimeFollowUpTimeout)

<table>
<thead>
<tr>
<th>Step 2</th>
<th>[CP]</th>
<th>Wait 800ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 3</td>
<td>[LT&lt;CAN&gt;]</td>
<td>Transmit OFS and OFNS message with:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TimeBaseId = 16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StbM_TimeStampType.nanoseconds = 0x00000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StbM_TimeStampType.seconds = 0x0000000A (10d)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StbM_TimeStampType.secondsHi = 0x0000</td>
</tr>
<tr>
<td></td>
<td>[SUT]</td>
<td>Get the offset time stamp values as below:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StbM_TimeStampType.nanoseconds = tA.Nano</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StbM_TimeStampType.seconds = tA.Sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StbM_TimeStampType.secondsHi = tA.SecHi</td>
</tr>
<tr>
<td>Step 4</td>
<td>[CP]</td>
<td>Wait 800ms</td>
</tr>
<tr>
<td>Step 5</td>
<td>[CP]</td>
<td>Start RUN_TimeUser</td>
</tr>
<tr>
<td>Step 6</td>
<td>[RUN&lt;RUN_TimeUser&gt;]</td>
<td>Execute Rte_Call_Client_GetCurrentTime</td>
</tr>
<tr>
<td></td>
<td>[RUN&lt;RUN_TimeUser&gt;]</td>
<td>Rte_Call returns RTE_E_OK.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time read at Time user shall be:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StbM_TimeStampType.nanoseconds = tA.Nano + &lt;TestWaitTime = 800ms&gt; + &lt;TestWaitTime = 800ms&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StbM_TimeStampType.seconds = tA.Sec + &lt;offset time = 0x0000000A&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StbM_TimeStampType.secondsHi = tA.SecHi</td>
</tr>
</tbody>
</table>

**Post-conditions:** None
### Affected Modules

<table>
<thead>
<tr>
<th>Affected Modules</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM, CanTSyn</td>
<td>reviewed</td>
</tr>
</tbody>
</table>

### Trace to Requirement on Acceptance Test Document

- ATR: ATR_ATR_00131
- ATR: ATR_ATR_00132
- ATR: ATR_ATR_00133

### Trace to SWS Item

- SynchronizedTimeBaseManager: SWS_StbM_00247
- TimeSyncOverCAN: SWS_CanTSyn_00071

### Requirements / Reference to Test Environment

- Use Case UC01.02

### Configuration Parameters

- **StbM:**
  - StbMSynchronizedTimeBaseIdentifier = 16
- **CanTSyn:**
  - CanTSynRxCrValidated = CRC_IGNORRED

### Summary

To verify that if OFNS message are not received within 'CanTSynGlobalTimeFollowUpTimeout' CanTSyn shall reset the sequence and wait for new OFS message.

To verify StbM shall set the bit TIMEOUT in Sync state (when UT requests for current time) if StbM_BusSetGlobalTime is not invoked within StbMSyncLossTimeout and Clear the bit on invocation of StbM_BusSetGlobalTime.

### Needed Adaptation to other Releases

- Not Applicable

### Pre-conditions

- SUT shall be initialized.
- StbM shall be initialized with base time:
  - StbM_TimeStampType.nanoseconds = 0x00000000
  - StbM_TimeStampType.seconds = 0x00000000
  - StbM_TimeStampType.secondsHi = 0x0000

### Main Test Execution

#### Test Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Pass Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[LT&lt;CAN&gt;] Transmit OFS message with TimeBaseId = 16 StbM_TimeStampType.seconds = 0x00000014 StbM_TimeStampType.secondsHi = 0x0000</td>
<td>[SUT] Receives OFS message with format Byte 0: Type = 0x30 Byte 1: Reserved Byte 2: D = 0x0 Byte 3: OfsTimeSecLsbHi = 8 Bit offset time stamp (LSB) from secondsHi SC = 0x0 Byte 4-7: OfsTimeSecLsbLo = StbM_TimeStampType.seconds</td>
</tr>
<tr>
<td>2</td>
<td>[LT&lt;CAN&gt;]</td>
<td>[SUT]</td>
</tr>
<tr>
<td>Step 3</td>
<td>[SUT] CanTSyn shall update the time base of StbM</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM.TimeStampType.nanoseconds = tA.Nano</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM.TimeStampType.seconds = tA.Sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM.TimeStampType.secondsHi = tA.SecHi</td>
<td></td>
</tr>
</tbody>
</table>

| Step 4 | [CP] Wait 800ms |

| Step 5 | [CP] Start RUN_TimeUser |

<table>
<thead>
<tr>
<th>Step 6</th>
<th>[RUN&lt;RUN_TimeUser&gt;] Execute Rte_Call_Client_GetCurrentTime</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rte_Call returns RTE_E_OK.</td>
</tr>
<tr>
<td></td>
<td>Time read at Time user shall be:</td>
</tr>
<tr>
<td></td>
<td>StbM.TimeStampType.nanoseconds = tA.Nano + &lt;TestWaitTime = 800ms&gt;</td>
</tr>
<tr>
<td></td>
<td>StbM.TimeStampType.seconds = tA.Sec</td>
</tr>
<tr>
<td></td>
<td>StbM.TimeStampType.secondsHi = tA.SecHi</td>
</tr>
</tbody>
</table>

| Step 7 | [LT] Transmit OFS message with |

|        | [SUT] Receives OFS message with format |

Transmit OFNS message with

StbM.TimeStampType.nanoseconds = t4r

Receives OFNS message with format

Byte 0: Type = 0x38
Byte 1: Reserved
Byte 2: D = 0x0
SC = 0x0
Byte 3: OfstTimeSecMsbHi = 8 Bit offset time stamp (MSB) from secondsHi
Byte 4-7: OfstTimeNSec = StbM_TimeStampType.nanoseconds
4.4.12 [ATS_GTS_01274] Time Slave: Handling of Offset message Sequence Mismatch failure

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Time Slave: Handling of Offset message Sequence Mismatch failure</th>
</tr>
</thead>
</table>

**Post-conditions**

None
## Acceptance Test Specification of Global Time Synchronization

### AUTOSAR TC Release 1.2.0

<table>
<thead>
<tr>
<th>ID</th>
<th>ATS_GTS_01274</th>
<th>AUTOSAR Releases</th>
<th>4.2.1</th>
<th>4.2.2</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Affected Modules</th>
<th>StbM, CanTSyn</th>
<th>State</th>
<th>reviewed</th>
</tr>
</thead>
</table>

**Trace to Requirement on Acceptance Test Document**
- ATR: ATR_ATR_00131
- ATR: ATR_ATR_00132
- ATR: ATR_ATR_00133

**Trace to SWS Item**
- SynchronizedTimeBaseManager: SWS_StbM_00247
- TimeSyncOverCAN: SWS_CanTSyn_00077
- TimeSyncOverCAN: SWS_CanTSyn_00078

**Requirement / Reference to Test Environment**
- Use Case UC01.02

**Configuration Parameters**
- StbM: StbMSynchronizedTimeBaseIdentifier = 16
- CanTSyn: CanTSynRxCrValidated = CRC_IGNORED

**Summary**
- To verify that if Sequence Counter of OFS and OFNS message are not matching, CanTSyn shall discard the received OFS message and shall ignore the received OFNS message.
- To verify Sequence Counter Jump Width between two OFS messages greater than CanTSynGlobalTimeSequenceCounterJumpWidth, the messages will be ignored.

**Needed Adaptation to other Releases**
- Not Applicable

**Pre-conditions**
- SUT shall be initialized.
- StbM shall be initialized with base time:
  - StbM_TimeStampType.nanoseconds = 0x00000000
  - StbM_TimeStampType.seconds = 0x00000000
  - StbM_TimeStampType.secondsHi = 0x0000

**Main Test Execution**

**Test Steps**

<table>
<thead>
<tr>
<th>Step 1</th>
<th>[LT&lt;CAN&gt;]</th>
<th>[SUT]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmit OFS message with TimeBaseId = 16</td>
<td>Receives OFS message with format Byte 0: Type = 0x30</td>
<td></td>
</tr>
<tr>
<td>StbM_TimeStampType.seconds = 0x0000000014</td>
<td>Byte 1: Reserved</td>
<td></td>
</tr>
<tr>
<td>StbM_TimeStampType.secondsHi = 0x0000</td>
<td>Byte 2: D = 0x0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SC = 0x0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Byte 3: OfsTimeSecLsbHi = 8 Bit offset time stamp (LSB) from secondsHi</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><strong>[LT&lt;CAN&gt;]</strong> Transmit OFNS message with StbM_TimeStampType.nanoseconds = t4r</td>
<td><strong>[SUT]</strong> Receives OFNS message with format Byte 0: Type = 0x38 Byte 1: Reserved Byte 2: D = 0x0 SC = 0x0 Byte 3: OfsTimeSecMsbHi = 8 Bit offset time stamp (MSB) from secondsHi Byte 4-7: OfsTimeNSec = StbM_TimeStampType.nanoseconds</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Step 3</td>
<td><strong>[SUT]</strong> CanTSyn shall update the offset time base of StbM StbM_TimeStampType.nanoseconds = tA.Nano StbM_TimeStampType.seconds = tA .Sec StbM_TimeStampType.secondsHi = tA .SecHi</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><strong>[CP]</strong> Wait 800ms</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td><strong>[CP]</strong> Start RUN_TimeUser</td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td><strong>[RUN&lt;RUN_TimeUser&gt;]</strong> Execute Rte_Call_Client_GetCurrentTime <strong>[RUN&lt;RUN_TimeUser&gt;]</strong> Rte_Call returns RTE_E_OK. Time read at Time user shall be: StbM_TimeStampType.nanoseconds = tA.Nano + &lt;TestWaitTime = 800ms&gt;. StbM_TimeStampType.seconds = tA .Sec StbM_TimeBaseStatusType.TIMEOUT is set to 0. StbM_TimeStampType.secondsHi = tA .SecHi</td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td><strong>[LT&lt;CAN&gt;]</strong> Transmit next periodic OFS message</td>
<td><strong>[SUT]</strong> Receives OFS message with format</td>
</tr>
</tbody>
</table>
with

<table>
<thead>
<tr>
<th>TimeBaseId = 16</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM_TimeStampType.seconds = 0x00000032</td>
</tr>
<tr>
<td>StbM_TimeStampType.secondsHi = 0x0000</td>
</tr>
</tbody>
</table>

Byte 0: Type = 0x30

Byte 1: Reserved

Byte 2: D = 0x0

SC = 0x1

Byte 3: OfsTimeSecLsbHi = 8 Bit offset time stamp (LSB) from secondsHi

Step 8

[LT<CAN->]

Transmit OFNS message with different sequence number

[SUT]

Receives OFNS message with format

Byte 0: Type = 0x38

Byte 1: Reserved

Byte 2: D = 0x0

SC = 0x2

Byte 3: OfsTimeSecMsbHi = 8 Bit offset time stamp (MSB) from secondsHi

Byte 4-7: OfsTimeNSec = t4

CanTSyn shall not update the offset time base of StbM

Step 9

[CP]

Wait 800ms

Step 10

[CP]

Start RUN_TimeUser

Step 11

[RUN<RUN_TimeUser>]

Execute Rte_Call_Client_GetCurrentTime

[RUN<RUN_TimeUser>]

Rte_Call returns RTE_E_OK.

Time read at Time user shall be:

StbM_TimeStampType.nanoseconds = tA.Nano + <TestWaitTime = 800ms>.

StbM_TimeStampType.seconds = tA.Sec + CanTSynGlobalTimeTxPeriod (2 second)

StbM_TimeStampType.secondsHi = tA .SecHi

StbM_TimeBaseStatusType.TIMEOUT is set to 1.
Step 12  
[LT] 
Transmit next periodic OFS message with  
TimeBaseId = 16  
StbM_TimeStampType.seconds = 0x00000050  
StbM_TimeStampType.secondsHi = 0x0000  
Sequence Counter = 3  

[SUT] 
Time base is not updated as per new incoming time base as sequence counter value is greater than CanTSynGlobalTimeSequenceCounterJumpWidth.

Step 13  
[CP] 
Wait 800ms

Step 14  
[CP] 
Start RUN_TimeUser

Step 15  
[RUN<RUN_TimeUser>] 
Execute Rte_Call_Clients_GetCurrentTime  
[RUN<RUN_TimeUser>] 
Rte_Call returns RTE_E_OK.

Time read at Time user shall be:  
StbM_TimeStampType.nanoseconds = tA.Nano + <TestWaitTime = 800ms> + <TestWaitTime = 800ms>.  
StbM_TimeStampType.seconds = tA .Sec + CanTSynGlobalTimeTxPeriod (2 second) + CanTSynGlobalTimeTxPeriod (2 second  
StbM_TimeStampType.secondsHi = tA .SecHi  
StbM_TimeBaseStatusType.TIMEOUT is set to 1.

Post-conditions None

4.4.13 [ATS_GTS_01270] Time Slave: Synchronize Runnable entities to time base

Test Objective Time Slave: Synchronize Runnable entities to time base

<table>
<thead>
<tr>
<th>ID</th>
<th>ATS_GTS_01270</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1/4.2.2</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>StbM, CanTSyn</td>
</tr>
<tr>
<td>State</td>
<td>reviewed</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance</td>
<td>ATR: ATR_ATR_00132</td>
</tr>
<tr>
<td>Test Document</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td></td>
</tr>
</tbody>
</table>
| Trace to SWS Item | SynchronizedTimeBaseManager: SWS_StbM_00020  
SynchronizedTimeBaseManager: SWS_StbM_00247 |
| Requirements / Reference to Test Environment | Use Case UC01.03 |
| Configuration Parameters | StbM:  
StbMMainFunctionPeriod 5ms  
StbMSynchronizedTimeBaseIdentifier = 1  
StbMTriggeredCustomerPeriod 10ms  
StbMOSSScheduleTableRef = Reference to OS ScheduleTable  
StbMSynchronizedTimeBaseRef = Reference to StbMSynchronizedTimeBase  
CanTSyn:  
CanTSynRxCrValidated = CRC_IGNORED  
OS:  
ScheduleTableID = 1 |
| Summary | Aim is to verify that StbM shall synchronize the OS schedule tables (StbMOSSScheduleTableRef) with updated time base as received from time master. |
| Needed Adaptation to other Releases | Not Applicable |
| Pre-conditions | SUT shall be initialized. |

### Main Test Execution

<table>
<thead>
<tr>
<th>Test Steps</th>
<th>Pass Criteria</th>
</tr>
</thead>
</table>
| **Step 1** | **[SUT]**  
Transmit SYNC message with  
TimeBaseId = 1  
StbM_TimeStampType.seconds = 0x00000E10  
StbM_TimeStampType.secondsHi = 0x0000  
StbM_UserDataType.User Data Byte 0 = 0xAA  
StbM_UserDataType.User Data Byte 1 = 0xBB | Receives SYNC message with format  
Byte 0: Type = 0x10  
Byte 1: User Byte 1 = 0xBB  
Byte 2: D = 0x1  
SC = 0x0  
Byte 3: User Byte 0 = 0xAA  
Byte 4-7: SyncTimeSec = StbM_TimeStampType.seconds |
| **Step 2** | **[SUT]**  
Transmit FUP message with  
StbM_TimeStampType.nanoseconds = 0x00000000 | Receives FUP message with format  
Byte 0: Type = 0x18  
Byte 1: Reserved  
Byte 2: D = 0x1  
SC = 0x0 |
<table>
<thead>
<tr>
<th>Step 3</th>
<th>[SUT]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>StbM updates its time base in 500ms with values</td>
</tr>
<tr>
<td></td>
<td>StbM.TimeStampType.nanoseconds = tA.Nano + 500ms</td>
</tr>
<tr>
<td></td>
<td>StbM.TimeStampType.seconds = tA.Sec</td>
</tr>
<tr>
<td></td>
<td>StbM.TimeStampType.secondsHi = tA.SecHi</td>
</tr>
<tr>
<td></td>
<td>Time updated in StbM, tA = tA + ToleranceTime_CanTSyn.</td>
</tr>
<tr>
<td></td>
<td>ToleranceTime_CanTSyn = Max one main function of CanTSyn as reception is asynchronous.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>[SUT]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>StbM synchronizes OS schedule tables</td>
</tr>
<tr>
<td></td>
<td>StbM shall invoke</td>
</tr>
<tr>
<td></td>
<td>SyncScheduleTable(</td>
</tr>
<tr>
<td></td>
<td>&lt;ScheduleTableID = 1&gt;,</td>
</tr>
<tr>
<td></td>
<td>&lt;Tick_value&gt;)</td>
</tr>
</tbody>
</table>

Post-conditions: None
5 RS_BRF_01660 - Global Time Synchronization over FlexRay

5.1 General Test Objective and Approach

This Test Specification intends to cover the Global Time Synchronization feature of StbM and FrTSyn as described in the AUTOSAR Feature [RS_BRF_01660].

The tests use a test bench environment and Embedded Software Components that use the feature.

This test case document has been established to cover the following features:

Below Features are not tested in this test suite:

- SGW(Sync to Global Time) Handling [SWS_FrTSyn_00020]: Feature is tested in the test suite 'Global Time Synchronization over Multiple Bus'.
- Customer/User Access of Time Information, Time Base Store/Restore from NvM, Current Time Derivation using Local Counter: These features are tested in test suite 'Global Time Synchronization over CAN'.

This specification gives the description of required tests environments (test bench, uses case, configuration files) and detailed tests cases for executing tests.

5.1.1 Test System

5.1.1.1 Overview on Architecture

In order to cover the required features / sub-features coverage, the environment has been separated in several use cases.

Test Cases are derived based on below use cases
5.1.1.1.1 UC 02.01: Global Time Master over FlexRay

SUT acts as Global time Master and Sets time base, offset time base, User Data and Trigger for transmission of Synchronization over FlexRay bus.

As shown in below figure, Functionalities of Global Time master of a time domain are tested over FlexRay in the use case 02.01
5.1.1.1.2 UC 02.02: Time Slave over FlexRay

SUT acts as Time Slave and Gets time base, offset time base and Synchronizes Local time to Global Time base.

As shown in the figure, functionalities of Time Slave of a time domain are tested over FlexRay in the use case 02.02.

5.1.1.2 Specific Requirements

Not Applicable.

5.1.1.3 Test Coordination Requirements

UC 01.01: Global Time Master over FlexRay
• Test System (LT <FlexRay>) shall read the FrTSyn FlexRay Frames and decode the same as per Frame Format provided in AUTOSAR_SWS_TimeSyncOverFlexRay.

UC 01.02: Time Slave over FlexRay

• Test System (LT <FlexRay>) shall encode the FrTSyn FlexRay Frames as per Frame Format provided in AUTOSAR_SWS_TimeSyncOverFlexRay and transmit over bus.

Requirements for CRC Calculation

• Test System (LT < FlexRay >) shall use the Crc_CalculateCRC8H2F() (Refer AUTOSAR Specification of CRC Routines AUTOSAR_SWS_CRCLibrary.pdf) to calculate the CRC of the Frame. Below are the parameters used for CRC calculation:
  - The CRC start value shall be 0xFF.
  - The CRC final XOR-value shall be 0xFF.
  - The CRC polynomial shall be 0x2F.
  - The DataIDList shall be same as provided in FrTSyn Configuration.

5.1.2 Test Case Design

Below diagrams explain test design for different use cases

Fig: UC 02.01 - Global Time Master over FlexRay Test Design
5.2 Configuration requirements

The configuration can be divided into two separate parts. The test configuration describes variables used to parameterize the test case. The static configuration describes the necessary settings of the DUT in order to allow a test case to perform.

5.2.1 Test Configuration

Communication data base for FrTSyn is depicted below

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FrIf Tx Pdu</td>
<td>AT_201_Ipdu201</td>
</tr>
<tr>
<td>FrIfTxPdu. FrIfImmediate</td>
<td>FALSE</td>
</tr>
<tr>
<td>FrIfFrameTriggering.FrIfBaseCycle</td>
<td>0</td>
</tr>
<tr>
<td>FrIfFrameTriggering.FrIfCycleRepetition</td>
<td>64</td>
</tr>
<tr>
<td>FrIfFrameTriggering.FrIfSlotId</td>
<td>2</td>
</tr>
<tr>
<td>FrIf Rx Pdu</td>
<td>AT_202_Ipdu202</td>
</tr>
</tbody>
</table>

5.2.2 Static Configuration

5.2.2.1 Static Configuration Groups

<table>
<thead>
<tr>
<th>SCG_ATS_GlobalTimeSync_Time Master</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Description Parameters</td>
</tr>
<tr>
<td>StbM</td>
</tr>
<tr>
<td>SystemTemplate::GlobalTime::GlobalTimeMaster.isSystemWideGlobalTimeMaster</td>
</tr>
</tbody>
</table>
## Acceptance Test Specification of Global Time Synchronization

### AUTOSAR TC Release 1.2.0

**Document ID:** AUTOSAR_ATS_GlobalTimeSynchronization

- **FrTSyn**
  - SystemTemplate::GlobalTime::GlobalTimeDomain.domainId: 1
  - SystemTemplate::GlobalTime::GlobalTimeMaster.syncPeriod: 320ms
  - SystemTemplate::GlobalTime::GlobalTimeDomain.subDomain: Ref. To PDU

### ECU Configuration Parameters

- **StbM**
  - StbM.StbMGeneral.StbMMainFunctionPeriod: 5ms
  - StbM.StbMSynchronizedTimeBase.StbMLocalTimeRef: Ref. to OSCounter

### FrTSyn

- FrTSyn.FrTSynGeneral.FrTSynMainFunctionPeriod: 5ms
  - FrTSyn.FrTSynGlobalTimeDomain.FrTSynSynchronizedTimeBaseRef: Ref. to StbM time base
  - FrTSyn.CanTSynGlobalTimeDomain.FrTSynGlobalTimeMaster.FrTSynGlobalTimeMasterPdu.FrTSynGlobalTimeMasterConfirmationHandleId: 0

### Test Cases

**UC 02.01**

---

## SCG_ATS_GlobalTimeSync_Time Slave

**System Description Parameters**

- **StbM**
  - SystemTemplate::GlobalTime::GlobalTimeMaster.isSystemWideGlobalTimeMaster: FALSE

### FrTSyn

- SystemTemplate::GlobalTime::GlobalTimeDomain.subDomain: Ref. To PDU

### ECU Configuration Parameters

- **StbM**
  - StbM.StbMGeneral.StbMMainFunctionPeriod: 5ms
  - StbM.StbMSynchronizedTimeBase.StbMLocalTimeRef: Ref. to OSCounter

### FrTSyn

- FrTSyn.FrTSynGeneral.FrTSynMainFunctionPeriod: 5ms
  - FrTSyn.FrTSynGlobalTimeDomain.FrTSynGlobalTimeDomainId: 1
  - FrTSyn.FrTSynGlobalTimeDomain.FrTSynGlobalTimeSequenceCounterJumpWidth: 1
  - FrTSyn.FrTSynGlobalTimeDomain.FrTSynSynchronizedTimeBaseRef: Ref. to StbM time base

### Test Cases
5.2.2.2 Required System Description
Refer Section 4.2.2.1

5.2.2.3 Required ECU Configuration
Refer Section 4.2.2.1

5.2.2.4 Required Software Components

Fig: UC 02.01 - Global Time Master over FlexRay SWC Overview
5.2.2.4.1 SWC Client GlobalTime_Provider

<table>
<thead>
<tr>
<th>SWC Name</th>
<th>GlobalTime_Provider</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Name: Client_SetGlobalTime</td>
</tr>
<tr>
<td></td>
<td>Type: RPortPrototype</td>
</tr>
<tr>
<td></td>
<td>Interface: GlobalTime_Master_Interface</td>
</tr>
<tr>
<td></td>
<td>Requirements</td>
</tr>
<tr>
<td>PORTS</td>
<td>Name: Client_SetUserData</td>
</tr>
<tr>
<td></td>
<td>Type: RPortPrototype</td>
</tr>
<tr>
<td></td>
<td>Interface: GlobalTime_Master_Interface</td>
</tr>
<tr>
<td></td>
<td>Requirements</td>
</tr>
<tr>
<td></td>
<td>Name: Client_SetOffset</td>
</tr>
<tr>
<td></td>
<td>Type: RPortPrototype</td>
</tr>
<tr>
<td></td>
<td>Interface: GlobalTime_Master_Interface</td>
</tr>
<tr>
<td></td>
<td>Requirements</td>
</tr>
</tbody>
</table>
### RUNNABLE ENTITIES

<table>
<thead>
<tr>
<th>Name</th>
<th>RUN_GlobalTimeProvider</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>Runnable shall be invoked by TCP</td>
</tr>
</tbody>
</table>

#### ServerCallPoint

<table>
<thead>
<tr>
<th>Name</th>
<th>sscp_GlobalTimeProvider</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>SynchronousServerCallPoint</td>
</tr>
<tr>
<td>Access to</td>
<td>Client_SetGlobalTime (Write operation)</td>
</tr>
<tr>
<td></td>
<td>Client_SetUserData (Write operation)</td>
</tr>
<tr>
<td></td>
<td>Client_SetOffset (Write operation)</td>
</tr>
</tbody>
</table>

#### Requirements

**5.2.2.4.2 SWC Client Time_User**

<table>
<thead>
<tr>
<th>SWC Name</th>
<th>Time_User</th>
</tr>
</thead>
</table>

#### PORTS

<table>
<thead>
<tr>
<th>Name</th>
<th>Client_GetCurrentTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>RPortPrototype</td>
</tr>
<tr>
<td>Interface</td>
<td>GlobalTime_Slave_Interface</td>
</tr>
</tbody>
</table>

#### Requirements

<table>
<thead>
<tr>
<th>Name</th>
<th>Client_GetCurrentTimeExtended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>RPortPrototype</td>
</tr>
<tr>
<td>Interface</td>
<td>GlobalTime_Slave_Interface</td>
</tr>
</tbody>
</table>

#### Requirements

**RUNNABLE ENTITIES**

<table>
<thead>
<tr>
<th>Name</th>
<th>RUN_TimeUser</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>Runnable shall be invoked by TCP</td>
</tr>
</tbody>
</table>

#### ServerCallPoint

<table>
<thead>
<tr>
<th>Name</th>
<th>sscp_TimeUser</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>SynchronousServerCallPoint</td>
</tr>
<tr>
<td>Access to</td>
<td>Client_GetCurrentTime (Read operation)</td>
</tr>
<tr>
<td></td>
<td>Client_GetCurrentTimeExtended (Read operation)</td>
</tr>
</tbody>
</table>
### 5.2.2.4.3 SWC Server StbM

<table>
<thead>
<tr>
<th>SWC Name</th>
<th>StbM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Server_SetGlobalTime</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>PPortPrototype</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>GlobalTime_Master_Interface</td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td></td>
</tr>
</tbody>
</table>

| **Name** | Server_SetOffset |
| **Type** | PPortPrototype |
| **Interface** | GlobalTime_Master_Interface |
| **Requirements** | |

| **Name** | Server_SetUserData |
| **Type** | PPortPrototype |
| **Interface** | GlobalTime_Master_Interface |
| **Requirements** | |

| **Name** | Server_GetCurrentTime |
| **Type** | PPortPrototype |
| **Interface** | GlobalTime_Slave_Interface |
| **Requirements** | |

| **Name** | Server_GetCurrentTimeExtended |
| **Type** | PPortPrototype |
| **Interface** | GlobalTime_Slave_Interface |
| **Requirements** | |

| **Name** | StbM_SetGlobalTime |
| **Requirements** | |

| **Name** | OIE_SetGlobalTime |
| **Requirements** | |
### RUNNABLE ENTITIES

<table>
<thead>
<tr>
<th>Started by Event</th>
<th>Type</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OperationInvokedEvent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port: Server_SetGlobalTime</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Operation: Read</td>
<td></td>
</tr>
</tbody>
</table>

| Name             | StbM_SetOffset                |              |

<table>
<thead>
<tr>
<th>Started by Event</th>
<th>Type</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OperationInvokedEvent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port: Server_SetOffset</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Operation: Read</td>
<td></td>
</tr>
</tbody>
</table>

| Name             | OIE_SetOffset                 |              |

<table>
<thead>
<tr>
<th>Started by Event</th>
<th>Type</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OperationInvokedEvent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port: Server_SetUserData</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Operation: Read</td>
<td></td>
</tr>
</tbody>
</table>

| Name             | StbM_SetUserData              |              |

<table>
<thead>
<tr>
<th>Started by Event</th>
<th>Type</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OperationInvokedEvent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port: Server_SetUserData</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Operation: Read</td>
<td></td>
</tr>
</tbody>
</table>

| Name             | OIE_SetUserData               |              |

<table>
<thead>
<tr>
<th>Started by Event</th>
<th>Type</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OperationInvokedEvent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port: Server_GetCurrentTime</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Operation: Read</td>
<td></td>
</tr>
</tbody>
</table>

| Name             | StbM_GetCurrentTime           |              |

<table>
<thead>
<tr>
<th>Started by Event</th>
<th>Type</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OperationInvokedEvent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port: Server_GetCurrentTimeExtended</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Operation: Read</td>
<td></td>
</tr>
</tbody>
</table>

| Name             | OIE_GetCurrentTimeExtended    |              |

<table>
<thead>
<tr>
<th>Started by Event</th>
<th>Type</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OperationInvokedEvent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port: Server_GetCurrentTimeExtended</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Operation: Read</td>
<td></td>
</tr>
</tbody>
</table>

| Name             | OIE_GetCurrentTimeExtended    |              |
### 5.3 Re-usable Test Steps

Not Applicable

### 5.4 Test Cases

#### 5.4.1 [ATS_GTS_01275] Global Time Master: Setting of Global Time base and user data by Active Customer and sending of SYNC frames (CRC_NOT_SUPPORTED) over FlexRay

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Global Time Master: Setting of Global Time base and user data by Active Customer and sending of SYNC frames (CRC_NOT_SUPPORTED) over FlexRay</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01275</td>
</tr>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>StbM, FrTSyn</td>
</tr>
<tr>
<td>State</td>
<td>reviewed</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00134</td>
</tr>
<tr>
<td>Trace to SWS Item</td>
<td>TimeSyncOverFlexRay: SWS_FrTSyn_00023</td>
</tr>
<tr>
<td>Requirements / Reference to Test Environment</td>
<td>Use Case UC02.01</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>StbM: StbMSynchronizedTimeBaseIdentifier = 1</td>
</tr>
<tr>
<td>FrTSyn: FrTSynGlobalTimeTxCrcSecured = CRC_NOT_SUPPORTED</td>
<td></td>
</tr>
<tr>
<td>CanTSynGlobalTimeSyncDataIDListIndex = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}</td>
<td></td>
</tr>
<tr>
<td>CanTSynGlobalTimeFupDataIDListIndex = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Aim is to: Verify that StbM accepts the global time base and user data from SWC Global Time Provider using client-server Interface. Verify that FrTSyn shall transmit the global time base to time slave periodically via SYNC message</td>
</tr>
<tr>
<td>Needed Adaptation to</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

---

**Note:** The content above is a excerpt from a document titled "Acceptance Test Specification of Global Time Synchronization" and is subject to AUTOSAR Confidentiality. The document details test cases and requirements for ensuring global time synchronization in AUTOSAR Release 1.2.0.
### Pre-conditions
SUT shall be initialized.

### Main Test Execution

<table>
<thead>
<tr>
<th>Test Steps</th>
<th>Pass Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>[CP] Start RUN_GlobalTimeProvider.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>[RUN&lt;RUN_GlobalTimeProvider&gt;] Execute Rte_Call_Client_SetGlobalTime and Rte_Call_Client_SetUserData with below values:&lt;br&gt;- timeBaseId = 1&lt;br&gt;- StbM_TimeStampType.nanoseconds = 0x00000000&lt;br&gt;- StbM_TimeStampType.seconds = 0x00000E10&lt;br&gt;- StbM_TimeStampType.secondsHi = 0x0000&lt;br&gt;- StbM_UserDataType.User Data Byte 0 = 0xAA&lt;br&gt;- StbM_UserDataType.User Data Byte 1 = 0xBB&lt;br&gt;- StbM_UserDataType.User Data Byte 2 = 0xCC&lt;br&gt;<strong>[RUN&lt;RUN_GlobalTimeProvider&gt;]</strong> Rte_Call returns RTE_E_OK.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>[CP] Wait 100ms.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>[CP] Start RUN_TimeUser.</td>
</tr>
</tbody>
</table>
| **Step 5** | [RUN<RUN_TimeUser>] Execute Rte_Call_Client_GetCurrentTime. **[RUN<RUN_TimeUser>]** Rte_Call returns RTE_E_OK. Time received from Time user shall be as mentioned below:<br>- StbM_TimeStampType.nanoseconds = 0x00000000 + <TestWaitTime = 100ms><br>- StbM_TimeStampType.seconds = 0x00000E10<br>- StbM_TimeStampType.secondsHi = 0x0000
### Step 6

**[LT<FLEXRAY>]**

Receives the SYNC message without CRC validation.

- Byte 0: Type = 0x10
- Byte 1: UserByte 2
- Byte 2: D = 0x1
  - SC = 0x0
- Byte 3: FCNT = FlexRay Cycle Counter 0 (Bit 7 to Bit 2)
- SGW (Bit 1)
- SyncToSubDomain = 1
  - reserved (Bit 0), default: 0
- Byte 4: User Byte 0
- Byte 5: User Byte 1
- Byte 6-11: SyncTimeSec = 48 Bit time stamp in seconds
- Byte 12-15: SyncTimeNSec = 32 Bit time stamp in nanoseconds

### Step 7

**[LT<FLEXRAY>]**

Receives Global time base.

Store the time as base for next periodic message processing

- StbM_TimeStampType.nanoseconds = tA.Nano
- StbM_TimeStampType.seconds = tA.Sec
- StbM_TimeStampType.secondsHi = tA.SecHi

### Step 8

**[LT<FLEXRAY>]**

Receives the next periodic SYNC message without CRC validation

Time stamp values shall be as

- StbM_USERDataType.User Data Byte 0 = 0xAA
- StbM_USERDataType.User Data Byte 1 = 0xBB
- StbM_USERDataType.User Data Byte 2 = 0xCC
### 5.4.2 [ATS_GTS_01287] Global Time Master: Setting of Offset Time base by Active Customer and sending Offset frames (CRC_SUPPORTED) over FlexRay

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Global Time Master: Setting of Offset Time base by Active Customer and sending Offset frames (CRC_SUPPORTED) over FlexRay</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01287</td>
</tr>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>StbM, FrTSyn</td>
</tr>
<tr>
<td>State</td>
<td>reviewed</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00134</td>
</tr>
<tr>
<td>Trace to SWS Item</td>
<td>TimeSyncOverFlexRay: SWS_FrTSyn_00031</td>
</tr>
<tr>
<td>Requirements / Reference to Test Environment</td>
<td>Use Case UC02.01</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>StbM: StbMSynchronizedTimeBaseIdentifier = 17</td>
</tr>
<tr>
<td></td>
<td>FrTSyn: FrTSynGlobalTimeTxCrcSecured = CRC_SUPPORTED</td>
</tr>
</tbody>
</table>
## Summary

Aim is to:

- Verify that StbM accepts the global time base from Upper Tester using client-server interface.
- Verify that FrTSyn shall transmit the offset time base to time slave periodically via OFS message with CRC secured.

## Needed Adaptation to other Releases

Not Applicable

## Pre-conditions

SUT shall be initialized.

## Main Test Execution

### Test Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Pass Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>[CP] Start RUN_GlobalTimeProvider</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
<td>[RUN&lt;RUN_GlobalTimeProvider&gt;] Rte_Call returns RTE_E_OK</td>
</tr>
<tr>
<td></td>
<td>[RUN&lt;RUN_GlobalTimeProvider&gt;] Execute Rte_Call_Client_SetGlobalTime with below values:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TimeBaseId = 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.nanoseconds = 0x00000000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.seconds = 0x00000064 (3600d)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.secondsHi = 0x0000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>[CP] Wait 100ms</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
<td>[RUN&lt;RUN_GlobalTimeProvider&gt;] Rte_Call returns RTE_E_OK</td>
</tr>
<tr>
<td></td>
<td>[RUN&lt;RUN_GlobalTimeProvider&gt;] Execute Rte_Call_Client_SetOffset with below values:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>timeBaseId = 17</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.nanoseconds = 0x00000000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.seconds = 0x00000064</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.secondsHi = 0x0000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>[CP] Wait 100ms</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>[CP] Start RUN_TimeUser</td>
<td></td>
</tr>
</tbody>
</table>
### Step 7

<table>
<thead>
<tr>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
<th>Execute Rte_Call_Client_GetCurrentTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rte_Call returns RTE_E_OK.</td>
<td></td>
</tr>
<tr>
<td>Time base = 17</td>
<td></td>
</tr>
<tr>
<td>Time received from Time user shall be as mentioned below:</td>
<td></td>
</tr>
<tr>
<td>StbM_TimeStampType.nanoseconds = 0x00000000 + &lt;TestWaitTime = 100ms&gt; + &lt;TestWaitTime = 100ms&gt;</td>
<td></td>
</tr>
<tr>
<td>StbM_TimeStampType.seconds = 0x000000E10 + 0x00000064</td>
<td></td>
</tr>
<tr>
<td>StbM_TimeStampType.secondsHi = 0x0000</td>
<td></td>
</tr>
</tbody>
</table>

### Step 8

<table>
<thead>
<tr>
<th>[LT&lt;FLEXRAY&gt;]</th>
<th>Receives the OFS message with CRC as per test co-ordination requirement for CRC.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receives OFS message with format</td>
<td></td>
</tr>
<tr>
<td>Byte 0: Type = 0x40</td>
<td></td>
</tr>
<tr>
<td>Byte 1: CRC</td>
<td></td>
</tr>
<tr>
<td>Byte 2: D = Time Domain 16 (Bit 7 to Bit 4)</td>
<td></td>
</tr>
<tr>
<td>SC = &lt;Sequence Counter 0x01&gt;(Bit 3 to Bit 0)</td>
<td></td>
</tr>
<tr>
<td>Byte 3: User Byte 0</td>
<td></td>
</tr>
<tr>
<td>Byte 4: User Byte 1</td>
<td></td>
</tr>
<tr>
<td>Byte 5: User Byte 2</td>
<td></td>
</tr>
<tr>
<td>Byte 6-11: OfsTimeSec = &lt;StbM_TimeStampType.seconds 0x000000000064&gt;</td>
<td></td>
</tr>
<tr>
<td>Byte 12-15: OfsTimeNSec = &lt;StbM_TimeStampType.nanoseconds 0x00000000&gt;</td>
<td></td>
</tr>
</tbody>
</table>

### Post-conditions

None

5.4.3 [ATS_GTS_01249] Time Slave: Reception of SYNC frames (CRC_VALIDATED) over FlexRay, Synchronize Local Time Base and share the current time to active customers

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Time Slave: Reception of SYNC frames (CRC_VALIDATED) over FlexRay, Synchronize Local Time Base and share the current time to active customers</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01249</td>
</tr>
</tbody>
</table>
### Acceptance Test Specification of Global Time Synchronization

**AUTOSAR TC Release 1.2.0**

**Document ID**: AUTOSAR_ATS_GlobalTimeSynchronization

**Release**: AUTOSAR Confidential

<table>
<thead>
<tr>
<th>Releases</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Affected Modules</strong></td>
</tr>
<tr>
<td><strong>State</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trace to Requirement on Acceptance Test Document</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Trace to SWS Item</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trace to Requirement on Acceptance Test Document</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Trace to Requirement on Acceptance Test Document</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Requirements / Reference to Test Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Case UC02.02</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Configuration Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>StbM</strong>: StbMSynchronizedTimeBaseIdentifier = 1</td>
</tr>
<tr>
<td><strong>FrTSyn</strong>: FrTSynRxCrcValidated = CRC_VALIDATED</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Verify that FrTSyn shall receive Synchronization frames after validating CRC.</strong></td>
</tr>
<tr>
<td><strong>Verify that StbM shall synchronize its local time base on reception of Time Base from FrTSyn.</strong></td>
</tr>
<tr>
<td><strong>Verify that Upper Tester shall get the valid current time, current time in extended format, user data using Client-Server Interface.</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Needed Adaptation to other Releases</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Not Applicable</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pre-conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SUT shall be initialized.</strong></td>
</tr>
</tbody>
</table>

### Main Test Execution

<table>
<thead>
<tr>
<th>Test Steps</th>
<th>Pass Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test Steps</strong></td>
<td><strong>Pass Criteria</strong></td>
</tr>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>[LT&lt;FLEXRAY&gt;]</strong></td>
</tr>
<tr>
<td>Transmit SYNC message with timeBaseId = 1</td>
<td>Receives SYNC message with CRC value in below format</td>
</tr>
<tr>
<td>StbM_UserDataType/User Data Byte 0 = 0xAA</td>
<td>Byte 0: Type = 0x20</td>
</tr>
<tr>
<td>StbM_UserDataType/User Data Byte 1 = 0xBB</td>
<td>Byte 1: CRC</td>
</tr>
<tr>
<td>Byte CRC = Calculated CRC</td>
<td>Byte 2: D = &lt;Time Domain = 1&gt; (Bit 7 to Bit 4)</td>
</tr>
<tr>
<td>And time value as T0</td>
<td>SC = &lt;Sequence Counter = 0&gt; (Bit 3 to Bit 0)</td>
</tr>
<tr>
<td>Where T0 = TSYNC + (MacroticksPerCycle * (64 - currentCycle) - currentMacroticks) * MacrotickDuration</td>
<td>Byte 3: FCNT= &lt;FlexRay Cycle Counter = 0&gt; (Bit 7 to Bit 2)</td>
</tr>
<tr>
<td>And TSYNC value as given below:</td>
<td>SGW (Bit 1)</td>
</tr>
<tr>
<td>StbM_TimeStampType.seconds =</td>
<td>SyncToSubDomain = 1</td>
</tr>
</tbody>
</table>

| **[SUT]** |
| Receives SYNC message with CRC value in below format |
| Byte 0: Type = 0x20 |
| Byte 1: CRC |
| Byte 2: D = <Time Domain = 1> (Bit 7 to Bit 4) |
| SC = <Sequence Counter = 0> (Bit 3 to Bit 0) |
| Byte 3: FCNT= <FlexRay Cycle Counter = 0> (Bit 7 to Bit 2) |
| SGW (Bit 1) |
| SyncToSubDomain = 1 |

<table>
<thead>
<tr>
<th><strong>Pass Criteria</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
</tr>
<tr>
<td>[LT&lt;FLEXRAY&gt;]</td>
</tr>
<tr>
<td>Transmit SYNC message with timeBaseId = 1</td>
</tr>
<tr>
<td>StbM_UserDataType/User Data Byte 0 = 0xAA</td>
</tr>
<tr>
<td>StbM_UserDataType/User Data Byte 1 = 0xBB</td>
</tr>
<tr>
<td>Byte CRC = Calculated CRC</td>
</tr>
<tr>
<td>And time value as T0</td>
</tr>
<tr>
<td>Where T0 = TSYNC + (MacroticksPerCycle * (64 - currentCycle) - currentMacroticks) * MacrotickDuration</td>
</tr>
<tr>
<td>And TSYNC value as given below:</td>
</tr>
<tr>
<td>StbM_TimeStampType.seconds =</td>
</tr>
</tbody>
</table>

| **[SUT]** |
| Receives SYNC message with CRC value in below format |
| Byte 0: Type = 0x20 |
| Byte 1: CRC |
| Byte 2: D = <Time Domain = 1> (Bit 7 to Bit 4) |
| SC = <Sequence Counter = 0> (Bit 3 to Bit 0) |
| Byte 3: FCNT= <FlexRay Cycle Counter = 0> (Bit 7 to Bit 2) |
| SGW (Bit 1) |
| SyncToSubDomain = 1 |
### Step 2

[SUT] StbM updates its time base with values

\[
\text{StbM\_TimeStampType.nanoseconds} = \text{tA.Nano} \\
\text{StbM\_TimeStampType.seconds} = \text{tA.Sec} \\
\text{StbM\_TimeStampType.secondsHi} = \text{tA.SecHi}
\]

Time updated in StbM, \( tA = tA + \text{ToleranceTime\_FrTSyn} \).

ToleranceTime\_FrTSyn = Max one main function of FrTSyn as reception is asynchronous.

### Step 3

[CP] Wait 200ms

### Step 4

[CP] Start RUN\_TimeUser

### Step 5

[RUN<\text{RUN\_TimeUser}>]

Execute Rte\_Call\_Client\_GetCurrentTime

Rte\_Call returns RTE\_E\_OK.

Time read at Time user shall be:

\[
\text{StbM\_TimeStampType.nanoseconds} = \text{tA.Nano} + \langle\text{TestWaitTime = 200ms}\rangle \\
\text{StbM\_TimeStampType.seconds} = \text{tA.Sec} \\
\text{StbM\_TimeStampType.secondsHi} = \text{tA.SecHi}
\]

### Post-conditions

None
5.4.4 [ATS_GTS_01250] Time Slave: Reception of SYNC frames (CRC_IGNORED) over FlexRay, Synchronize Local Time Base and share the current time to active customers.

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Time Slave: Reception of SYNC frames (CRC_IGNORED) over FlexRay, Synchronize Local Time Base and share the current time to active customers.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01250</td>
</tr>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>StbM, FrTSyn</td>
</tr>
<tr>
<td>State</td>
<td>reviewed</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00134</td>
</tr>
<tr>
<td>Trace to SWS Item</td>
<td>TimeSyncOverFlexRay: SWS_FrTSyn_00046</td>
</tr>
<tr>
<td>Requirements / Reference to Test Environment</td>
<td>Use Case UC02.02</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>StbM: StbMSynchronizedTimeBaseIdentifier = 1</td>
</tr>
<tr>
<td>FrTSyn: FrTSynRxCrcValidated = CRC_IGNORED</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Aim is to Verify that FrTSyn shall call StbM to update time base on reception of SYNC message even with invalid CRC when FrTSynRxCrcValidated = CRC_IGNORED.</td>
</tr>
<tr>
<td>Needed Adaptation to other Releases</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Pre-conditions</td>
<td>SUT shall be initialized.</td>
</tr>
</tbody>
</table>

Main Test Execution

<table>
<thead>
<tr>
<th>Test Steps</th>
<th>Pass Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td></td>
</tr>
<tr>
<td>[LT&lt;FLEXRAY&gt;]</td>
<td>[SUT]</td>
</tr>
<tr>
<td>Transmit SYNC message with timeBaseId = 1</td>
<td>Receives SYNC message ignoring CRC value (if there) in below format</td>
</tr>
<tr>
<td>StbM._UserDataType.User Data Byte 0 = 0xAA</td>
<td>Byte 0: Type = 0x10</td>
</tr>
<tr>
<td>StbM._UserDataType.User Data Byte 1 = 0xBB</td>
<td>Byte 1: User Byte 2</td>
</tr>
<tr>
<td>And time value as T0</td>
<td>Byte 2: D = &lt;Time Domain = 1&gt; (Bit 7 to Bit 4)</td>
</tr>
<tr>
<td>Where T0 = TSYNC + (MacroticksPerCycle * (64 - currentCycle) - currentMacroticks) * MacrotickDuration</td>
<td>Byte 3: FCNT= &lt;FlexRay Cycle Counter = 0&gt; (Bit 7 to Bit 2)</td>
</tr>
<tr>
<td>And TSYNC value as given below:</td>
<td>SGW (Bit 1)</td>
</tr>
<tr>
<td>StbM._TimeStampType.seconds =</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>[SUT]</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td></td>
<td>StbM updates its time base with values</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.nanoseconds = tA.Nano</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.seconds = tA.Sec</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.secondsHi = tA.SecHi</td>
</tr>
<tr>
<td></td>
<td>Time updated in StbM, tA = tA + ToleranceTime_FrTSyn.</td>
</tr>
<tr>
<td></td>
<td>ToleranceTime_FrTSyn = Max one main function of FrTSyn as reception is asynchronous.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>[CP]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Wait 200ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>[CP]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Start RUN_TimeUser</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>[RUN-RUN_TimeUser&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Execute Rte_Call_ Client_GetCurrentTime</td>
</tr>
<tr>
<td></td>
<td>Rte_Call returns RTE_E_OK.</td>
</tr>
<tr>
<td></td>
<td>Time read at Time user shall be:</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.nanoseconds = tA.Nano + &lt;TestWaitTime = 200ms&gt;</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.seconds = tA.Sec</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.secondsHi = tA.SecHi</td>
</tr>
</tbody>
</table>

| Post- | None |
### 5.4.5 [ATS_GTS_01256] Time Slave: Handling of SYNC message Sequence Mismatch failures

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Time Slave: Handling of SYNC message Sequence Mismatch failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01256</td>
</tr>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>SbM, FrTSyn</td>
</tr>
<tr>
<td>State</td>
<td>reviewed</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00134</td>
</tr>
<tr>
<td>Trace to SWS Item</td>
<td>TimeSyncOverFlexRay: SWS_FrTSyn_00048</td>
</tr>
<tr>
<td>Requirement s / Reference to Test Environment</td>
<td>Use Case UC02.02</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>SbM: StbMSynchronizedTimeBaseIdentifier = 1</td>
</tr>
<tr>
<td></td>
<td>FrTSyn: FrTSynRxCrValidated = CRC_IGNORED</td>
</tr>
<tr>
<td>Summary</td>
<td>Aim is to verify that if Sequence Counter Jump Width between two SYNC messages is greater than FrTSynGlobalTimeSequenceCounterJumpWidth, then messages will be ignored.</td>
</tr>
<tr>
<td>Needed Adaptation to other Releases</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Pre-conditions</td>
<td>SUT shall be initialized.</td>
</tr>
</tbody>
</table>

#### Main Test Execution

<table>
<thead>
<tr>
<th>Test Steps</th>
<th>Pass Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td></td>
</tr>
<tr>
<td>[LT&lt;FLEXRAY&gt;]</td>
<td></td>
</tr>
<tr>
<td>Transmit SYNC message with TimeBaseId = 1</td>
<td>Receives SYNC message ignoring CRC value in below format</td>
</tr>
<tr>
<td>StbM_UserDataType.User Data Byte 0 = 0xAA</td>
<td>Byte 0: Type = 0x10</td>
</tr>
<tr>
<td>StbM_UserDataType.User Data Byte 1 = 0xBB</td>
<td>Byte 1: User Byte 2</td>
</tr>
<tr>
<td></td>
<td>Byte 2: D = &lt;Time Domain = 1&gt; (Bit 7 to Bit 4)</td>
</tr>
<tr>
<td></td>
<td>SC = &lt;Sequence Counter = 0&gt; (Bit 3 to Bit 0)</td>
</tr>
</tbody>
</table>
And time value as $T_{0a}$

Where $T_{0a} = TSYNC + (\text{MacroticksPerCycle} \times (64 - \text{currentCycle}) - \text{currentMacroticks}) \times \text{MacrotickDuration}$

And $TSYNC$ value as given below:

$\text{StbM\_TimeStampType.seconds} = 0x00000000$

$\text{StbM\_TimeStampType.secondsHi} = 0x0000$

$\text{StbM\_TimeStampType.nanoseconds} = 0x00000000$

Byte 3: $\text{FCNT=} <\text{FlexRay Cycle Counter = 0}> (\text{Bit 7 to Bit 2})$

SGW (Bit 1)

SyncToSubDomain = 1

reserved (Bit 0)

Byte 4: User Byte 0

Byte 5: User Byte 1

Byte 6-11: SyncTimeSec = $T_{0a}.secondsHi$ and $T_{0a}.seconds$

Byte 12-15: SyncTimeNSec = $T_{0a}.nanoseconds$

---

### Step 2

- **[SUT]**
  - FrTSyn shall update the time base of StbM
  - $\text{StbM\_TimeStampType.nanoseconds} = tA.Nano$
  - $\text{StbM\_TimeStampType.seconds} = tA .Sec$
  - $\text{StbM\_TimeStampType.secondsHi} = tA .SecHi$

  Time updated in StbM, $tA = tA + \text{ToleranceTime\_FrTSyn}$.

  $\text{ToleranceTime\_FrTSyn} = \text{Max one main function of FrTSyn as reception is asynchronous.}$

---

### Step 3

- **[CP]**
  - Wait 200ms

---

### Step 4

- **[CP]**
  - Start $\text{RUN\_TimeUser}$

---

### Step 5

- **[RUN<RUN\_TimeUser>]**
  - Execute $\text{Rte\_Call\_Client\_GetCurrentTime}$

  Rte_Call returns RTE\_E\_OK.

  Time read at Time user shall be:

  $\text{StbM\_TimeStampType.nanoseconds} = tA.Nano + <\text{TestWaitTime = 200ms}>.$

  $\text{StbM\_TimeStampType.seconds} = tA .Sec$
### Step 6

**[LT<FLEXRAY>]**

Transmit next periodic SYNC message with:

- timeBaseId = 1
- StbM_UserDataType.User Data Byte 0 = 0xEE
- StbM_UserDataType.User Data Byte 1 = 0xFF

And time value as T0b:

Where T0b = TSYNC + (MacroticksPerCycle * (64 - currentCycle) - currentMacroticks) * MacrotickDuration

And TSYNC value as given below:

- StbM_TimeStampType.seconds = 0x00000E10
- StbM_TimeStampType.secondsHi = 0x0000
- StbM_TimeStampType.nanosecond s = 0x00000000

Sequence counter = 3

**[SUT]**

Time base is not updated as per incoming time base as sequence counter value is greater than FrTSynGlobalTimeSequenceCounterJumpWidth

### Step 7

**[CP]**

Wait 200ms

### Step 8

**[CP]**

Start RUN_TimeUser

### Step 9

**[RUN<RUN_TimeUser>]**

Execute Rte_Call_Client_GetCurrentTime

Rte_Call returns RTE_E_OK.

Time read at Time user shall be:

- StbM_TimeStampType.nanoseconds = tA.Nano + <TestWaitTime = 200ms> + <TestWaitTime = 200ms>.
- StbM_TimeStampType.seconds = tA .Sec + FrTSynGlobalTimeTxPeriod (320ms)
### 5.4.6 [ATS_GTS_01277] Time Slave: Reception of Offset frames (CRC_NOT_VALIDATED) over FlexRay, Synchronize Local Time Base and share the current time to active customers.

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Time Slave: Reception of Offset frames (CRC_NOT_VALIDATED) over FlexRay, Synchronize Local Time Base and share the current time to active customers.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01277</td>
</tr>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>StbM, FrTSyn</td>
</tr>
<tr>
<td>State</td>
<td>reviewed</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00134</td>
</tr>
<tr>
<td>Trace to SWS Item</td>
<td>TimeSyncOverFlexRay: SWS_FrTSyn_00011</td>
</tr>
<tr>
<td>TimeSyncOverFlexRay: SWS_FrTSyn_00047</td>
<td></td>
</tr>
<tr>
<td>Requirements / Reference to Test Environment</td>
<td>Use Case UC02.02</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>StbM: StbMSynchronizedTimeBaseIdentifier = 16</td>
</tr>
<tr>
<td>FrTSyn: FrTSynRxCrcValidated = CRC_NOT_VALIDATED</td>
<td></td>
</tr>
<tr>
<td>CanTSynGlobalTimeOfsDataListIndex = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Aim is to:</td>
</tr>
<tr>
<td>Verify that FrTSyn shall Receive the offset time base periodically via offset message.</td>
<td></td>
</tr>
<tr>
<td>Verify that StbM shall synchronize its local offset time base on reception of Time Base from FrTSyn.</td>
<td></td>
</tr>
<tr>
<td>Needed Adaptation to other Releases</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Pre-conditions</td>
<td>SUT shall be initialized.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Main Test Execution</th>
<th>Test Steps</th>
<th>Pass Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>[LT&lt;FLEXRAY&gt;]</td>
<td>[SUT]</td>
</tr>
</tbody>
</table>
Transmit SYNC message with:

- **TimeBaseId = 1**
- And time value as **T0**

Where **T0 = TSYNC + (MacroticksPerCycle * (64 - currentCycle) - currentMacroticks) * MacrotickDuration**

And TSYNC value as given below:

- **StbM_TimeStampType.seconds = 0x00000E10**
- **StbM_TimeStampType.secondsHi = 0x0000**
- **StbM_TimeStampType.nanoseconds = 0x00000000**

Get the time stamp values as below:

- **StbM_TimeStampType.nanoseconds = tA.Nano**
- **StbM_TimeStampType.seconds = tA.Sec**
- **StbM_TimeStampType.secondsHi = tA.SecHi**

Time updated in StbM, **tA = tA + ToleranceTime_FrTSyn.**

ToleranceTime_FrTSyn = Max one main function of FrTSyn as reception is asynchronous.

### Step 2

[CP]

Wait 200ms

### Step 3

[LT<FLEXRAY>] [SUT]

Transmit OFS message with:

- **TimeBaseId = 16**
- **StbM_TimeStampType.nanoseconds = 0x00000000**
- **StbM_TimeStampType.seconds = 0x0000000A (10d)**
- **StbM_TimeStampType.secondsHi = 0x0000**

Get the offset time stamp values as below:

- **StbM_TimeStampType.nanoseconds = tA.Nano**
- **StbM_TimeStampType.seconds = tA.Sec**
- **StbM_TimeStampType.secondsHi = tA.SecHi**

### Step 4

[CP]

Wait 200ms

### Step 5

[CP]

Start RUN_TimeUser

### Step 6

[RUN<RUN_TimeUser>]

Execute Rte_Call_Client_GetCurrentTime

Rte_Call returns RTE_E_OK.

Time read at Time user shall be:

- **StbM_TimeStampType.nanoseconds = tA.Nano + <TestWaitTime = 200ms> + <TestWaitTime = 200ms>**
- **StbM_TimeStampType.seconds = tA.Sec + <offset time = 0x0000000A>**
5.4.7 [ATS_GTS_01280] Time Slave: Handling of Offset message Sequence Mismatch failures

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Time Slave: Handling of Offset message Sequence Mismatch failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01280</td>
</tr>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>StbM, FrTSyn</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00134</td>
</tr>
<tr>
<td>Trace to SWS Item</td>
<td>TimeSyncOverFlexRay: SWS_FrTSyn_00048</td>
</tr>
<tr>
<td>Requirement s / Reference to Test Environment</td>
<td>Use Case UC02.02</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>StbM:</td>
</tr>
<tr>
<td></td>
<td>StbMSynchronizedTimeBaseldentifier = 16</td>
</tr>
<tr>
<td></td>
<td>FrTSyn:</td>
</tr>
<tr>
<td></td>
<td>FrTSynRxCrcValidated = CRC_IGNORED</td>
</tr>
<tr>
<td>Summary</td>
<td>Aim is to verify that if Sequence Counter Jump Width between two OFS messages is greater than FrTSynGlobalTimeSequenceCounterJumpWidth, then messages will be ignored.</td>
</tr>
<tr>
<td>Needed Adaptation to other Releases</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Pre-conditions</td>
<td>SUT shall be initialized.</td>
</tr>
<tr>
<td></td>
<td>StbM shall be initialized with base time:</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.nanoseconds = 0x00000000</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.seconds = 0x00000000</td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.secondsHi = 0x0000</td>
</tr>
<tr>
<td>Main Test Execution</td>
<td></td>
</tr>
<tr>
<td>Test Steps</td>
<td>Pass Criteria</td>
</tr>
<tr>
<td>Step 1</td>
<td></td>
</tr>
<tr>
<td>[LT&lt;FLEXRAY&gt;]</td>
<td>[SUT]</td>
</tr>
<tr>
<td>Transmit OFS message with TimeBaseId = 16</td>
<td>Receives OFS message with format Byte 0: Type = 0x30</td>
</tr>
<tr>
<td>StbM_TimeStampType.nanoseconds</td>
<td></td>
</tr>
</tbody>
</table>
### Acceptance Test Specification of Global Time Synchronization

**AUTOSAR TC Release 1.2.0**

**Document ID**: AUTOSAR_ATS_GlobalTimeSynchronization

**StbM_TimeStampType**

- **seconds**: 0x0000000014
- **secondsHi**: 0x0000

**Step 1**

**[SUT]**

FrTSyn shall update the offset time base of StbM

- **StbM_TimeStampType.nanoseconds**: tA.Nano
- **StbM_TimeStampType.seconds**: tA.Sec
- **StbM_TimeStampType.secondsHi**: tA.SecHi

**Step 2**

**[SUT]**

FrTSyn shall update the offset time base of StbM

- **StbM_TimeStampType.nanoseconds**: tA.Nano
- **StbM_TimeStampType.seconds**: tA.Sec
- **StbM_TimeStampType.secondsHi**: tA.SecHi

**Step 3**

**[CP]**

Wait 200ms

**Step 4**

**[CP]**

Start RUN_TimeUser

**Step 5**

**[RUN<RUN_TimeUser>]**

Execute Rte_Call_Client_GetCurrentTime.

Rte_Call returns RTE_E_OK.

Time read at Time user shall be:

- **StbM_TimeStampType.nanoseconds**: tA.Nano + <TestWaitTime = 200ms>.
- **StbM_TimeStampType.seconds**: tA.Sec
- **StbM_TimeStampType.secondsHi**: tA.SecHi

**Step 6**

**[LT]**

Transmit next periodic OFS message with

- **TimeBaseld**: 16
- **StbM_TimeStampType.seconds**:

<table>
<thead>
<tr>
<th>Byte 1: User Byte 3</th>
<th>Byte 2: D = &lt;Time Domain = 16&gt; (Bit 7 to Bit 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte 3: User Byte 0</td>
<td>Byte 4: User Byte 1</td>
</tr>
<tr>
<td>Byte 5: User Byte 2</td>
<td>Byte 6-11: OfsTimeSec = 0x00000000000014</td>
</tr>
<tr>
<td>Byte 12-15: OfsTimeNSec = 0x00000000000000014</td>
<td>Byte 10: OfsTimeNSec = 0x000000000000000000</td>
</tr>
</tbody>
</table>

**[SUT]**

Time base is not updated as per new incoming time base as sequence counter value is greater than FrTSynGlobalTimeSequenceCounterJumpWidth.
<table>
<thead>
<tr>
<th>Step 7</th>
<th>[CP]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Wait 200ms.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th>[CP]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Start RUN_TimeUser</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 9</th>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Execute Rte_Call_Client_GetCurrentTime</td>
<td>Rte_Call returns RTE_E_OK.</td>
</tr>
<tr>
<td></td>
<td>Time read at Time user shall be:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.nanoseconds = tA.Nano + &lt;TestWaitTime = 200ms&gt; + &lt;TestWaitTime = 200ms&gt;.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.seconds = tA .Sec + FrTSynGlobalTimeTxPeriod (320ms) + FrTSynGlobalTimeTxPeriod (320ms)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.secondsHi = tA .SecHi</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeBaseStatusType.TIMEOUT is set to 1.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Post-conditions</th>
<th>None</th>
</tr>
</thead>
</table>

0x00000050
StbM_TimeStampType.secondsHi = 0x0000
Sequence number = 3
6 RS_BRF_01660 - Global Time Synchronization over Multiple Bus

6.1 General Test Objective and Approach

This Test Specification intends to cover the Global Time Synchronization feature of StbM CanTSync, and FrTSync as described in the AUTOSAR Feature [RS_BRF_01660].

The tests use a test bench environment and Embedded Software Components that use the feature.

This test case document has been established to cover the following features:

- Features not tested in this test suite are either tested in 'Global Time Synchronization over CAN' or 'Global Time Synchronization over FlexRay'.

This specification gives the description of required tests environments (test bench, uses case, configuration files) and detailed tests cases for executing tests.
6.1.1  Test System

6.1.1.1  Overview on Architecture

In order to cover the required features / sub-features coverage, the environment has been separated in several use cases.

Test Cases are derived based on below use cases

6.1.1.1.1  UC 03.01.01 Global Time Master (Single Time Domain) over both CAN and FlexRay

SUT acts as Global time Master over Single time domain and Transmits Time Synchronization messages over CAN and FlexRay bus.

As shown in below figure, Functionalities of Global Time master for single time domain are tested over CAN and FlexRay in the use case 03.01.01
6.1.1.1.2 UC 03.01.02 Global Time Master (Multiple Time Domain) over both CAN and FlexRay

SUT acts as Global time Master over multiple time domain and Transmits Time Synchronization messages over CAN and FlexRay bus.

As shown in the below figure, functionalities of Global Time master for multiple time domain are tested over CAN and FlexRay in the use case 03.01.02.
6.1.1.1.3 UC 03.02.01 Time Gateway: Time Slave over FlexRay and time master over CAN

SUT acts as Time Gateway. It receives time base, offset time base and Synchronizes Local time to Global Time base and transmits the received time base to its slaves. Time Slave over FlexRay and Time Master on CAN

As shown in below figure, Functionalities of Time Gateway - Time Slave over FlexRay and Time Master over CAN in the use case 03.02.01
6.1.1.1.4 UC 03.02.02 Time Gateway: Time Slave over CAN and time master over FlexRay

SUT acts as Time Gateway with Time Slave over CAN and Time Master on FlexRay

As shown in the figure, Functionalities of Time Gateway - Time Slave over CAN and Time Master over FlexRay in the use case 03.02.02

6.1.1.1.5 UC 03.02.03 Time Gateway: Time Slave over CAN (Network 1) and time master over CAN (Network 2)
SUT acts as Time Gateway with Time Slave over CAN1 and Time Master on CAN2.

As shown in below figure, Functionalities of Time Gateway - Time Slave over CAN1 and Time Master over CAN2 in the use case 03.02.03

6.1.1.2 Specific Requirements
Not Applicable.

6.1.1.3 Test Coordination Requirements
UC 03.01, UC 03.02: Time Master over CAN and/or FlexRay
- Test System (LT <CAN>) shall read the CanTSyn CAN Frames and decode the same as per Frame Format provided in AUTOSAR_SWS_TimeSyncOverCAN.
- Test System (LT <FlexRay>) shall read the FrTSyn FlexRay Frames and decode the same as per Frame Format provided in AUTOSAR_SWS_TimeSyncOverFlexRay.
UC 03.01, UC 03.02: Time Slave over CAN and/or FlexRay

- Test System (LT <CAN>) shall encode the CanTSyn CAN Frames as per Frame Format provided in AUTOSAR_SWS_TimeSyncOverCAN and transmit over bus.
- Test System (LT <FlexRay>) shall encode the FrTSyn FlexRay Frames as per Frame Format provided in AUTOSAR_SWS_TimeSyncOverFlexRay and transmit over bus.

Requirements for CRC Calculation

- Test System (LT <CAN>) shall use the Crc_CalculateCRC8H2F() (Refer AUTOSAR Specification of CRC Routines AUTOSAR_SWS_CRCLibrary.pdf) to calculate the CRC of the Frame. Below are the parameters used for CRC calculation:
  - The CRC start value shall be 0xFF.
  - The CRC final XOR-value shall be 0xFF.
  - The CRC polynomial shall be 0x2F.
  - The DataIDList shall be same as provided in CanTSyn Static Configuration.

6.1.2 Test Case Design

- Global Time Master over Multiple Bus (Single Time Domain)
  The Time domain for both CAN and FlexRay will be same. And Synchronization frames shall be sent over bus as per respective TSyn module configuration
- Global Time Master over Multiple Bus (Single Time Domain)
  The Time domain for both CAN and FlexRay will be different. And Synchronization frames shall be sent over bus with respective time domain information and respective TSyn module configuration

Below figures provides detailed info about how the time base information is validated by the tester.

Fig: Global Time Master over Multiple Bus (Single/Multiple Time Domain) Test Design

- Time Gateway - Time Slave on FlexRay and Time Master on CAN
Fig: Time Gateway - Time Slave on FlexRay and Time Master on CAN Test Design

- **Time Gateway - Time Slave on CAN and Time Master on FlexRay**

Fig: Time Gateway - Time Slave on CAN and Time Master on FlexRay Test Design

- **Time Gateway - Time Slave on CAN (Network 1) and Time Master on CAN (Network 2)**
6.2 Configuration requirements

The configuration can be divided into two separate parts. The test configuration describes variables used to parameterize the test case. The static configuration describes the necessary settings of the DUT in order to allow a test case to perform.

6.2.1 Test Configuration

Communication data base for CanTSyn is depicted below

<table>
<thead>
<tr>
<th>test configuration parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-Pdu</td>
<td></td>
</tr>
<tr>
<td>AT_101_Ipdu</td>
<td>101</td>
</tr>
<tr>
<td>AT_102_Ipdu</td>
<td>102</td>
</tr>
</tbody>
</table>

Communication data base for FrTSyn is depicted below

<table>
<thead>
<tr>
<th>test configuration parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Value</td>
</tr>
<tr>
<td>FrIf Tx Pdu</td>
<td>AT_201_Ipdu201</td>
</tr>
<tr>
<td>FrIfTxPdu. FrIfImmediate</td>
<td>FALSE</td>
</tr>
<tr>
<td>FrIfFrameTriggering.FrIfBaseCycle</td>
<td>0</td>
</tr>
<tr>
<td>FrIfFrameTriggering.FrIfCycleRepetition</td>
<td>64</td>
</tr>
<tr>
<td>FrIfFrameTriggering.FrIfSlotId</td>
<td>2</td>
</tr>
<tr>
<td>FrIf Rx Pdu</td>
<td>AT_202_Ipdu202</td>
</tr>
</tbody>
</table>
### 6.2.2 Static Configuration

#### 6.2.2.1 Static Configuration Groups

<table>
<thead>
<tr>
<th>SCG_ATS_GlobalTimeSync_Single and Multiple time Domain</th>
<th>System Description Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM</td>
<td></td>
</tr>
<tr>
<td>SystemTemplate::GlobalTime::GlobalTimeMaster.isSystemWideGlobalTimeMaster</td>
<td>TRUE</td>
</tr>
<tr>
<td>CanTSyn</td>
<td></td>
</tr>
<tr>
<td>SystemTemplate::GlobalTime::GlobalTimeMaster.syncPeriod</td>
<td>2000ms</td>
</tr>
<tr>
<td>SystemTemplate::GlobalTime::CAN::GlobalTimeCanMaster.syncConfirmationTimeout</td>
<td>80ms</td>
</tr>
<tr>
<td>SystemTemplate::GlobalTime::GlobalTimeDomain.globalTimePdu</td>
<td>Ref. To PDU</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ECU Configuration Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM</td>
</tr>
<tr>
<td>StbM.StbMGeneral.StbMMainFunctionPeriod</td>
</tr>
<tr>
<td>StbM.StbMSynchronizedTimeBase.StbMLocalTimeRef</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CanTSyn</th>
</tr>
</thead>
<tbody>
<tr>
<td>CanTSyn.CanTSynGeneral.CanTSynMainFunctionPeriod</td>
</tr>
<tr>
<td>CanTSyn.CanTSynGlobalTimeDomain. CanTSynGlobalTimeDomainId</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FrTSyn</th>
</tr>
</thead>
<tbody>
<tr>
<td>FrTSyn.FrTSynGeneral.FrTSynMainFunctionPeriod</td>
</tr>
<tr>
<td>FrTSyn.FrTSynGlobalTimeDomain. FrTSynSynchronizedTimeBaseRef</td>
</tr>
<tr>
<td>FrTSyn.CanTSynGlobalTimeDomain.FrTSynGlobalTimeMaster. FrTSynGlobalTimeMasterPdu.</td>
</tr>
</tbody>
</table>

Test Cases

UC 03.01
### SCG_ATS_GlobalTimeSync_Time Gateway

<table>
<thead>
<tr>
<th>System Description Parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM</td>
<td></td>
</tr>
<tr>
<td>SystemTemplate::GlobalTime::GlobalTimeMaster.isSystemWideGlobalTimeMaster</td>
<td>FALSE</td>
</tr>
<tr>
<td>CanTSyn</td>
<td>SystemTemplate::GlobalTime::GlobalTimeDomain.globalTimePdu</td>
</tr>
<tr>
<td>FrTSyn</td>
<td>SystemTemplate::GlobalTime::GlobalTimeDomain.subDomain</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ECU Configuration Parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM</td>
<td></td>
</tr>
<tr>
<td>StbM.StbMGeneral.StbMMainFunctionPeriod</td>
<td>5ms</td>
</tr>
<tr>
<td>StbM.StbMSynchronizedTimeBase.StbMLocalTimeRef</td>
<td>Ref. to OSCounter</td>
</tr>
<tr>
<td>CanTSyn</td>
<td></td>
</tr>
<tr>
<td>CanTSyn.CanTSynGeneral.CanTSynMainFunctionPeriod</td>
<td>5ms</td>
</tr>
<tr>
<td>CanTSynCanTSynGlobalTimeDomain.</td>
<td>0</td>
</tr>
<tr>
<td>CanTSynGlobalTimeMasterConfirmationHandleId</td>
<td></td>
</tr>
<tr>
<td>FrTSyn</td>
<td></td>
</tr>
<tr>
<td>FrTSyn.FrTSynGeneral.FrTSynMainFunctionPeriod</td>
<td>5ms</td>
</tr>
<tr>
<td>FrTSynCanTSynGlobalTimeDomain.FrTSynGlobalTimeMaster.</td>
<td></td>
</tr>
<tr>
<td>FrTSynGlobalTimeMasterPdu.</td>
<td>0</td>
</tr>
<tr>
<td>FrTSynGlobalTimeMasterConfirmationHandleId</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Cases</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>UC 03.02</td>
<td></td>
</tr>
</tbody>
</table>

### 6.2.2.2 Required System Description

Refer Section 5.2.2.1

### 6.2.2.3 Required ECU Configuration

Refer Section 5.2.2.1
6.2.2.4 Required Software Components

Fig: UC 03.01.01 - Global Time Master over Multiple Bus (Single Time Domain)
SWC Overview

Fig: UC 03.01.02 - Global Time Master over Multiple Bus (Multiple Time Domain)
SWC Overview
Fig: UC 03.02.01, UC 03.02.02 and UC 03.02.03 – Time Gateway SWC Overview

### 6.2.2.4.1 SWC Client GlobalTime_Provider

<table>
<thead>
<tr>
<th>SWC Name</th>
<th>GlobalTime_Provider</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Client_SetGlobalTime</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>RPortPrototype</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>GlobalTime_Master_Interface</td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td></td>
</tr>
</tbody>
</table>

**PORTS**

| **Name** | Client_SetUserData |
| **Type** | RPortPrototype |
| **Interface** | GlobalTime_Master_Interface |
| **Requirements** | |

| **Name** | Client_SetOffset |
| **Type** | RPortPrototype |
| **Interface** | GlobalTime_Master_Interface |
| **Requirements** | |
## RUNNABLE ENTITIES

<table>
<thead>
<tr>
<th>Name</th>
<th>RUN_GlobalTimeProvider</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements</strong></td>
<td>Runnable shall be invoked by TCP</td>
</tr>
</tbody>
</table>

### ServerCallPoint

<table>
<thead>
<tr>
<th>Name</th>
<th>sscp_GlobalTimeProvider</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>SynchronousServerCallPoint</td>
</tr>
<tr>
<td><strong>Access to</strong></td>
<td>Client_SetGlobalTime (Write operation)</td>
</tr>
<tr>
<td></td>
<td>Client_SetUserData (Write operation)</td>
</tr>
<tr>
<td></td>
<td>Client_SetOffset (Write operation)</td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td></td>
</tr>
</tbody>
</table>

## 6.2.2.4.2 SWC Client Time_User

### SWC Name

<table>
<thead>
<tr>
<th>Name</th>
<th>Time_User</th>
</tr>
</thead>
</table>

### PORTS

<table>
<thead>
<tr>
<th>Name</th>
<th>Client_GetCurrentTime</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>RPortPrototype</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>GlobalTime_Slave_Interface</td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td></td>
</tr>
</tbody>
</table>

### RUNNABLE ENTITIES

<table>
<thead>
<tr>
<th>Name</th>
<th>Client_GetCurrentTimeExtended</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>RPortPrototype</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>GlobalTime_Slave_Interface</td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>RUN_TimeUser</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements</strong></td>
<td>Runnable shall be invoked by TCP</td>
</tr>
</tbody>
</table>

### ServerCallPoint

<table>
<thead>
<tr>
<th>Name</th>
<th>sscp_TimeUser</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>SynchronousServerCallPoint</td>
</tr>
<tr>
<td><strong>Access to</strong></td>
<td>Client_GetCurrentTime (Read operation)</td>
</tr>
<tr>
<td></td>
<td>Client_GetCurrentTimeExtended</td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td></td>
</tr>
</tbody>
</table>
### 6.2.2.4.3 SWC Server StbM

<table>
<thead>
<tr>
<th>SWC Name</th>
<th>StbM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Server_SetGlobalTime</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>PPortPrototype</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>GlobalTime_Master_Interface</td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td>(Read operation)</td>
</tr>
</tbody>
</table>

| **Name**  | Server_SetOffset |
| **Type**  | PPortPrototype |
| **Interface**  | GlobalTime_Master_Interface |
| **Requirements**  | |

| **Name**  | Server_SetUserData |
| **Type**  | PPortPrototype |
| **Interface**  | GlobalTime_Master_Interface |
| **Requirements**  | |

| **Name**  | Server_GetCurrentTime |
| **Type**  | PPortPrototype |
| **Interface**  | GlobalTime_Slave_Interface |
| **Requirements**  | |

| **Name**  | Server_GetCurrentTimeExtended |
| **Type**  | PPortPrototype |
| **Interface**  | GlobalTime_Slave_Interface |
| **Requirements**  | |

| **Name**  | StbM_SetGlobalTime |
| **Requirements**  | |
### RUNNABLE ENTITIES

<table>
<thead>
<tr>
<th>Started by Event</th>
<th>Name</th>
<th>Type</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OIE_SetGlobalTime</td>
<td>OperationInvokedEvent</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Port: Server_SetGlobalTime</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Operation: Read</td>
<td></td>
</tr>
</tbody>
</table>

**ServerCallPoint**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>OIE_SetOffset</td>
<td>OperationInvokedEvent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port: Server_SetOffset</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Operation: Read</td>
<td></td>
</tr>
</tbody>
</table>

**Started by Event**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM_SetOffset</td>
<td>OperationInvokedEvent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port: Server_SetOffset</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Operation: Read</td>
<td></td>
</tr>
</tbody>
</table>

**Started by Event**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM_SetUserData</td>
<td>OperationInvokedEvent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port: Server_SetUserData</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Operation: Read</td>
<td></td>
</tr>
</tbody>
</table>

**Started by Event**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM_GetCurrentTime</td>
<td>OperationInvokedEvent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port: Server_GetCurrentTime</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Operation: Read</td>
<td></td>
</tr>
</tbody>
</table>

**Started by Event**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM_GetCurrentTimeExtended</td>
<td>OperationInvokedEvent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port: Server_GetCurrentTimeExtended</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Operation: Read</td>
<td></td>
</tr>
</tbody>
</table>
6.3 Re-usable Test Steps

Not Applicable

6.4 Test Cases

6.4.1 [ATS_GTS_01281] Global Time Master over Multiple Bus (Single Time Domain): Setting of Global Time base and user data and sending of SYNC frame over CAN and FlexRay

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Global Time Master over Multiple Bus (Single Time Domain): Setting of Global Time base and user data and sending of SYNC frame over CAN and FlexRay</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01281</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>StbM, CanTSyn, FrTSyn</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00133, ATR: ATR_ATR_00134</td>
</tr>
<tr>
<td>Requirements Reference to Test Environment</td>
<td>Use Case UC03.01</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>StbM: StbMSynchronizedTimeBaseIdentifier = 1, CanTSyn: CanTSynGlobalTimeTxCrcSecured = CRC_NOT_SUPPORTED, FrTSyn: FrTSynGlobalTimeTxCrcSecured = CRC_NOT_SUPPORTED</td>
</tr>
<tr>
<td>Summary</td>
<td>Aim is to test the functionality of global time master a time domain and Transmission of Synchronization message over FlexRay and CAN Bus. Verify that StbM accepts the global time base from Upper Tester using client-server</td>
</tr>
</tbody>
</table>
Acceptance Test Specification of Global Time Synchronization

AUTOSAR TC Release 1.2.0

Document ID: AUTOSAR_ATS_GlobalTimeSynchronization

- AUTOSAR Confidential -

| Interface. |
| Verify that CanTSyn shall transmit the global time base to time slave periodically via SYNC and FUP message. |
| Verify that FrTSyn shall transmit the global time base to time slave periodically via SYNC message. |

##### Needed Adaptation to other Releases

Not Applicable.

##### Pre-conditions

SUT shall be initialized.

##### Main Test Execution

<table>
<thead>
<tr>
<th>Test Steps</th>
<th>Pass Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>[CP] Start RUN_GlobalTimeProvider.</td>
</tr>
</tbody>
</table>
|            | **Step 2** [RUN>RUN_GlobalTimeProvider>]
|            | Execute Rte_Call_Client_SetGlobalTime and Rte_Call_Client_SetUserData with below values: |
|            | timeBaseld = 1 |
|            | StbM_TimeStampType.nanoseconds = 0x00000000 |
|            | StbM_TimeStampType.seconds = 0x0000E10 |
|            | StbM_TimeStampType.secondsHi = 0x0000 |
|            | StbM_UserDataType.User Data Byte 0 = 0xAA |
|            | StbM_UserDataType.User Data Byte 1 = 0xBB |
|            | StbM_UserDataType.User Data Byte 2 = 0xCC |
|            | [RUN<RUN_GlobalTimeProvider>]
|            | Rte_Call returns RTE_E_OK. |
|            | **Step 3** [CP] Wait 100ms. |
|            | **Step 4** [CP] Start RUN_TimeUser. |
|            | **Step 5** [RUN>RUN_TimeUser>]
|            | Execute Rte_Call_Client_GetCurrentTime. |
|            | [RUN<RUN_TimeUser>]
<p>|            | Time received from Time user shall be |</p>
<table>
<thead>
<tr>
<th>Step 6</th>
<th>[LT&lt;CAN&gt;]</th>
<th>[LT]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Receives the SYNC message.</td>
<td>Receives frame format as mentioned below:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Byte 0: Type = 0x10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Byte 1: User Byte 1 = 0xBB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Byte 2: D = 0x1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SC = 0x0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Byte 3: User Byte 0 = 0xAA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Byte 4-7: SyncTimeSec = StbM_TimeStampType.seconds</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>[LT&lt;CAN&gt;]</th>
<th>[LT]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Receives the FUP message</td>
<td>Receives frame format mentioned below:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Byte 0: Type = 0x18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Byte 1: User Byte 2 = 0xCC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Byte 2: D = 0x1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SC = 0x0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Byte 3: reserved (Bit 7 to Bit 3) = 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SGW (Bit 2) = 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OVS = Overflow of seconds (Bit 1 to Bit 0)</td>
</tr>
</tbody>
</table>

as mentioned below:

StbM_TimeStampType.nanoseconds = 0x00000000 + <TestWaitTime = 100ms>
StbM_TimeStampType.seconds = 0x00000E10
StbM_TimeStampType.secondsHi = 0x0000
StbM_UserDataType.User Data Byte 0 = 0xAA
StbM_UserDataType.User Data Byte 1 = 0xBB
StbM_UserDataType.User Data Byte 2 = 0xCC
<table>
<thead>
<tr>
<th>Step 8</th>
<th><strong>[LT&lt;CAN&gt;]</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Receives Global time base.</strong></td>
<td></td>
</tr>
<tr>
<td>Store the time as base for next periodic message processing</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 9</th>
<th><strong>[LT&lt;CAN&gt;]</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Receives the next periodic SYNC and FUP message</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 10</th>
<th><strong>[LT&lt;FLEXRAY&gt;]</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Receives the SYNC message</strong></td>
<td></td>
</tr>
</tbody>
</table>

---

Byte 4-7: `SyncTimeNSec = StbM_TimeStampType.nanoseconds + <Test Wait Time = 100ms>`

---

**Step 8**

- Receives Global time base.
- Store the time as base for next periodic message processing

**Step 9**

- Receives the next periodic SYNC and FUP message

**Step 10**

- Receives the SYNC message
0 (Bit 7 to Bit 2)
SGW (Bit 1)
SyncToGTM = 0
reserved (Bit 0), default: 0
Byte 4: User Byte 0 = 0xAA
Byte 5: User Byte 1 = 0xBB
Byte 6-11: SyncTimeSec = 48 Bit time stamp in seconds
Byte 12-15: SyncTimeNSec = 32 Bit time stamp in nanoseconds

Step 11

[LT<FLEXRAY>]
Receives Global time base.
Store the time as base for next periodic message processing

[LT]
Get the time stamp values as below:
StbM_TimeStampType.nanoseconds = tA.Nano
StbM_TimeStampType.seconds = tA.Sec
StbM_TimeStampType.secondsHi = tA.SecHi

Step 12

[LT<FLEXRAY>]
Receives the next periodic SYNC message

[LT]
Time stamp values shall be as mentioned below:
StbM_TimeStampType.nanoseconds = tB.Nano
StbM_TimeStampType.seconds = tB.Sec
StbM_TimeStampType.secondsHi = tB.SecHi
StbM_UserDataType.User Data Byte 0 = 0xAA
StbM_UserDataType.User Data Byte 1 = 0xBB
StbM_UserDataType.User Data Byte 2 = 0xCC

The difference between two time base shall be
6.4.2 [ATS_GTS_01257] Global Time Master over Multiple Bus (Multiple Time Domain): Setting of Global Time base and user data and sending of SYNC frame over CAN and FlexRay

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Global Time Master over Multiple Bus (Multiple Time Domain): Setting of Global Time base and user data and sending of SYNC frame over CAN and FlexRay</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01257</td>
</tr>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>StbM, CanTSyn, FrTSyn</td>
</tr>
<tr>
<td>State</td>
<td>reviewed</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00133 ATR: ATR_ATR_00134</td>
</tr>
<tr>
<td>Trace to SWS Item</td>
<td>TimeSyncOverCAN: SWS_CanTSyn_00038 TimeSyncOverFlexRay: SWS_FrTSyn_00023</td>
</tr>
<tr>
<td>Requirements Reference to Test Environment</td>
<td>Use Case UC03.01</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>StbM: StbMSynchronizedTimeBaseIdentifier = 1, 2 CanTSyn: CanTSynGlobalTimeTxCrcSecured = CRC_NOT_SUPPORTED FrTSyn: FrTSynGlobalTimeTxCrcSecured = CRC_NOT_SUPPORTED</td>
</tr>
<tr>
<td>Summary</td>
<td>Aim is to test the functionality of global time master on Multiple Time Domain and Transmission of Synchronization Messages over Multiple Bus Verify that StbM accepts the global time base from Upper Tester using client-server interface for Time Domain 1 and Time Domain 2 Verify that CanTSyn as Global time master shall transmit the Global time base to time slaves periodically via SYNC and FUP message. Verify that FrTSyn as Global time master shall transmit the Global Time Base to time slave periodically via SYNC message.</td>
</tr>
<tr>
<td>Needed Adaptation to other Releases</td>
<td>Not Applicable.</td>
</tr>
<tr>
<td>Pre-conditions</td>
<td>SUT shall be initialized.</td>
</tr>
</tbody>
</table>

---

**Main Test Execution**

### Test Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Test Steps</th>
<th>Pass Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[CP]</td>
<td>Start RUN_GlobalTimeProvider.</td>
</tr>
<tr>
<td>Step 2</td>
<td>[RUN&lt;RUN_GlobalTimeProvider&gt;] Execute Rte_Call_Client_SetGlobalTime and Rte_Call_Client_SetUserData with below values:</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>timeBaseId = 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.nanoseconds = 0x00000000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.seconds = 0x00000E10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.secondsHi = 0x0000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_UserDataType.User Data Byte 0 = 0xAA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_UserDataType.User Data Byte 1 = 0xBB</td>
<td></td>
</tr>
<tr>
<td>[RUN&lt;RUN_GlobalTimeProvider&gt;]</td>
<td>Rte_Call returns RTE_E_OK.</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>[CP] Wait 100ms.</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>[RUN&lt;RUN_TimeUser&gt;] Execute Rte_Call_Client_GetCurrentTime.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[RUN&lt;RUN_TimeUser&gt;] Rte_Call returns RTE_E_OK.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Time received from Time user shall be as mentioned below:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.nanoseconds = 0x00000000 + &lt;TestWaitTime = 100ms&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.seconds = 0x00000E10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.secondsHi = 0x0000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_UserDataType.User Data Byte 0 = 0xAA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_UserDataType.User Data Byte 1 = 0xBB</td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>[LT&lt;CAN&gt;] Receives the SYNC message</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[LT] Receives frame format as mentioned below:</td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td>[LT&lt;CAN&gt;]</td>
<td>[LT]</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
<td>------</td>
</tr>
<tr>
<td>Receives the FUP message</td>
<td>Receives frame format mentioned below:</td>
<td></td>
</tr>
<tr>
<td>Byte 0: Type = 0x10</td>
<td>Byte 0: Type = 0x18</td>
<td></td>
</tr>
<tr>
<td>Byte 1: User Byte 1 = 0xBB</td>
<td>Byte 1: User Byte 2 = 0x00</td>
<td></td>
</tr>
<tr>
<td>Byte 2: D = 0x1</td>
<td>Byte 2: D = 0x1</td>
<td></td>
</tr>
<tr>
<td>SC = 0x0</td>
<td>SC = 0x0</td>
<td></td>
</tr>
<tr>
<td>Byte 3: User Byte 0 = 0xAA</td>
<td>Byte 3: reserved (Bit 7 to Bit 3) = 0</td>
<td></td>
</tr>
<tr>
<td>SGW (Bit 2) = 0</td>
<td>OVS = Overflow of seconds (Bit 1 to Bit 0)</td>
<td></td>
</tr>
<tr>
<td>Byte 4-7: SyncTimeSec = StbM_TimeStampType.seconds</td>
<td>Byte 4-7: SyncTimeNSec = StbM_TimeStampType.nanoseconds + &lt;Test Wait Time = 100ms&gt;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th>[LT&lt;CAN&gt;]</th>
<th>[LT]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receives Global time base.</td>
<td>Get the time stamp values as below:</td>
<td></td>
</tr>
<tr>
<td>Store the time as base for next periodic message processing</td>
<td>StbM_TimeStampType.nanoseconds = tA.Nano</td>
<td></td>
</tr>
<tr>
<td>StbM_TimeStampType.seconds = tA.Sec</td>
<td>StbM_TimeStampType.seconds = tA.Sec</td>
<td></td>
</tr>
<tr>
<td>StbM_TimeStampType.secondsHi = tA.SecHi</td>
<td>StbM_TimeStampType.secondsHi = tA.SecHi</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 9</th>
<th>[LT&lt;CAN&gt;]</th>
<th>[LT]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receives the next periodic SYNC and FUP message</td>
<td>Time stamp values shall be as mentioned below:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StbM_TimeStampType.nanoseconds = tB.Nano</td>
<td></td>
</tr>
</tbody>
</table>
Acceptance Test Specification of Global Time Synchronization

AUTOSAR
TC
Release 1.2.0

Document ID: AUTOSAR_ATS_GlobalTimeSynchronization

StbM_TimeStampType.seconds = tB.Sec
StbM_TimeStampType.secondsHi = tB.SecHi
StbM_UserDataType.User Data Byte 0 = 0xAA

The difference between two time base shall be

\[ tB - tA = CanTSynGlobalTimeTxPeriod (2 second) + CanTSynGlobalTimeTxFollowUpOffset. \]

Step 10 [CP]
Start RUN_GlobalTimeProvider.

Step 11 [RUN<RUN_GlobalTimeProvider>]
Execute Rte_Call_Client_SetGlobalTime and Rte_Call_Client_SetUserData with below values:

- timeBaseId = 2
- StbM_TimeStampType.nanoseconds = 0x00000000
- StbM_TimeStampType.seconds = 0x00001C20
- StbM_TimeStampType.secondsHi = 0x0000
- StbM_UserDataType.User Data Byte 0 = 0xAA
- StbM_UserDataType.User Data Byte 1 = 0xBB
- StbM_UserDataType.User Data Byte 2 = 0xCC

[RUN<RUN_GlobalTimeProvider>]
Rte_Call returns RTE_E_OK.

Step 12 [CP]
Wait 100ms.

Step 13 [CP]
Start RUN_TimeUser.

Step 14 [RUN<RUN_TimeUser>]
Execute Rte_Call_Client_GetCurrentTime for timeBaseId = 2

[RUN<RUN_TimeUser>]
Rte_Call returns RTE_E_OK.

Time received from Time user shall be
<table>
<thead>
<tr>
<th>Step 15</th>
<th>[LT&lt;FLEXRAY&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receives the SYNC message</td>
<td>[LT]</td>
</tr>
<tr>
<td>Receives frame format as mentioned below:</td>
<td></td>
</tr>
<tr>
<td>Byte 0: Type = 0x10</td>
<td></td>
</tr>
<tr>
<td>Byte 1: UserByte 2 = 0xCC</td>
<td></td>
</tr>
<tr>
<td>Byte 2: D = 0x1</td>
<td></td>
</tr>
<tr>
<td>SC = 0x0</td>
<td></td>
</tr>
<tr>
<td>Byte 3: FCNT= FlexRay Cycle Counter 0 (Bit 7 to Bit 2)</td>
<td></td>
</tr>
<tr>
<td>SGW (Bit 1)</td>
<td></td>
</tr>
<tr>
<td>SyncToGTM = 0</td>
<td></td>
</tr>
<tr>
<td>reserved (Bit 0), default: 0</td>
<td></td>
</tr>
<tr>
<td>Byte 4: User Byte 0 = 0xAA</td>
<td></td>
</tr>
<tr>
<td>Byte 5: User Byte 1 = 0xBB</td>
<td></td>
</tr>
<tr>
<td>Byte 6-11: SyncTimeSec = 48 Bit time stamp in seconds</td>
<td></td>
</tr>
<tr>
<td>Byte 12-15: SyncTimeNSec = 32 Bit time stamp in nanoseconds + &lt;Test Wait Time = 100ms&gt;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 16</th>
<th>[LT&lt;FLEXRAY&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[LT]</td>
<td>[LT]</td>
</tr>
</tbody>
</table>
Receives Global time base. Store the time as base for next periodic message processing.

Get the time stamp values as below:

| StbM_TimeStampType.nanoseconds = tA.Nano |
| StbM_TimeStampType.seconds = tA.Sec |
| StbM_TimeStampType.secondsHi = tA.SecHi |

Step 17

[LT<FLEXRAY>]

Receives the next periodic SYNC message

[LT]

Time stamp values shall be as mentioned below:

| StbM_TimeStampType.nanoseconds = tB.Nano |
| StbM_TimeStampType.seconds = tB.Sec |
| StbM_TimeStampType.secondsHi = tB.SecHi |

| StbM_UserDataType.User Data Byte 0 = 0xAA |
| StbM_UserDataType.User Data Byte 1 = 0xBB |
| StbM_UserDataType.User Data Byte 2 = 0xCC |

The difference between two time base shall be

\[
 tB - tA = FrTSynGlobalTimeTxPeriod (320ms).
\]

Post-conditions

None

6.4.3 [ATS_GTS_01258] Time Gateway- Time Slave on FlexRay and Time Master on CAN

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Time Gateway- Time Slave on FlexRay and Time Master on CAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01258</td>
</tr>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>StbM, CanTSyn, FrTSyn</td>
</tr>
<tr>
<td>State</td>
<td>reviewed</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00133 ATR: ATR_ATR_00134</td>
</tr>
</tbody>
</table>
Acceptance Test Specification of Global Time Synchronization
AUTOSAR
TC Release 1.2.0

<table>
<thead>
<tr>
<th>Trace to SWS Item</th>
<th>TimeSyncOverCAN: SWS_CanTSyn_00028</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TimeSyncOverCAN: SWS_CanTSyn_00030</td>
</tr>
<tr>
<td></td>
<td>TimeSyncOverFlexRay: SWS_FrTSyn_00046</td>
</tr>
</tbody>
</table>

| Requirements / Reference to Test Environment | Use Case UC03.02 |

<table>
<thead>
<tr>
<th>Configuration Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM:</td>
</tr>
<tr>
<td>StbM_IsSystemWideGlobalTimeMaster = FALSE</td>
</tr>
<tr>
<td>StbM_SynchronizedTimeBaseIdentifier = 1</td>
</tr>
<tr>
<td>CanTSyn:</td>
</tr>
<tr>
<td>CanTSyn_GlobalTimeDomainId = 1</td>
</tr>
<tr>
<td>CanTSyn_SynchronizedTimeBaseRef = Reference to StbM_SynchronizedTimeBase</td>
</tr>
<tr>
<td>CanTSyn_GlobalTimeTx_Crc_Secured = CRC_NOT_SUPPORTED</td>
</tr>
<tr>
<td>CanTSyn_GlobalTimeTx_FollowUp_Offset = 100ms</td>
</tr>
<tr>
<td>CanTSyn_GlobalTimeTx_Period = 2000ms</td>
</tr>
<tr>
<td>CanTSyn_MasterConfirmation_Timeout = 80ms</td>
</tr>
<tr>
<td>FrTSyn:</td>
</tr>
<tr>
<td>FrTSyn_Rx_Crc_Validated = CRC_NOT_VALIDATED</td>
</tr>
<tr>
<td>FrTSyn_GlobalTimeDomainId = 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aim is to verify the functionality of time gateway</td>
</tr>
<tr>
<td>Verify that FrTSyn as time slave shall receive the global time base periodically via SYNC message.</td>
</tr>
<tr>
<td>Verify that StbM synchronizes the local time as per the received global time base from FrTSyn (using API StbM_BusSetGlobalTime)</td>
</tr>
<tr>
<td>Verify that CanTSyn as time master shall transmit the global time base received from global time domain to time subdomain periodically via SYNC and FUP message respectively.</td>
</tr>
<tr>
<td>Verify that UT shall get the valid current time using Client-Server Interface.</td>
</tr>
</tbody>
</table>

| Needed Adaptation to other Releases | Not Applicable |

| Pre-conditions | SUT shall be initialized. |

| Main Test Execution |

<table>
<thead>
<tr>
<th>Test Steps</th>
<th>Pass Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td></td>
</tr>
<tr>
<td>[LT&lt;FLEXRAY&gt;]</td>
<td>[SUT]</td>
</tr>
<tr>
<td>Transmit SYNC message with timeBaseId = 1</td>
<td>Receives SYNC message ignoring CRC value (if there) in below format</td>
</tr>
<tr>
<td>StbM_UserDataType.User Data Byte 0 = 0xAA</td>
<td>Byte 0: Type = 0x10</td>
</tr>
<tr>
<td>StbM_UserDataType.User Data Byte 1 = 0xBB</td>
<td>Byte 1: User Byte 2</td>
</tr>
<tr>
<td>And time value as T0</td>
<td>Byte 2: D = &lt;Time Domain = 1&gt; (Bit 7 to Bit 4)</td>
</tr>
<tr>
<td>Where T0 = TSYNC + (MacroticksPerCycle * SC)</td>
<td>SC = &lt;Sequence Counter = 0&gt; (Bit 3 to Bit 0)</td>
</tr>
</tbody>
</table>
And TSYNC value as given below:

\[
\begin{align*}
\text{StbM\_TimeStampType\_seconds} &= 0x00000E10 \\
\text{StbM\_TimeStampType\_secondsHi} &= 0x0000 \\
\text{StbM\_TimeStampType\_nanoseconds} &= 0x00000000 \\
\end{align*}
\]

Step 2

[SUT]

StbM updates its time base with values

\[
\begin{align*}
\text{StbM\_TimeStampType\_nanoseconds} &= tA\_Nano \\
\text{StbM\_TimeStampType\_seconds} &= tA\_Sec \\
\text{StbM\_TimeStampType\_secondsHi} &= tA\_SecHi \\
\end{align*}
\]

Step 3

[CP]

Wait 200ms

Step 4

[CP]

Start RUN\_TimeUser

Step 5

[RUN\_RUN\_TimeUser>]

Execute Rte\_Call\_Client\_GetCurrentTime

Rte\_Call returns RTE\_E\_OK.

Time read at Time user shall be:

\[
\begin{align*}
\text{StbM\_TimeStampType\_nanoseconds} &= tA\_Nano + \text{<TestWaitTime = 200ms>} \\
\text{StbM\_TimeStampType\_seconds} &= tA\_Sec \\
\text{StbM\_TimeStampType\_secondsHi} &= tA\_SecHi \\
\end{align*}
\]

Step 6

[LT\<CAN>]

Receives the SYNC message

[LT]

Receives SYNC message in the
### Step 7

**[LT<CAN>]**  
Receives the FUP message

**[LT]**  
Receives FUP message in the format mentioned below:

- Byte 0: Type = 0x18
- Byte 1: User Byte 2 = 0x00
- Byte 2: D = 0x1
- SC = 0x0
- Byte 3: reserved (Bit 7 to Bit 3) = 0
- SGW (Bit 2) = 0
- OVS = Overflow of seconds (Bit 1 to Bit 0)
- Byte 4-7: SyncTimeNSec = StbM_TimeStampType.nanoseconds

### Step 8

**[LT<CAN>]**  
Receives Global time base.  
Store the time as base for next periodic message processing

**[LT]**  
Get the timestamp values as below:

- StbM_TimeStampType.nanoseconds = tC.Nano
- StbM_TimeStampType.seconds = tC.Sec
- StbM_TimeStampType.secondsHi = tC.SecHi

Time Received by LT<CAN> should be tC (Time base transmitted by LT<FlexRay> + ToleranceTime_FrTSyn + ToleranceTime_CanTSyn)
ToleranceTime_FrTSyn = Max one main function of FrTSyn as reception is asynchronous

ToleranceTime_CanTSyn = Max one main function of CanTSyn as gateway is asynchronous.

| Post-conditions | None |

### 6.4.4 [ATS_GTS_01259] Time Gateway - Time Slave on CAN and Time Master on FlexRay

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Time Gateway - Time Slave on CAN and Time Master on FlexRay</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01259</td>
</tr>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>StbM, CanTSyn, FrTSyn</td>
</tr>
<tr>
<td>State</td>
<td>reviewed</td>
</tr>
</tbody>
</table>
| Trace to Requirement on Acceptance Test Document | ATR: ATR_ATR_00133
ATR: ATR_ATR_00134 |
| Trace to SWS Item | SynchronizedTimeBaseManager: SWS_StbM_00184
TimeSyncOverCAN: SWS_CanTSyn_00064
TimeSyncOverCAN: SWS_CanTSyn_00072
TimeSyncOverFlexRay: SWS_FrTSyn_00023 |
| Requirements / Reference to Test Environment | Use Case UC03.02 |
| Configuration Parameters | StbM:
StbMIsSystemWideGlobalTimeMaster = FALSE
StbMSynchronizedTimeBaseIdentifier = 1
CanTSyn:
CanTSynGlobalTimeDomainId = 1
CanTSynGlobalTimeFollowUpTimeout = 300ms
CanTSynGlobalTimeSequenceCounterJumpWidth = 2
CanTSynSynchronizedTimeBaseRef = Reference to StbMSynchronizedTimeBase
CanTSynRxCrcValidated = CRC_IGNORED
CanTSynGlobalTimeSlaveHandleId = 0
FrTSyn:
FrTSynGlobalTimeDomainId = 1
FrTSynGlobalTimeSequenceCounterJumpWidth = 2
FrTSynSynchronizedTimeBaseRef = Reference to StbMSynchronizedTimeBase
FrTSynGlobalTimeTxCrcSecured = CRC_NOT_SUPPORTED
FrTSynGlobalTimeTxPeriod = 320ms
FrTSynGlobalTimePduRef Reference to Pdu |
| Summary | Aim is to verify the functionality of time gateway
Verify that CanTSyn as time slave shall receive the global offset time periodically via SYNC and FUP message. |
Acceptance Test Specification of Global Time Synchronization
AUTOSAR TC Release 1.2.0

Verify that StbM synchronizes the offset time as per the received global time base from CanTSyn (using API StbM_SetOffset)

Verify that FrTSyn as time master shall transmit the offset time base received from global time domain to time sub-domain periodically via OFS message respectively.

Verify that UT shall get the valid current time using Client-Server Interface.

<table>
<thead>
<tr>
<th>Needed Adaptation to other Releases</th>
<th>Not Applicable.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Pre-conditions</th>
<th>SUT shall be initialized.</th>
</tr>
</thead>
</table>

### Main Test Execution

#### Test Steps | Pass Criteria
---|---

<table>
<thead>
<tr>
<th><strong>Step 1</strong></th>
<th><strong>[LT&lt;CAN&gt;]</strong></th>
<th><strong>[SUT]</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmit SYNC message with</td>
<td>Receives SYNC message ignoring CRC value in below format</td>
<td></td>
</tr>
<tr>
<td>timeBaseId = 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StbM_TimeStampType.seconds = 0x00000E10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StbM_TimeStampType.secondsHi = 0x0000</td>
<td></td>
<td>Byte 2: D = 0x1</td>
</tr>
<tr>
<td>StbM_UserDataType.User Data Byte 0 = 0xAA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StbM_UserDataType.User Data Byte 1 = 0xBB</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>[LT&lt;CAN&gt;]</strong></td>
<td><strong>[SUT]</strong></td>
</tr>
<tr>
<td>Transmit FUP message with</td>
<td>Receives FUP message in below format</td>
<td></td>
</tr>
<tr>
<td>StbM_TimeStampType.nanoseconds = t4r</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Byte 0: Type = 0x18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Byte 1: User Byte 2 = 0x00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Byte 2: D = 0x1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SC = 0x0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Byte 3: reserved (Bit 7 to Bit 3) = 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SGW (Bit 2) = 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OVS = Overflow of seconds (Bit 1 to Bit 0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Byte 4-7: SyncTimeNSec = StbM_TimeStampType.nanoseconds</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>[SUT]</strong></td>
<td></td>
</tr>
</tbody>
</table>
StbM updates its time base with values
StbM._TimeStampType.nanoseconds = tA.Nano
StbM._TimeStampType.seconds = tA.Sec
StbM._TimeStampType.secondsHi = tA.SecHi

<table>
<thead>
<tr>
<th>Step 4</th>
<th>[CP]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Wait 800ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>[CP]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Start RUN_TimeUser</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Execute Rte_Call_Client_GetCurrentTime</td>
</tr>
</tbody>
</table>

[RUN<RUN_TimeUser>]

Rte_Call returns RTE_E_OK.

Time read at Time user shall be:
StbM._TimeStampType.nanoseconds = tA.Nano + <TestWaitTime = 800ms>
StbM._TimeStampType.seconds = tA.Sec
StbM._TimeStampType.secondsHi = tA.SecHi

<table>
<thead>
<tr>
<th>Step 7</th>
<th>[LT&lt;FLEXRAY&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Receives the SYNC message without CRC validation.</td>
</tr>
</tbody>
</table>

[LT]

Receives frame format as mentioned below:

Byte 0: Type = 0x10
Byte 1: UserByte 2
Byte 2: D = 0x1
   SC = 0x0
Byte 3: FCNT= FlexRay Cycle Counter 0 (Bit 7 to Bit 2)
SGW (Bit 1)
SyncToSubDomain = 1
reserved (Bit 0), default: 0
### 6.4.5 [ATS_GTS_01260] Time Gateway - Time Slave on CAN (Network 1) and Time Master on CAN (Network 2)

<table>
<thead>
<tr>
<th>Test Objective</th>
<th>Time Gateway - Time Slave on CAN (Network 1) and Time Master on CAN (Network 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ATS_GTS_01260</td>
</tr>
<tr>
<td>AUTOSAR Releases</td>
<td>4.2.1/4.2.2</td>
</tr>
<tr>
<td>Affected Modules</td>
<td>StbM, CanTSyn</td>
</tr>
<tr>
<td>State</td>
<td>reviewed</td>
</tr>
<tr>
<td>Trace to Requirement on Acceptance Test Document</td>
<td>ATR: ATR_ATR_00133</td>
</tr>
<tr>
<td>Trace to SWS Item</td>
<td>SynchronizedTimeBaseManager: SWS_StbM_00184</td>
</tr>
<tr>
<td></td>
<td>TimeSyncOverCAN: SWS_CanTSyn_00028</td>
</tr>
</tbody>
</table>

#### Step 8

**[LT<FLEXRAY>]**

Receives Global time base.

Store the time as base for next periodic message processing

**[LT]**

Get the time stamp values as below:

- StbM_TimeStampType.nanoseconds = tC.Nano
- StbM_TimeStampType.seconds = tC.Sec
- StbM_TimeStampType.secondsHi = tC.SecHi

Time Received by LT<FlexRay> should be tC (Time base transmitted by LT<CAN> + ToleranceTime_CanTSyn + ToleranceTime_FrTSyn )

- ToleranceTime_CanTSyn = Max one main function of CanTSyn as reception is asynchronous
- ToleranceTime_FrTSyn = Max one main function of FrTSyn as gateway is asynchronous

**Post-conditions**

None
Acceptance Test Specification of Global Time Synchronization

AUTOSAR TC Release 1.2.0

TimeSyncOverCAN: SWS_CanTSyn_00030
TimeSyncOverCAN: SWS_CanTSyn_00064

Requirements / Reference to Test Environment

Use Case UC03.02

Configuration Parameters

StbM:
- StbMIsSystemWideGlobalTimeMaster = FALSE
- StbMSynchronizedTimeBaseIdentifier = 1

CanTSyn for Master:
- CanTSynGlobalTimeDomainId = 1
- CanTSynGlobalTimeSequenceCounterJumpWidth = 2
- CanTSynSynchronizedTimeBaseRef = Reference to StbMSynchronizedTimeBase
- CanTSynGlobalTimeTxCrcSecured = CRC_NOT_SUPPORTED
- CanTSynGlobalTimeTxFollowUpOffset = 100ms
- CanTSynGlobalTimeTxPeriod = 2000ms
- CanTSynMasterConfirmationTimeout = 80ms

CanTSyn for slave:
- CanTSynGlobalTimeFollowUpTimeout = 300ms
- CanTSynRxCrcValidated = CRC_IGNORED

Summary

Aim is to verify the functionality of time gateway

Verify that CanTSyn as time slave(CAN 1) shall receive the global time base periodically via SYNC and FUP message.

Verify that StbM synchronizes the local time as per the received global time base from CanTSyn (using API StbM_BusSetGlobalTime)

Verify that CanTSyn as time master(CAN 2) shall transmit the global time base received from global time domain to time sub-domain periodically via SYNC and FUP message.

Verify that UT shall get the valid current time using Client-Server Interface.

Needed Adaptation to other Releases

Not Applicable.

Pre-conditions

SUT shall be initialized.

Main Test Execution

Test Steps | Pass Criteria
--- | ---
Step 1 | [LT<CAN>]
- Transmit SYNC message with timeBaseId = 1
- StbM_TimeStampType.seconds = 0x00000E10
- StbM_TimeStampType.secondsHi = 0x0000
- StbM_UserDataType.User Data Byte 0 = 0xAA
- StbM_UserDataType.User Data Byte 1 = [SUT]
- Receives SYNC message ignoring CRC value in below format
- Byte 0: Type = 0x10
- Byte 1: User Byte 1 = 0xBB
- Byte 2: D = 0x1
- SC = 0x0
- Byte 3: User Byte 0 = 0xAA
<table>
<thead>
<tr>
<th>Step 2</th>
<th>[LT&lt;CAN&gt;]</th>
<th>[SUT]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmit FUP message with StbM_TimeStampType.nanoseconds = t4r Byte CRC = Invalid CRC</td>
<td>Receives FUP message ignoring CRC value in below format Byte 0: Type = 0x18 Byte 1: User Byte 2 = 0x00 Byte 2: D = 0x1 SC = 0x0 Byte 3: reserved (Bit 7 to Bit 3) = 0 SGW (Bit 2) = 0 OVS = Overflow of seconds (Bit 1 to Bit 0) Byte 4-7: SyncTimeNSec = StbM_TimeStampType.nanoseconds</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>[SUT]</th>
</tr>
</thead>
<tbody>
<tr>
<td>StbM updates its time base with values StbM_TimeStampType.nanoseconds = tA.Nano StbM_TimeStampType.seconds = tA.Sec StbM_TimeStampType.secondsHi = tA.SecHi</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>[CP]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait 800ms</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>[CP]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start RUN_TimeUser</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
<th>[RUN&lt;RUN_TimeUser&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute Rte_Call_Client_GetCurrentTime</td>
<td>Rte_Call returns RTE_E_OK. Time read at Time user shall be: StbM_TimeStampType.nanoseconds = tA.Nano + &lt;TestWaitTime = 800ms&gt; StbM_TimeStampType.seconds = tA</td>
<td></td>
</tr>
</tbody>
</table>
### Step 7

<table>
<thead>
<tr>
<th><strong>[LT&lt;CAN&gt;]</strong></th>
<th><strong>[LT]</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Receives the SYNC message</td>
<td>Receives SYNC message in the format mentioned below:</td>
</tr>
<tr>
<td>Byte 0: Type = 0x10</td>
<td>Byte 0: Type = 0x10</td>
</tr>
<tr>
<td>Byte 1: User Byte 1 = 0xBB</td>
<td>Byte 1: User Byte 1 = 0xBB</td>
</tr>
<tr>
<td>Byte 2: D = 0x1</td>
<td>Byte 2: D = 0x1</td>
</tr>
<tr>
<td>SC = 0x0</td>
<td>SC = 0x0</td>
</tr>
<tr>
<td>Byte 3: User Byte 0 = 0xAA</td>
<td>Byte 3: User Byte 0 = 0xAA</td>
</tr>
<tr>
<td>Byte 4-7: SyncTimeSec = StbM_TimeStampType.seconds</td>
<td>Byte 4-7: SyncTimeSec = StbM_TimeStampType.seconds</td>
</tr>
</tbody>
</table>

### Step 8

<table>
<thead>
<tr>
<th><strong>[LT&lt;CAN&gt;]</strong></th>
<th><strong>[LT]</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Receives the FUP message</td>
<td>Receives FUP message in the format mentioned below:</td>
</tr>
<tr>
<td>Byte 0: Type = 0x18</td>
<td>Byte 0: Type = 0x18</td>
</tr>
<tr>
<td>Byte 1: User Byte 2 = 0x00</td>
<td>Byte 1: User Byte 2 = 0x00</td>
</tr>
<tr>
<td>Byte 2: D = 0x1</td>
<td>Byte 2: D = 0x1</td>
</tr>
<tr>
<td>SC = 0x0</td>
<td>SC = 0x0</td>
</tr>
<tr>
<td>Byte 3: reserved (Bit 7 to Bit 3) = 0</td>
<td>Byte 3: reserved (Bit 7 to Bit 3) = 0</td>
</tr>
<tr>
<td>SGW (Bit 2) = 0</td>
<td>SGW (Bit 2) = 0</td>
</tr>
<tr>
<td>OVS = Overflow of seconds (Bit 1 to Bit 0)</td>
<td>OVS = Overflow of seconds (Bit 1 to Bit 0)</td>
</tr>
<tr>
<td>Byte 4-7: SyncTimeNSec = StbM_TimeStampType.nanoseconds</td>
<td>Byte 4-7: SyncTimeNSec = StbM_TimeStampType.nanoseconds</td>
</tr>
</tbody>
</table>

### Step 9

<table>
<thead>
<tr>
<th><strong>[LT&lt;CAN&gt;]</strong></th>
<th><strong>[LT]</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Receives Global time base. Store the time as base for next periodic message processing</td>
<td>Get the time stamp values as below:</td>
</tr>
<tr>
<td>StbM_TimeStampType.nanoseconds = tC.Nano</td>
<td>StbM_TimeStampType.nanoseconds = tC.Nano</td>
</tr>
<tr>
<td>StbM_TimeStampType.seconds = tC.Sec</td>
<td>StbM_TimeStampType.seconds = tC.Sec</td>
</tr>
<tr>
<td>StbM_TimeStampType.secondsHi = tA.SecHi</td>
<td>StbM_TimeStampType.secondsHi = tA.SecHi</td>
</tr>
<tr>
<td><strong>Post-conditions</strong></td>
<td>None</td>
</tr>
<tr>
<td>---------------------</td>
<td>------</td>
</tr>
</tbody>
</table>

TC .SecHi

Time Received by LT<CAN> should be TC (Time base transmitted by LT<FlexRay> + ToleranceTime_CanTSyn + ToleranceTime_CanTSyn).

ToleranceTime_CanTSyn = Max one main function of CanTSyn as reception is asynchronous.

ToleranceTime_CanTSyn = Max one main function of CanTSyn as gateway is asynchronous.