Document Title	Acceptance Test Specification of Communication on CAN bus
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	632
Document Classification	Auxiliary

Document Status	Final
Part of AUTOSAR Standard	Acceptance Tests for Classic Platform
Part of Standard Release	1.2.0

Document Change History

Date	Release	Changed by	Change Description
2016-12-15	1.2.0	AUTOSAR Release Management	• Checked and adapted to Classic Platform Release 4.2.2 • New Test Suite for Large Data Com added (RS_BRF_01649)

2015-10-31	1.1.0	AUTOSAR Release Management	• Changes done on Bus-off test suite ATS_COMCAN_00269, step 11
			  • When "CANSM_BOR_TX_CONFIRMATION_POLLING" is disabled, then wait time "CanSMBorTimeTxEnsured" should be considered.
			• Checked and adapted to Classic Platform Release 4.2.1
			  • ATS_COMCAN_00210 (SWS_Com_00305, SWS_Com_00767)
			  • ATS_COMCAN_00211 (SWS_Com_00742, SWS_Com_00743)
			  • ATS_COMCAN_00214 (SWS_Com_00741, SWS_Com_00769)
			• New test cases related to CANIF Software filtering, DLC check: ATS_COMCAN_00715 to ATS_COMCAN_00720
			• Formalization of the point of control and observation for actions and expected results

2014-07-30	1.0.0	AUTOSAR Release Management	• Initial release

Disclaimer

Advice for users

AUTOSAR specifications 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 specifications for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such specifications, 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

Abbreviation / Acronym:	***Description:***
AT	Acceptance Test
CAN	Controller Area Network
ECU	Electronic Control Unit
LT	Lower Tester
NM	Network Management
PCO	Point of Control and Observation
PDU	Protocol Data Unit
RfC	Request for Change
Rx	Reception
SUT	System Under Test
SWC	Software Component
TCP	Test Coordination Procedures
Tx	Transmission
UT	Upper Tester

2 Related Documentation

2.1 Input documents

[1] Specification of Module Efficient COM for Large Data
AUTOSAR_SWS_LargeDataCOM.pdf

[2] Specification of RTE
AUTOSAR_SWS_RTE.pdf

[3] Specification of CAN Transport Layer
AUTOSAR_SWS_CANTransportLayer.pdf

[4] Requirements on Runtime Environment
AUTOSAR_SRS_RTE.pdf

[5] Requirements on Communication
AUTOSAR_SRS_COM.pdf

[6] System Template
AUTOSAR_TPS_SystemTemplate.pdf

[7] Requirements on Acceptance Tests
AUTOSAR_ATR_Requirements.pdf

3 Scope

The following test cases are used to verify the correct behavior of all the communication features which are dependent on the CAN bus.

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_01592 - Data Transfer

4.1 General Test Objective and Approach

This Test Specification intends to cover the Data Transfer feature of the Com as described in the AUTOSAR Feature [RS_BRF_01592].

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:

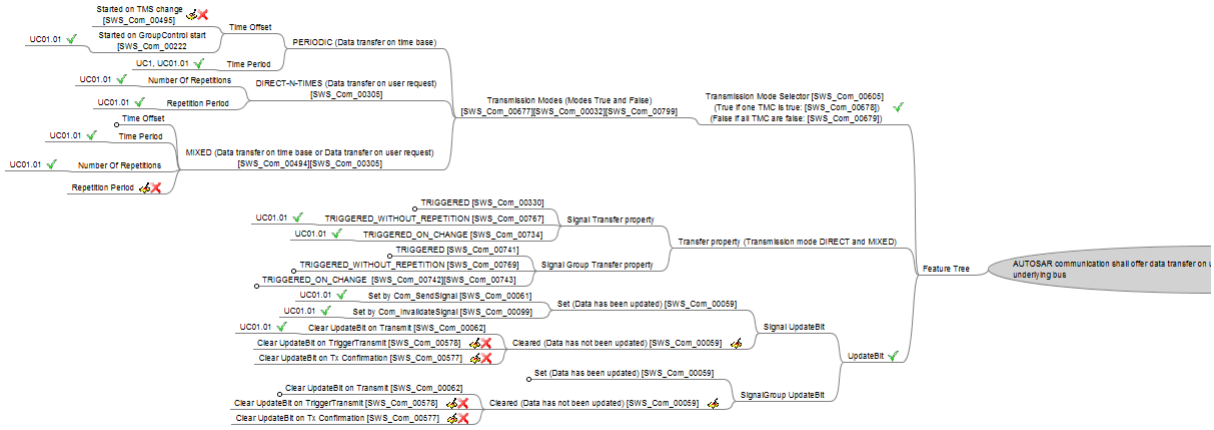

Fig A: Requirement on Data Transfer.

This specification gives the description of required tests environments (test bench, uses case, arxml 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, the different uses cases are created.

4.1.1.1.1 Use case 01.01: CAN Bus

For this use case, the aim is to test the data transfer on CAN bus, In this architecture, COM focus will be on signals with 1Byte, 2 Bytes and 4 Bytes:

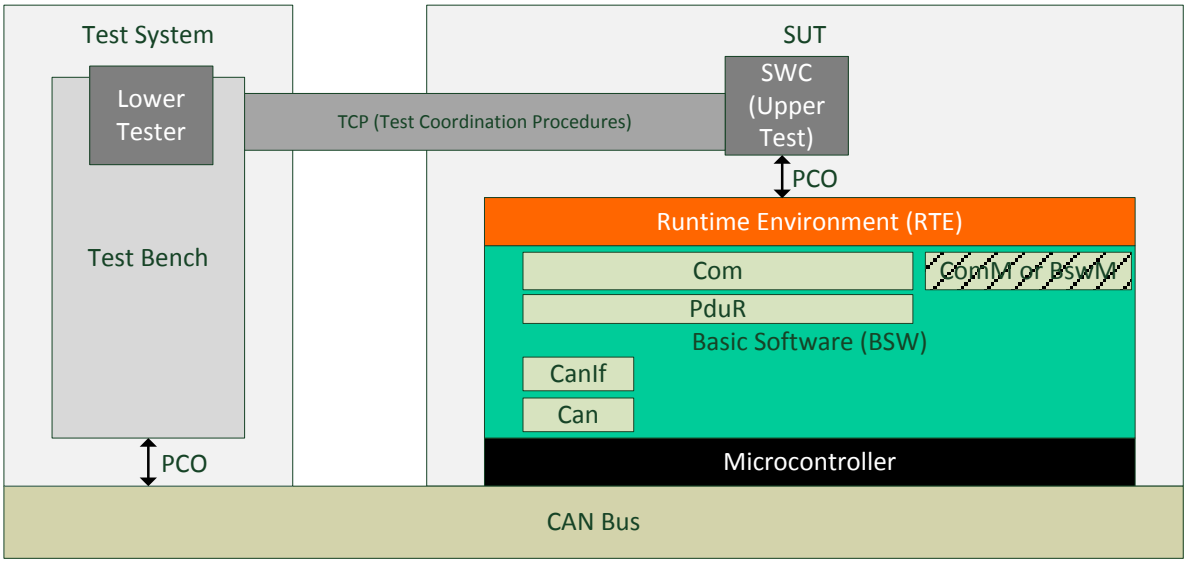

Fig B: Test System Architecture.

The test system architecture consists of Test Bench that executes only test sequencer and gives actions request through Test coordination Procedures to embedded SWC.

4.1.1.2 Specific Requirements

Not Applicable.

4.1.1.3 Test Coordination Requirements

Not Applicable.

4.1.2 Test Configuration

This section describes sets of requirements on configuration.
These sets are later referenced by test cases.
No configuration files are provided. They need to be developed when the test suite is implemented.

4.1.2.1 Required ECU Extract of System Description Files

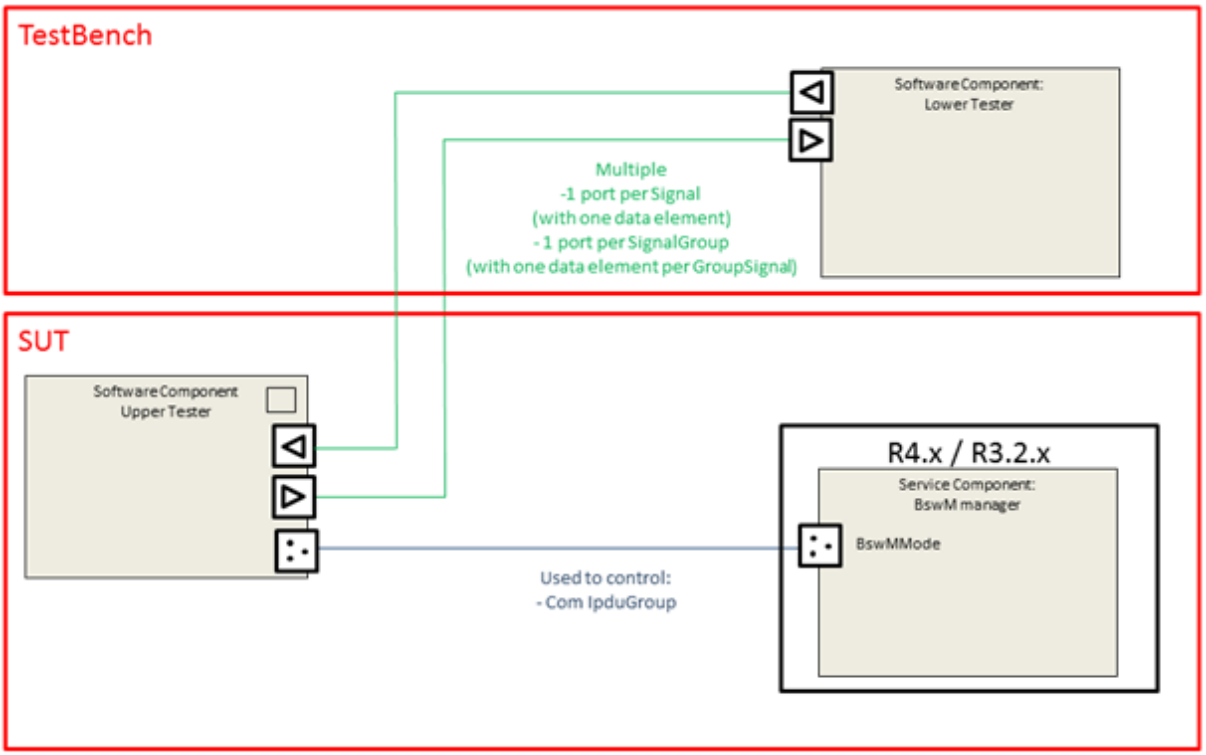

Fig C: SWC Overview.

A Mode-Switch Interface IF_AT_SwC_ActionsBswM must be created. The SWC Upper Tester should trigger BSW action's and BswM read the state through BswMMode Port. BswM shall launch actions according to following table (check 4.3 Test Cases for details):

ModeDeclaration	BswM Actions
IPDU_ACTIVATED	OnEntry: -Start IpduGroup

For the Software Component point of view, for each test case, the communication interfaces are defined as follow:

Port name	Data element type	Dataelement	Mapping	Type
<TestCaseName>_<SignalName>	uint8	<SignalName>	<SignalName>	Signal

<TestCaseName>_<SignalGroupName>	Struct
	{
	uint8: GroupSignal1;
	…
	uint8: GroupSignalx;
	}
		GroupSignal	GroupSignal1->
			<Signal1Name>
			GroupSignal2->
			<Signal2Name>
			<PortName>->
			<SignalGroupName>
				Signal Group

Table 1:

Therefore ports and signals names change according to Test Case Name, but the building rule is the same.

4.1.2.1.1 Use Case 01.01: CAN Bus

The communication database is depicted below:

ComIPduGroup	I-Pdu	SignalGroup	Signal	Tx ECU	Rx ECU
AT_208_IpduGroup	AT_208_Ipdu		AT_208_Sg1	SUT	TestBench
AT_209_IpduGroup	AT_209_Ipdu	AT_209_SgGr1	AT_209_GrSg1	SUT	TestBench
			AT_209_GrSg2		
AT_210_IpduGroup	AT_210_Ipdu		AT_210_Sg1	SUT	TestBench
			AT_210_Sg2		
AT_211_IpduGroup	AT_211_Ipdu	AT_211_SgGr1	AT_211_GrSg1	SUT	TestBench
			AT_211_GrSg2		
		AT_211_SgGr2	AT_211_GrSg3		
			AT_211_GrSg4		
			AT_211_GrSg5		
AT_213_IpduGroup	AT_213_Ipdu		AT_213_Sg1	SUT	TestBench
AT_214_IpduGroup	AT_214_Ipdu	AT_214_SgGr1	AT_214_GrSg1	SUT	TestBench
			AT_214_GrSg2		

Table 2:

4.1.2.2 Required ECU Configuration Description Files

The section describes the common EcuC parameters between test cases that are required by the implementer of the test cases.

No specific configuration requirements for ECU Configuration files as they can be derived from EcuExtract.

4.1.2.3 Required Software Component Description Files

The section describes the SWC-D that are required by the implementer of the test cases.

Refer to Fig C.

4.1.2.4 Mandatory vs. Customizable Parts

Mandatory parameters are listed in Tests Cases (see 4.3 Test Cases).

Customizable parameters are (these values are test case independent):

- ComSignalType (ISignal.networkRepresentationProps.swBaseType), ComSignalLength (baseTypeSize) and ComBitSize (ISignal.length) => must be consistent to associated dataElement
- ComSignalInitValue (ISignal.initValue)
- PduLength (Pdu.length)
- ComBitPosition (ISignalToIPduMapping.startPosition) and ComUpdateBitPosition (ISignalToIPduMapping.updateIndicationBitPosition) values => the location of these elements in the pdu
- CAN frames identifiers

NOTE: ComSignalInitValue and ComSignalDataInvalidValue are specific to test implementer and signal type.

4.1.3 Test Case Design

Not Applicable.

4.2 Re-usable Test Steps

Not Applicable.

4.3 Test Cases

4.3.1 [ATS_COMCAN_00208] Signal on Time Base frame (PERIODIC)

Test Objective	Signal on Time Base frame (PERIODIC)		
ID	ATS_COMCAN_00208	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	Com, PduR, CanIf, Can	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00115 ATR: ATR_ATR_00116		
Trace to SWS Item	COM: SWS_Com_00059 COM: SWS_Com_00061 COM: SWS_Com_00062 COM: SWS_Com_00099 COM: SWS_Com_00222		
Requirements / Reference to Test Environment	Use Case UC01.01		
Configuration Parameters	ComIpdu(SignalIPdu): AT_208_Ipdu1 (Mapped on CAN Frame => CanTopology) - ComIPduDirection(CommConnectorPort.communicationDirection) = SEND - ComTxModeTrue (IPduTiming.TransmissionModeDeclaration.transmissionModeTrueTiming) -- PERIODIC (CyclicTiming) --- timeOffset > 0 --- timePeriod > 0 (different from timeOffset) - ComTxIPduClearUpdateBit(no upstream template parameter) = Transmit ComSignal(ISignalToPduMapping): Sg1 - updateIndicationBitPosition is configured - ComSignalInitValue(ISignal.initValue) = Sg1_Value_Init - ComSignalDataInvalidValue (SwDataDefProps.invalidValue) = Sg1_Value_Invalid		
Summary	Aim: - Check that send signal and invalidate signal are taken into account in the periodic frame Sequence: 1) Action: Start ComIPduGroup - Result: I-PDU is sent out after Offset Time [SWS_Com_00222] - Result: Frames are sent out periodically (see ComTxModeTimePeriod)"		

	- Result: Signal value is initial value - Result: Signal update bit is 0 2) Action: Update signal - Result: Periodic Time is not changed (value is Period Time) - Result: UpdateBit is set to 1 for the first message after update/invalidation, only. [SWS_Com_00059][SWS_Com_00061][SWS_Com_00062] - Result: After successful transmission the UpdateBit is cleared. - Result: Signal value is changed for all new Tx frame occurrences 3) Action: Invalidate signal - Result: Periodic Time is not changed - Result: UpdateBit is set to 1 for the first message after update/invalidation, only.[SWS_Com_00059][SWS_Com_00099][SWS_Com_00062] -Result: After successful transmission the UpdateBit is cleared. - Result: Signal value is the invalid value for all new Tx frame occurrences	
Needed Adaptation to other Releases	None	
Pre-conditions	Com stack is initialized, but ComIPduGroup are not running	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	***[SWC]*** Request ModeSwitch (call to BswMModeRequest port) to IPDU_ACTIVATED (start ComIPduGroup AT_208_IpduGroup)	***[LT<CAN>]*** AT_208_Ipdu is sent out after Offset Time. Next AT_208_Ipdu are sent out every Period Time AT_208_Sg1 value is AT_208_Sg1_Value_Init AT_208_Sg1 update bit is 0
Step 2	***[SWC]*** Update signal AT_208_Sg1 (call Rte_Write() API for Port AT_208_Sg1) with AT_208_Sg1_Value_1	***[LT<CAN>]*** AT_208_Ipdu Periodic Time is not changed (value is Period Time) AT_208_Sg1 UpdateBit is set to 1 in the first send, after that it is 0 AT_208_Sg1 value is now AT_208_Sg1_Value_1
Step 3	***[SWC]*** Invalidate signal AT_208_Sg1 (by calling API Rte_Invalidate())	***[LT<CAN>]*** AT_208_Ipdu Periodic Time is not changed (value is Period Time) AT_208_Sg1 UpdateBit is set to 1 in the first send, after that it is 0 AT_208_Sg1 value is now AT_208_Sg1_Value_Invalid
Post-conditions	Not applicable	

4.3.2 [ATS_COMCAN_00209] SignalGroup on Time Base frame (PERIODIC)

Test Objective	SignalGroup on Time Base frame (PERIODIC)		
ID	ATS_COMCAN_00209	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	Com, PduR, CanIf, Can	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00115 ATR: ATR_ATR_00116		
Trace to SWS Item	COM: SWS_Com_00059 COM: SWS_Com_00062 COM: SWS_Com_00222 COM: SWS_Com_00286 COM: SWS_Com_00801		
Requirements / Reference to Test Environment	Use Case UC01.01		
Configuration Parameters	ComIpdu(SignalIPdu): AT_209_Ipdu1 (Mapped on CAN Frame => CanTopology) - ComIPduDirection(CommConnectorPort.communicationDirection) = SEND - ComTxModeTrue (IPduTiming.TransmissionModeDeclaration.transmissionModeTrueTiming) -- PERIODIC (CyclicTiming) --- timeOffset > 0 --- timePeriod > 0 (different from timeOffset) - ComTxIPduClearUpdateBit(no upstream template parameter) = Transmit ComSignalGroup(ISignalToPduMapping): SgGr1 - updateIndicationBitPosition is configured - ComGroupSignal(ISignalToPduMapping): GrSg1 -- ComSignalInitValue(ISignal.initValue) = GrSg1_Value_Init -- ComSignalDataInvalidValue (SwDataDefProps.invalidValue) = GrSg1_Value_Invalid - ComGroupSignal(ISignalToPduMapping): GrSg2 -- ComSignalInitValue(ISignal.initValue) = GrSg2_Value_Init -- ComSignalDataInvalidValue (SwDataDefProps.invalidValue) = GrSg2_Value_Invalid		
Summary	Aim: - Check that send signal group and invalidate signal group are taken into account in the periodic frame Sequence: 1) Action: Start ComIPduGroup - Result: I-PDU is sent out after Offset Time [SWS_Com_00222] - Result: Frames are sent out periodically (see ComTxModeTimePeriod) - Result: Group Signal values are initial value 2) Action: Update group signal - Result: Periodic Time is not changed - Result: SignalGroup UpdateBit is set to 1 for the first message after update/invalidation, only. [SWS_Com_00059][SWS_Com_00801][SWS_Com_00062] - Result: After successful transmission the UpdateBit is cleared. - Result: Group Signal values are changed for all new Tx frame occurrences 3) Action: Invalidate signal group - Result: Periodic Time is not changed		

	- Result: SignalGroup UpdateBit is set to 1, only in the first send after step 3. After it is 0. [SWS_Com_00059][SWS_Com_00286][SWS_Com_00062] - Result: All Group Signal values are the invalid values for all new Tx frame occurrences	
Needed Adaptation to other Releases		
Pre-conditions	Com stack is initialized, but ComIPduGroup are not running	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	***[SWC]*** Request ModeSwitch (call to BswMModeRequest port) to IPDU_ACTIVATED (start ComIPduGroup AT_209_IpduGroup)	***[LT<CAN>]*** AT_209_Ipdu is sent out after Offset Time Next AT_209_Ipdu are sent out every Period Time AT_209_GrSg1 value is AT_209_GrSg1_Value_Init AT_209_GrSg2 value is AT_209_GrSg2_Value_Init
Step 2	***[SWC]*** AT_209_SgGr1.AT_209_GrSg1=AT_209_GrSg1_Value_1 AT_209_SgGr1.AT_209_GrSg2=AT_209_GrSg2_Value_Init Call Rte_Write() for Port AT_209_SgGr1	***[LT<CAN>]*** AT_209_Ipdu Periodic Time is not changed AT_209_SgGr1 UpdateBit is set to 1 in the first send, after that, it is 0. AT_209_GrSg1 value is now AT_209_GrSg1_Value_1 AT_209_GrSg2 value is kept to AT_209_GrSg2 AT_209_GrSg2_Value_Init
Step 3	***[SWC]*** Invalidate signal group AT_209_SgGr1 by calling Rte_Invalidate() API	***[LT<CAN>]*** AT_209_Ipdu Periodic Time is not changed AT_209_SgGr1 UpdateBit is set to 1 in the first send, after that it is 0 AT_209_GrSg1 value is now AT_209_GrSg1_Value_Invalid AT_209_GrSg2 value is now AT_209_GrSg2_Value_Invalid
Post-conditions	Not Applicable	

4.3.3 [ATS_COMCAN_00210] Signal on User Request frame (DIRECT-N-TIMES)

Test Objective	Signal on User Request frame (DIRECT-N-TIMES)		
ID	ATS_COMCAN_00210	**AUTOSAR Releases**	4.2.1|4.2.2
Affected Modules	Com, PduR, CanIf, Can	**State**	reviewed

<table>
<tr><td>Trace to Requirement on Acceptance Test Document</td><td>ATR: ATR_ATR_00115
ATR: ATR_ATR_00116</td></tr>
<tr><td>Trace to SWS Item</td><td>COM: SWS_Com_00305
COM: SWS_Com_00767</td></tr>
<tr><td>Requirements / Reference to Test Environment</td><td>Use Case UC01.01</td></tr>
<tr><td>Configuration Parameters</td><td>ComIpdu(SignalIPdu): AT_210_Ipdu1 (Mapped on CAN Frame => CanTopology)
- ComIPduDirection(CommConnectorPort.communicationDirection) = SEND
- ComTxModeTrue (IPduTiming.TransmissionModeDeclaration.transmissionModeTrueTiming)
-- DIRECT (EventControlledTiming)
--- NumberOfRepetitions = 2
--- RepetitionPeriod = 100ms (This value can be changed by the test implementer)

ComSignal(ISignalToPduMapping): Sg1
- ComTransferProperty (transferProperty) = TRIGGERED
- ComSignalInitValue(ISignal.initValue) = Sg1_Value_Init

ComSignal(ISignalToPduMapping): Sg2
- ComTransferProperty (transferProperty) = TRIGGERED_WITHOUT_REPETITION
- ComSignalInitValue(ISignal.initValue) = Sg2_Value_Init
- ComSignalDataInvalidValue (SwDataDefProps.invalidValue) = Sg2_Value_Invalid</td></tr>
<tr><td>Summary</td><td>Aim:
- Check that send signal and invalidate signal are taken into account in the direct frame

Sequence:
1) Action: Start ComIPduGroup
- Result: I-PDU is not sent out
2) Action: Update signal 1 (triggered) [SWS_Com_00305]
- Result: I-PDU is sent two times (interval is Repetition Period)
- Result: Signal 1 value is changed for the 2 occurrences of the Tx frame
3) Action: Invalidate signal 2 (Triggered without repetition) [SWS_Com_00767]
- Result: I-PDU is sent only one time
- Result: Signal 2 value is the invalid value</td></tr>
<tr><td>Needed Adaptation to other Releases</td><td>Needed Adaptation for any Release earlier than [4.2.1]
<table><tr><td>Test Step</td><td>Expected Result</td></tr><tr><td>2</td><td>[LT<CAN>]
AT_210_Ipdu is sent two times (interval is Repetition Period)
First AT_210_Ipdu sent is immediate
Signal AT_210_Sg1 value is AT_210_Sg1_Value_1
Signal AT_210_Sg2 value is AT_210_Sg2_Value_Init</td></tr></table></td></tr>
<tr><td>Pre-conditions</td><td>Com stack is initialized, but ComIPduGroup are not running</td></tr>
<tr><td colspan="2">Main Test Execution</td></tr>
<tr><td colspan="2">Test Steps | Pass Criteria</td></tr>
</table>

Step 1	***[SWC]*** Request ModeSwitch (call to BswMModeRequest port) to IPDU_ACTIVATED (start ComIPduGroup AT_210_IpduGroup)	***[LT<CAN>]*** AT_210_Ipdu is not sent out
Step 2	***[SWC]*** Update signal AT_210_Sg1 by calling Rte_Write() API for Port AT_210_Sg1 (triggered) with AT_210_Sg1_Value_1	***[LT<CAN>]*** AT_210_Ipdu is sent three times (interval is Repetition Period) First AT_210_Ipdu sent is immediate Signal AT_210_Sg1 value is AT_210_Sg1_Value_1 Signal AT_210_Sg2 value is AT_210_Sg2_Value_Init
Step 3	***[SWC]*** Invalidate signal AT_210_Sg2 by calling Rte_Invalidate() API (triggered without repetition)	***[LT<CAN>]*** AT_210_Ipdu is sent only one time Signal AT_210_Sg1 value AT_210_Sg1_Value_1 Signal AT_210_Sg2 value AT_210_Sg2_Value_Invalid
Post-conditions	Not Applicable	

4.3.4 [ATS_COMCAN_00211] SignalGroup on User Request frame (DIRECT-N-TIMES)

Test Objective	SignalGroup on User Request frame (DIRECT-N-TIMES)		
ID	ATS_COMCAN_00211	**AUTOSAR Releases**	4.2.1|4.2.2
Affected Modules	Com, PduR, CanIf, Can	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00115 ATR: ATR_ATR_00116		
Trace to SWS Item	COM: SWS_Com_00741 COM: SWS_Com_00769		
Requirements / Reference to Test Environment	Use Case UC01.01		
Configuration Parameters	ComIpdu(SignalIPdu): AT_211_Ipdu1 (Mapped on CAN Frame => CanTopology) - ComIPduDirection(CommConnectorPort.communicationDirection) = SEND - ComTxModeTrue (IPduTiming.TransmissionModeDeclaration.transmissionModeTrueTiming) -- DIRECT (EventControlledTiming) --- NumberOfRepetitions = 2		

	--- RepetitionPeriod = 100ms (This value can be changed by the test implementer) ComSignalGroup(ISignalToPduMapping): SgGr1 - ComTransferProperty (transferProperty) = TRIGGERED - ComGroupSignal(ISignalToPduMapping): GrSg1/GrSg2 -- ComSignalInitValue(ISignal.initValue) = GrSg1_Value_Init ComSignalGroup(ISignalToPduMapping): SgGr2 - ComTransferProperty (transferProperty) = TRIGGERED_WITHOUT_REPETITION - ComGroupSignal(ISignalToPduMapping): GrSg3/GrSg4/GrSg5 -- ComSignalInitValue(ISignal.initValue) = GrSg2_Value_Init -- ComSignalDataInvalidValue (SwDataDefProps.invalidValue) = GrSg2_Value_Invalid
Summary	Aim: - Check that send signal group and invalidate group signal are taken into account in the direct frame Sequence: 1) Action: Start ComIPduGroup - Result: I-PDU is not sent out 2) Action: Send signal group 1 (triggered) without group signals Initial values - Result: I-PDU is sent two times (interval is Repetition Period) - Result: Group Signal values of Signal Group 1 are the initial values 3a) Action: Invalidate a group signal contained in signal group 2 (Triggered without repetition) [SWS_Com_00769] 3b) Action: Send signal group 2 (Triggered without repetition) [SWS_Com_00769] - Result: I-PDU is sent only one time - Result: All group signal of signal group 2 have invalid value
Needed Adaptation to other Releases	**Needed Adaptation for any Release earlier than [4.2.1]** Test Step: 2 — Expected Result: [LT<CAN>] AT_211_Ipdu is sent two times (interval is Repetition Period) First AT_211_Ipdu sent is immediate AT_211_GrSg1 value is AT_211_GrSg1_Value_Init AT_211_GrSg2 value is AT_211_GrSg2_Value_Init All Group Signals of AT_211_SgGr2 are set to Value_Init
Pre-conditions	Com stack is initialized, but ComIPduGroup are not running

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	***[SWC]*** Request ModeSwitch (call to BswMModeRequest port) to IPDU_ACTIVATED (start ComIPduGroup AT_211_IpduGroup)	***[LT<CAN>]*** AT_211_Ipdu is not sent out
Step 2	***[SWC]*** AT_211_SgGr1.AT_211_GrSg1=AT_211_GrSg1_Value_Init AT_211_SgGr1.AT_211_GrSg2=AT_211_GrSg2_Value_Init Call Rte_Write() for Port AT_211_SgGr1	***[LT<CAN>]*** AT_211_Ipdu is sent Three times (interval is Repetition Period) First AT_211_Ipdu sent is immediate AT_211_GrSg1 value is AT_211_GrSg1_Value_Init AT_211_GrSg2 value is

		AT_211_GrSg2_Value_Init All Group Signals of AT_211_SgGr2 are set to Value_Init
Step 3	***[SWC]*** Invalidate group signal AT_211_SgGr2 by calling Rte_Invalidate() API (triggered without repetition)	***[LT<CAN>]*** AT_211_Ipdu is sent only one time SgGr1 is unchanged: - AT_211_GrSg1 value is AT_211_GrSg1_Value_Init - AT_211_GrSg2 value is AT_211_GrSg2_Value_Init SgGr2 is Invalid: AT_211_GrSg3 value is AT_211_GrSg3_Value_Invalid AT_211_GrSg4 value is AT_211_GrSg4_Value_Invalid AT_211_GrSg5 value is AT_211_GrSg5_Value_Invalid
Post-conditions	Not Applicable	

4.3.5 [ATS_COMCAN_00213] Signal on Time Base and User Request frame (MIXED)

Test Objective	Signal on Time Base and User Request frame (MIXED)		
ID	ATS_COMCAN_00213	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	Com, PduR, CanIf, Can	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00115 ATR: ATR_ATR_00116		
Trace to SWS Item	COM: SWS_Com_00222 COM: SWS_Com_00734		
Requirements / Reference to Test Environment	Use Case UC01.01		
Configuration Parameters	ComIpdu(SignalIPdu): AT_213_Ipdu1 (Mapped on CAN Frame => CanTopology) - ComIPduDirection(CommConnectorPort.communicationDirection) = SEND - ComTxModeTrue (IPduTiming.TransmissionModeDeclaration.transmissionModeTrueTiming) -- MIXED (EventControlledTiming and CyclicTiming) --- NumberOfRepetitions = 1 --- timeOffset != timePeriod (Different from 0) ComSignal(ISignalToPduMapping): Sg1 - ComTransferProperty (transferProperty) = TRIGGERED_ON_CHANGE - ComSignalInitValue(ISignal.initValue) = Sg1_Value_Init		
Summary	Aim: - Check that send signal is taken into account in the mixed frame		

	Sequence: 1) Action: Start ComIPduGroup - Result: I-PDU is sent out after Offset Time [SWS_Com_00222] - Result: Next frames are sent out every Period Time - Result: Signal value is initial value 2) Action: Update signal (triggered on change) with a new value [SWS_Com_00734] - Result: an I-PDU sent out event is added between two I-PDU sent out period - Result: Signal value is the new value 3) Action: Update signal (triggered on change) with the same value [SWS_Com_00734] - Result: I-PDU send out period is not change (event I-PDU was not sent) - Result: Signal value is the same value
Needed Adaptation to other Releases	None.
Pre-conditions	Com stack is initialized, but ComIPduGroup are not running

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	***[SWC]*** Request ModeSwitch (call Rte_Switch associated to BswMMode port) to IPDU_ACTIVATED (start ComIPduGroupAT_213_IpduGroup)	***[LT<CAN>]*** AT_213_Ipdu is sent out after Offset Time. Next AT_213_Ipdu sent out are every Period Time AT_213_Sg1 value is AT_213_Sg1_Value_Init
Step 2	***[SWC]*** Update signal AT_213_Sg1 by calling Rte_Write() API for Port AT_213_Sg1 (triggered on change) with AT_213_Sg1_Value_1	***[LT<CAN>]*** AT_213_Ipdu sent out event is added between two AT_213_Ipdu sent out period Signal AT_213_Sg1 value is AT_213_Sg1_Value_1
Step 3	***[SWC]*** Update signal AT_213_Sg1 (call Rte_Write() API for Port AT_213_Sg1) (triggered on change) with the same value AT_213_Sg1_Value_1	***[LT<CAN>]*** AT_213_Ipdu send out period is not change (event I-PDU was not sent) Signal AT_213_Sg1 value is AT_213_Sg1_Value_1
Post-conditions	Not Applicable	

4.3.6 [ATS_COMCAN_00214] Signal Group on Time Base and User Request frame (MIXED)

Test Objective	Signal Group on Time Base and User Request frame (MIXED)		
ID	ATS_COMCAN_00214	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	Com, PduR, CanIf, Can	**State**	reviewed
Trace to	ATR: ATR_ATR_00115		

Requirement on Acceptance Test Document	ATR: ATR_ATR_00116
Trace to SWS Item	COM: SWS_Com_00222 COM: SWS_Com_00742 COM: SWS_Com_00743
Requirements / Reference to Test Environment	Use Case UC01.01
Configuration Parameters	ComIpdu(SignalIPdu): AT_214_Ipdu1 (Mapped on CAN Frame => CanTopology) - ComIPduDirection(CommConnectorPort.communicationDirection) = SEND - ComTxModeTrue (IPduTiming.TransmissionModeDeclaration.transmissionModeTrueTiming) -- MIXED (EventControlledTiming and CyclicTiming) --- NumberOfRepetitions = 0 --- timeOffset != timePeriod (Different from 0) ComSignalGroup(ISignalToPduMapping): SgGr1 - ComTransferProperty (transferProperty) = TRIGGERED_ON_CHANGE - ComGroupSignal(ISignalToPduMapping): GrSg1 -- ComSignalInitValue(ISignal.initValue) = GrSg1_Value_Init - ComGroupSignal(ISignalToPduMapping): GrSg2 -- ComSignalInitValue(ISignal.initValue) = GrSg2_Value_Init
Summary	Aim: - Check that send signal group is taken into account in the mixed frame Sequence: 1) Action: Start ComIPduGroup - Result: I-PDU is sent out after Offset Time [SWS_Com_00222] - Result: Next frames are sent out every Period Time - Result: Group Signal values are initial values 2a) Action: Update group signal (triggered on change) with the initial value 2b) Action: Send signal group (triggered on change) [SWS_Com_00743][SWS_Com_00742] - Result: I-PDU send out period is not changed (event I-PDU was not sent) - Result: Group Signal values are initial values 3a) Action: Update group signal (triggered on change) with a new value 3b) Action: Send signal group (triggered on change) [SWS_Com_00743][SWS_Com_00742] - Result: an I-PDU send out event is added between two I-PDU sent out period - Result: Group Signal value is the new value
Needed Adaptation to other Releases	
Pre-conditions	Com stack is initialized, but ComIPduGroup are not running

Main Test Execution

Test Steps		**Pass Criteria**
Step 1	***[SWC]*** Request ModeSwitch (call Rte_Switch associated to BswMMode port) to IPDU_ACTIVATED (start ComIPduGroup AT_214_IpduGroup)	***[LT<CAN>]*** AT_214_Ipdu is sent out after Offset Time Next AT_214_Ipdu sent out are every Period Time Group Signal AT_214_GrSg1 value is AT_214_GrSg1_Value_Init Group Signal AT_214_GrSg2 value

		is AT_214_GrSg2_Value_Init
Step 2	***[SWC]*** Call Rte_Write() API for Port AT_214_SgGr1 with AT_214_SgGr1 structure value {AT_214_GrSg1_Value_Init ; AT_214_GrSg2_Value_Init} (Rte will Send group signal AT_214_GrSg1 with AT_214_GrSg1_Value_Init Send group signal AT_214_GrSg2 with AT_214_GrSg2_Value_Init Send signal group AT_214_SgGr1 (triggered on change))	***[LT<CAN>]*** AT_214_Ipdu send out period is not changed (event I-PDU was not sent) AT_214_GrSg1 value is AT_214_GrSg1_Value_Init AT_214_GrSg1 value is AT_214_GrSg2_Value_Init
Step 3	***[SWC]*** Call Rte_Write() API for Port AT_214_SgGr1 with AT_214_SgGr1 structure value {AT_214_GrSg1_Value_1 ; AT_214_GrSg2_Value_Init} (Rte will Send group signal AT_214_GrSg1 with a new value AT_214_GrSg1_Value_1 Send group signal AT_214_GrSg2 with AT_214_GrSg2_Value_Init Send signal group AT_214_SgGr1 (triggered on change))	***[LT<CAN>]*** AT_214_Ipdu send out event is added between two AT_214_Ipdu sent out period Group signal AT_214_GrSg1 value is AT_214_GrSg1_Value_1 Group signal AT_214_GrSg2 value is AT_214_GrSg2_Value_Init
Post-conditions	Not Applicable	

4.3.7 [ATS_COMCAN_00715] Software filtering for PASS and FAIL of the received L-PDU when receive indication is given to CanIF

Test Objective	Software filtering for PASS and FAIL of the received L-PDU when receive indication is given to CanIF		
ID	ATS_COMCAN_00715	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CAN, CANIF	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANInterface: SWS_CANIF_00465 CANInterface: SWS_CANIF_00664 CANInterface: SWS_CANIF_00389 CANInterface: SWS_CANIF_00390 CANInterface: SWS_CANIF_00056 CANInterface: SWS_CANIF_00135 CANInterface: SWS_CANIF_00646 CANInterface: SWS_CANIF_00026 CANInterface: SWS_CANIF_00829 CANInterface: SWS_CANIF_00330 CANInterface: SWS_CANIF_00442		
Requirements /			

Reference to Test Environment	
Configuration Parameters	Example configuration: CanHandleType = BASIC CanIdValue = 0x2BC CanObjectType = RECEIVE CanIfRxPduDlc = 0x03 CanIfRxPduReadNotifyStatus = TRUE CanIfPublicReadRxPduDataApi = TRUE CanIfHrhSoftwareFilter = TRUE CanIfHrhRangeRxPduLowerCanId = 0x2BD CanIfHrhRangeRxPduUpperCanId = 0x2C6 CanIfRxPduUserRxIndicationUL = PDUR CanIfRxPduUserRxIndicationName = PduR_CanIfRxIndication CanIfPrivateDlcCheck = TRUE ComFirstTimeout = 400 ms ComTimeout = 100 ms Com_CbkRxTOut = App_Rte_Com_CbkTOut_TC_001 ComNotification = Rte_COMCbk_RteRx_Byte_0.
Summary	To verify the software filtering functionality for pass and fail of the received L-PDU when receive indication is given to CanIf, then checking of the DLC afterwards, followed by verifying the target configured upper layer which is to be called providing receive indication for the received L-PDU. **Test Description:** Using configuration parameters, the software filtering and DLC check shall be enabled. After this CAN-ID's will be sent from the TESTER to check the software filter algorithm and DLC check shall be processed afterwards. As this is an indirect testing i.e. the Com notification will be given to the test manager software component about the reception. Also, for the software filter algorithm to fail, it is only after the ComFirstTimeout the deadline monitoring will be starting and after ComTimeout the notification that the data has not been received will be given to test manager software component.
Needed Adaptation to other Releases	NA
Pre-conditions	DUT shall be initialized. ComM module shall be in FULL communication.

Main Test Execution

Test Steps		**Pass Criteria**
Step 1	[LT] Transmit a frame within the range having CAN-ID with data.	[SWC] Com notification for the configured signal shall be invoked in Test SWC
Step 2	[SWC] Requests Rte_Read for a signal	[SWC] RTE shall return E_OK and the received data shall be the same as seen on the bus

Post-conditions		

4.3.8 [ATS_COMCAN_00716] Software filtering for PASS and FAIL of the received L-PDU when receive indication is given to CanIF from where it is given to the configured CANTP

Test Objective	Software filtering for PASS and FAIL of the received L-PDU when receive indication is given to CanIF from where it is given to the configured CANTP		
ID	ATS_COMCAN_00716	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CAN,CANIF and CANTP	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANInterface: SWS_CANIF_00465 CANInterface: SWS_CANIF_00389 CANInterface: SWS_CANIF_00390 CANInterface: SWS_CANIF_00056 CANInterface: SWS_CANIF_00135 CANInterface: SWS_CANIF_00026 CANInterface: SWS_CANIF_00829 CANInterface: SWS_CANIF_00330 CANInterface: SWS_CANIF_00442		
Requirements / Reference to Test Environment			
Configuration Parameters	Example configuration: CanHandleType = BASIC CanIdValue = 0x2BC CanObjectType = RECEIVE CanIfRxPduDlc = 0x03 CanIfRxPduReadNotifyStatus = TRUE CanIfPublicReadRxPduDataApi = TRUE CanIfHrhSoftwareFilter = TRUE CanIfHrhRangeRxPduLowerCanId = 0x2BD CanIfHrhRangeRxPduUpperCanId = 0x2C6 CanIfRxPduUserRxIndicationUL = CAN_TP CanIfRxPduUserRxIndicationName = CanTp_RxIndication CanIfPrivateDlcCheck = TRUE ComFirstTimeout = 400 ms ComTimeout = 100 ms Com_CbkRxTOut = App_Rte_Com_CbkTOut_TC_003 ComNotification = Rte_COMCbk_RteRx_Byte_0.		
Summary	To verify the software filtering functionality for pass and fail of the received L-PDU when receive indication is given to CanIf. Then DLC and The target configured upper layer which is to be called providing receive indication service for the received L-PDU.		

	Test Description: Using configuration parameters, the software filtering and DLC check shall be enabled. After this CanIds will be sent from the TESTER to check the software filter algorithm and DLC check shall be processed afterwards. As this is an indirect testing i.e. the Com notification will be given to the test manager software component about the reception. Also, for the software filter algorithm to fail it is only after the ComFirstTimeout the deadline monitoring will be starting and after ComTimeout the notification that the data has not been received will be given to test manager software component.
Needed Adaptation to other Releases	NA
Pre-conditions	DUT shall be initialized. ComM module shall be in FULL communication

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] Transmit a frame within the range of DLC having Can-Id and Data (same number of bytes as mentioned in DLC) to the DUT from TESTER.	[SWC] Com notification for the configured signal shall be invoked in Test SWC.
Step 2	[SWC] Request Rte_Read for the signal	[SWC] Data shall be updated with the same content seen on the bus
Step 3	[LT] Transmit a frame outside the range of DLC having Can-Id with Data (less than the number specified in DLC) to the DUT from TESTER	[SWC] Com notification for the configured signal shall not be invoked in Test SWC.
Post-conditions		

4.3.9 [ATS_COMCAN_00717] DLC Check not configured for the received L-PDU

Test Objective	DLC Check not configured for the received L-PDU		
ID	ATS_COMCAN_00717	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CAN and CANIF	**State**	reviewed
Trace to			

Requirement on Acceptance Test Document	
Trace to SWS Item	CANInterface: SWS_CANIF_00830
Requirements / Reference to Test Environment	
Configuration Parameters	Example configuration: CanHandleType = BASIC CanIdType = STANDARD CanIdValue = 0x100 CanObjectType = RECEIVE CanFilterMask = CAN_FILTER_MASK_F CanIfHrhSoftwareFilter = TRUE CanIfHrhRangeRxPduLowerCanId = 0x100 CanIfHrhRangeRxPduRangeCanIdType = STANDARD CanIfHrhRangeRxPduUpperCanId = 0x10A CanIfRxPduReadNotifyStatus = TRUE CanIfRxPduUserRxIndicationUL = PDUR CanIfPrivateDlcCheck = FALSE ComIPduCallout = App_CanIf_TC_003
Summary	To check how CanIf behaves for a received L-PDU when DLC check is not configured for the received L-PDU. **Test Description:** Using Configuration Parameters, the software filtering shall be enabled and a received L-PDU with any DLC length is accepted when DLC check is not configured.
Needed Adaptation to other Releases	NA
Pre-conditions	DUT shall be initialized. ComM module shall be in FULL communication
Main Test Execution	
Test Steps	**Pass Criteria**
Step 1	[LT] Transmit frame with DLC value which is greater than the configured DLC value [SWC] Com notification for the configured signal shall be invoked in Test SWC.
Step 2	[SWC] Request Rte_Read for the signal [SWC] Signal value shall be the same as sent on the bus
Post-conditions	

4.3.10 [ATS_COMCAN_00718] Transmission request with Tx cancellation disabled

Test Objective	Transmission request with Tx cancellation disabled

ID	ATS_COMCAN_00718	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CAN and CANIF	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANInterface: SWS_CANIF_00281 CANInterface: SWS_CANIF_00070 CANInterface: SWS_CANIF_00183 CANInterface: SWS_CANIF_00383 CANInterface: SWS_CANIF_00058 CANInterface: SWS_CANIF_00438 CANInterface: SWS_CANIF_00542 CANInterface: SWS_CANIF_00439		
Requirements / Reference to Test Environment			
Configuration Parameters	Example configuration: CanIfBufferSize = 0x01 CanIfBufferHthRef = 0x00 CanIfHthCanCtrlIdRef = 0x01 CanIfHthIdSymRef = 0x00 CanHandleType = BASIC CanObjectId = HTH0-0 CanObjectType = TRANSMIT CanIfPublicTxBuffering = TRUE CanIfCtrlDrvTxCancellation = FALSE CanIfPublicReadTxPduNotifyStatusApi = TRUE CanIfTxPduReadNotifyStatus = TRUE CanIfTxPduCanId = 0x708,0x709,0x710 CanIfTxPduCanIdType = STANDARD_CAN CanIfTxPduDlc = 0x03 CanIfTxPduType = STATIC CanIfTxPduUserTxConfirmationUL = PDUR CanIfTxPduUserTxConfirmationName = PduR_CanIfTxConfirmation CanIfTxPduBufferRef = Reference to [CanIfBufferCfg]		
Summary	To check the behaviour of CanIf when test manager software component requests for the transmission of the higher and lower priority Can-Id, in case transmit cancellation is disabled and the upper layer (PduR) is configured for the transmit confirmation. **Test Description:** The test manager software component will request for the transmission of the higher and lower priority Can-Id and the frames will be observed on the bus. As this is an indirect testing for reading the transmitted data, the data observed on the bus will be checked for the most recent values.		
Needed Adaptation to other Releases	NA		
Pre-conditions	DUT shall be initialized. ComM module shall be in FULL communication.		

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Requests Rte_Write for a signal with value	[SWC] Rte_Write shall return RTE_E_OK.
Step 2	[SWC] Now simultaneously send a signal that belongs to a higher priority CAN ID and trigger Rte_Write for a signal with some value	[LT] Frames shall be observed with the value of first signal. After a particular interval Frame with second signal will be seen.
Post-conditions		

4.3.11 [ATS_COMCAN_00719] Transmission Request With Tx Cancellation Disabled And Having Upper Layer As CanTp

Test Objective	Transmission Request With Tx Cancellation Disabled And Having Upper Layer As CanTp		
ID	ATS_COMCAN_00719	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CAN and CANIF	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANInterface: SWS_CANIF_00281 CANInterface: SWS_CANIF_00070 CANInterface: SWS_CANIF_00383 CANInterface: SWS_CANIF_00058 CANInterface: SWS_CANIF_00320 CANInterface: SWS_CANIF_00439		
Requirements / Reference to Test Environment			
Configuration Parameters	Example configuration: CanIfBufferSize = 0x01 CanIfBufferHthRef = 0x00 CanIfHthCanCtrlIdRef = 0x01 CanIfHthIdSymRef = 0x00 CanHandleType = BASIC CanObjectId = HTH0 - 0 CanObjectType = TRANSMIT CanIfPublicTxBuffering = TRUE CanIfCtrlDrvTxCancellation = FALSE CanIfPublicReadTxPduNotifyStatusApi = TRUE CanIfTxPduReadNotifyStatus = TRUE CanIfTxPduCanId = 0x711,0x712,0x713 CanIfTxPduCanIdType = STANDARD_CAN CanIfTxPduDlc = 0x03		

	CanIfTxPduType = STATIC CanIfTxPduUserTxConfirmationUL = CAN_TP CanIfTxPduUserTxConfirmationName = CanTp_TxConfirmation CanIfTxPduBufferRef = Reference to [CanIfBufferCfg]	
Summary	To check the behaviour of CanIf when test manager software component requests for the transmission of the higher and lower priority Can-Id, in case transmit cancellation is disabled and the upper layer (CanTp) is configured for the transmit confirmation. **Test Description:** The test manager software component will request for the transmission of the higher and lower priority Can-Id and the frames will be observed on the bus. As this is an indirect testing for reading the transmitted data, the data observed on the bus will be checked for the most recent values.	
Needed Adaptation to other Releases	NA	
Pre-conditions	DUT shall be initialized. ComM module shall be in FULL communication	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Requests Rte_Write for a signal	[SWC] Rte_Write shall return RTE_E_OK.
Step 2	[SWC] Now simultaneously request Rte_Write for another signal that belongs to the higher priority CAN ID	[LT] Frames shall be observed with the value of first signal. After a perticular interval Frames for the second signal will be visible.
Post-conditions		

4.3.12 [ATS_COMCAN_00720] Transmission Request With Tx Cancellation Enabled

Test Objective	Transmission Request With Tx Cancellation Enabled		
ID	ATS_COMCAN_00720	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CAN and CANIF	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANInterface: SWS_CANIF_00281 CANInterface: SWS_CANIF_00070 CANInterface: SWS_CANIF_00183 CANInterface: SWS_CANIF_00383 CANInterface: SWS_CANIF_00521 CANInterface: SWS_CANIF_00439		
Requirements /			

Reference to Test Environment		
Configuration Parameters	Example configuration: CanIfBufferSize = 0x02 CanIfBufferHthRef = 0x01 CanIfHthCanCtrlIdRef = 0x01 CanHandleType = BASIC CanObjectId = HTH0-0 CanIfPublicTxBuffering = TRUE CanIfTxPduUserTxConfirmationUL = PDUR CanIfTxPduUserTxConfirmationName = PduR_CanIfTxConfirmation CanIfTxPduDlc = 3 CanIfCtrlDrvTxCancellation = TRUE	
Summary	To check the behaviour of CanIf when test manager software component requests for the transmission of the higher and lower priority Can-Id, in case transmit cancellation is enabled. **Test Description:** The test manager software component will request for the transmission of the higher and lower priority Can-Id and the frames will be observed on the bus. As this is an indirect testing for reading the transmitted data, the data observed on the bus will be checked for the most recent values. The transmit cancellation is enabled by configuration to avoid the inner priority inversion of the L-PDU transmitted on the CAN network.	
Needed Adaptation to other Releases	NA	
Pre-conditions	DUT shall be initialized. ComM module shall be in FULL communication.	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Requests Rte_Write for a signal	[SWC] Rte_Write shall return RTE_E_OK.
Step 2	[SWC] Now simultaneously request Rte_Write for another signal that belongs to the higher priority CAN ID	[SWC] Rte_Write shall return RTE_E_OK.
Step 3	-	[LT] Lower priority Frames shall be observed after the higher priority frames
Post-conditions		

5 RS_BRF_01648 - Large Data Type

5.1 General Test Objective and Approach

This Test Specification intends to cover the communication transfer of data sizes larger than the maximum transmission unit of the underlying bus as described in the AUTOSAR Feature [RS_BRF_01648].

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:

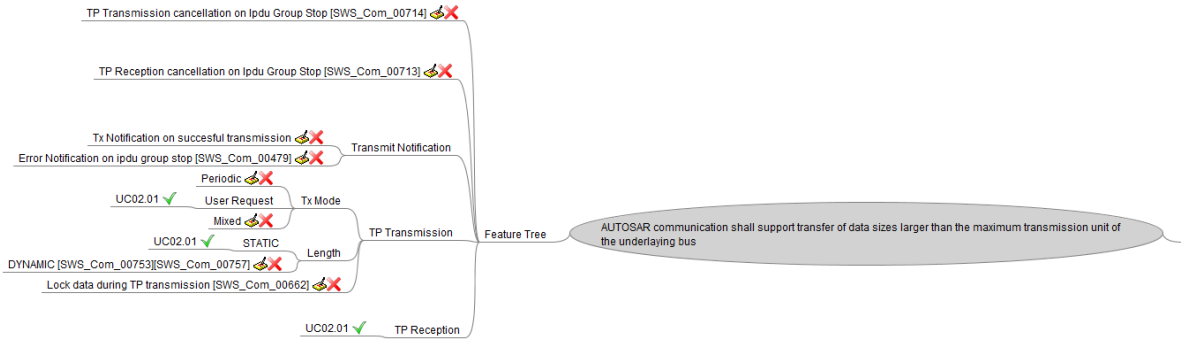

Fig D: Requirement on Large Data Type.

This specification gives the description of required tests environments (test bench, uses case, arxml 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, the different uses cases are created.

5.1.1.1.1 Use case 02.01: CAN Bus

For this use case, the aim is to test the large data type transfer on CAN bus:

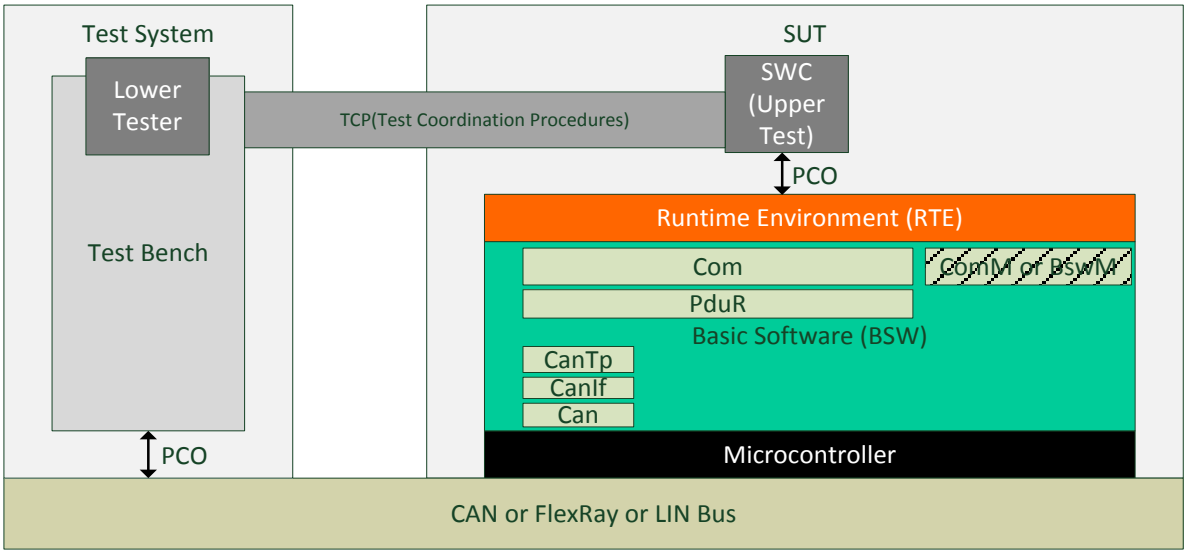

Fig E: Test System Architecture.

The test system architecture consists of Test Bench that executes only test sequencer and gives actions request through Test coordination Procedures to embedded SWC.

5.1.1.2 Specific Requirements

Not Applicable.

5.1.1.3 Test Coordination Requirements

Not Applicable.

5.1.2 Test Configuration

This section describes sets of requirements on configuration.
These sets are later referenced by test cases.
No configuration files are provided. They need to be developed when the test suite is implemented.

5.1.2.1 Required ECU Extract of System Description Files

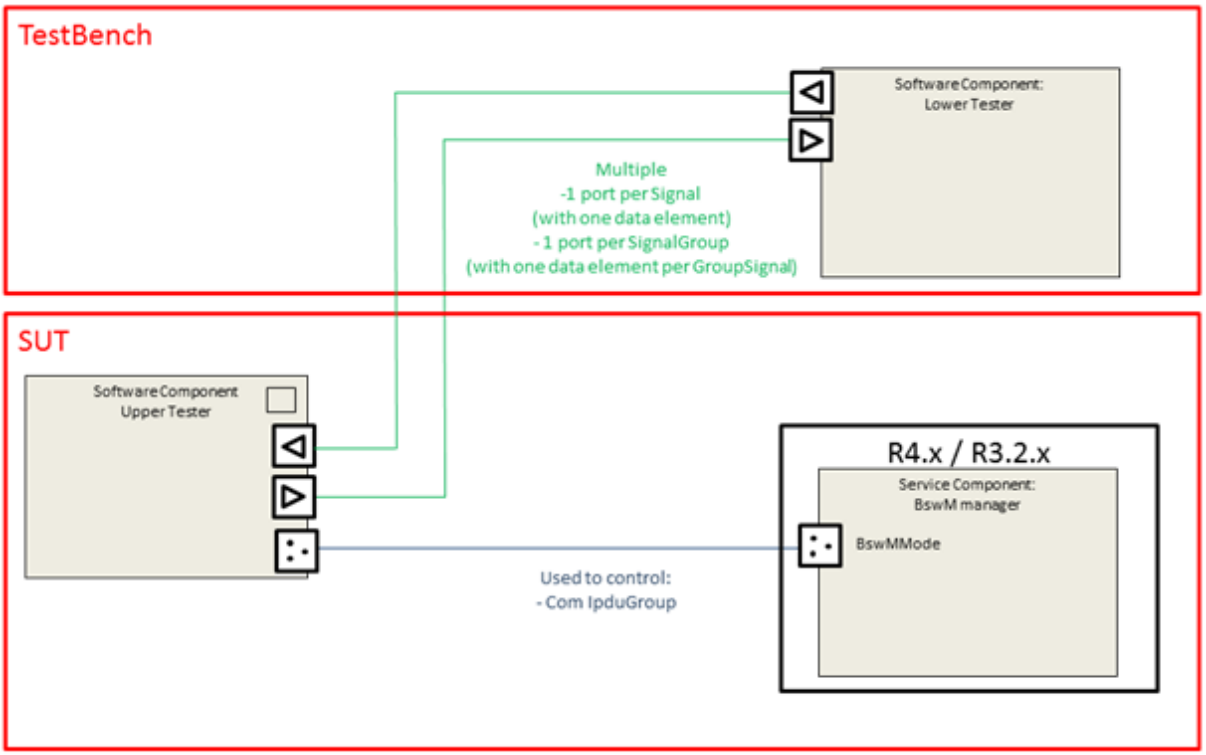

Fig F: SWC Overview on Large Data Type.

A Mode-Switch Interface IF_AT_SwC_ActionsBswM must be created. The SWC Upper Tester is the owner of this state machine and BswM read the state through BswMMode Port. BswM shall launch actions according to following table (check 5.3 Test Cases for details):

ModeDeclaration	BswM Actions
IPDU_ACTIVATED	OnEntry: -Start IpduGroup

For the Software Component point of view, for each test case, the communication interfaces are defined as follow:

Port name	Data element type	Dataelement	Mapping	Type
<TestCaseName>_<signalname>	uint8	<signalname>	<Signalname>	Signal

<TestCaseName>_<signalgroupname>	Struct
	{
	uint8: groupsignal1;
	…
	uint8: groupsignalx;
	}
		Groupsignal	Groupsignal1->
			<signal1name>
			Groupsignal2->
			<signal2name>
			<PortName>->
			<signalgroupname>
				Signal Group

Table 3:

Therefore ports and signals names change according to Test Case number, but the building rule is the same.

5.1.2.1.1 Use Case 02.01: CAN Bus

The communication database is depicted below:

IPduGroup	IPdu	SignalGroup	Signal	Tx ECU	Rx ECU
AT_239_IpduGroup	AT_239_Ipdu		AT_239_Sg1	SUT	TestBench
AT_276_IpduGroup	AT_276_Ipdu		AT_276_Sg1	TestBench	SUT

Table 4:

5.1.2.2 Required ECU Configuration Description Files

The section describes the common EcuC parameters between test cases that are required by the implementer of the test cases.

No specific configuration requirements for ECU Configuration files as they can be derived from EcuExtract.

5.1.2.3 Required Software Component Description Files

The section describes the SWC-D that are required by the implementer of the test cases.

Refer to Fig F.

5.1.2.4 Mandatory vs. Customizable Parts

Mandatory parameters are listed in Tests Cases (see 5.3 Test Cases).

Customizable parameters are (these values are test case independent):

- ComSignalType (ISignal.networkRepresentationProps.swBaseType), ComSignalLength (baseTypeSize) and ComBitSize (ISignal.length) => must be consistent to associated dataElement
- ComSignalInitValue (ISignal.initValue)
- PduLength (Pdu.length)
- ComBitPosition (ISignalToIPduMapping.startPosition) values => the location of these elements in the pdu
- CAN frames identifiers

NOTE: ComSignalInitValue and ComSignalDataInvalidValue are specific to test implementer and signal type.

5.1.3 Test Case Design

Not Applicable.

5.2 Re-usable Test Steps

Not Applicable.

5.3 Test Cases

5.3.1 [ATS_COMCAN_00239] Large Data TP transmission on CAN (>= 8 bytes)

Test Objective	Large Data TP transmission on CAN (>= 8 bytes)		
ID	ATS_COMCAN_00239	**AUTOSAR Releases**	4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	Com, PduR, CanTp, CanIf, Can	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00118		
Trace to SWS Item	COM: ECUC_Com_00761		
Requirements / Reference to Test Environment	Use Case UC02.01		
Configuration Parameters	ComIpdu(SignalIPdu): AT_239_Ipdu1 (large I-PDU) - length = 9 (large, greater than a Single Frame) - ComIPduType = TP(TpConfig.TpConnection) - ComIPduDirection(CommConnectorPort.communicationDirection) = SEND - ComTxModeTrue (IPduTiming.TransmissionModeDeclaration.transmissionModeTrueTiming) -- DIRECT(EventControlledTiming) --- NumberOfRepetitions = 1 ComSignal(ISignalToPduMapping): Sg1 - dataElement with queued swImplPolicy - DataSendCompletedEvent mapped on signal transmission (ComNotification is configured) - ComTransferProperty (transferProperty) = TRIGGERED PduRRoutingPath: - Routing path for ComIpdu with PduRSrcBswModuleRef = BswMod_Com - PduRDestPdu with PduRDestBswModuleRef = BswMod_CanTP		
Summary	Aim: - Check that Application layer can initiate a TP transmission greater than or equal to 8 bytes on CAN bus		
Needed Adaptation to other Releases	Configuration: [n/a] Test Steps: [n/a]	`Large data types and TP for regular COM is not possible in R3.x.` This test case is not applicable for R3.x.	
Pre-conditions	Com stack is initialized AT_239_IpduGroup is running		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	***[SWC]*** Call Rte_Send() for Port AT_239_Sg1 with AT_239_Sg1_Value_1 (Send Signal		***[LT<CAN>]*** First Frame is received Frame length is 8 byte, FF_DL is 9

	AT_239_Sg1 with AT_239_Sg1_Value_1 (this will initiate a TP transmission with 9 bytes))	bytes
Step 2	***[LT<CAN>]*** Send Flow Control Clear to Send (BlockSize = 0, STMin = 0). 3 bytes length if PADDING is not activated, 8 bytes otherwise.	***[LT<CAN>]*** One Consecutive Frame is received (4 bytes length if PADDING is not activated) AT_239_Sg1 value is AT_239_Sg1_Value_1
Post-conditions	Not Applicable	

5.3.2 [ATS_COMCAN_00276] Large Data TP reception on CAN (>= 8 bytes)

Test Objective	Large Data TP reception on CAN (>= 8 bytes)		
ID	ATS_COMCAN_00276	**AUTOSAR Releases**	4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	Com, PduR, CanTp, CanIf, Can	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00118		
Trace to SWS Item	COM: ECUC_Com_00761		
Requirements / Reference to Test Environment	Use Case UC02.01		
Configuration Parameters	ComIpdu(SignalIPdu): AT_276_Ipdu1 (large I-PDU) - length = 9 (large, greater than a Single Frame) - ComIPduType = TP(TpConfig.TpConnection) - ComIPduDirection(CommConnectorPort.communicationDirection) = RECEIVE ComSignal(ISignalToPduMapping): Sg1 - dataElement with queued swImplPolicy - DataReceivedEvent mapped on signal reception (ComNotification is configured) PduRRoutingPath: - Routing path for ComIpdu with PduRSrcBswModuleRef = BswMod_CanTP - PduRDestPdu with PduRDestBswModuleRef = BswMod_Com		
Summary	Aim: - Check that Application layer can receive a TP Data greater or equal than 8 bytes on CAN bus		
Needed Adaptation to other Releases	Configuration: [n/a] Test Steps: [n/a]	`Large data types and TP for regular COM is not possible in R3.x.` This test case is not applicable for R3.x.	
Pre-conditions	Com stack is initialized AT_276_IpduGroup is running		

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	***[LT<CAN>]*** Send Signal AT_276_Sg1 with AT_276_Sg1_Value_1 (this will initiate a TP transmission with 9 bytes)	***[LT<CAN>]*** First Frame is sent Frame length is 8 byte, FF_DL is 9 bytes
Step 2	***[LT<CAN>]*** Wait reception of Flow Control Clear to Send	***[LT<CAN>]*** Flow Control Clear to Send is received
Step 3	***[LT<CAN>]*** Send Consecutive Frame with last data bytes (4 bytes length if PADDING is not activated)	***[LT<CAN>]*** One Consecutive Frame is received (4 bytes length if PADDING is not activated)
Step 4	***[SWC]*** Wait DataReceivedEvent	***[SWC]*** DataReceivedEvent is activated
Step 5	***[SWC]*** Call Rte_Receive() for AT_276_Sg1	***[SWC]*** AT_276_Sg1 value is AT_276_Sg1_Value_1 Return Value of Rte_Receive is RTE_E_OK
Post-conditions	Not Applicable	

5.3.3 [ATS_COMCAN_00836] Transmission Of The Single Frames And Notification For Pdu Transfer Using Standard Addressing Format

Test Objective	Transmission Of The Single Frames And Notification For Pdu Transfer Using Standard Addressing Format		
ID	ATS_COMCAN_00836	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANTP	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANTransportLayer: SWS_CanTp_00177 CANTransportLayer: SWS_CanTp_00231 CANTransportLayer: SWS_CanTp_00204		
Requirements / Reference to Test Environment	none		
Configuration Parameters	CanTpTxAddressingFormat = CanTpStandard Sample configuration: CanTpNas = 0.1 sec CanTpNcs = - CanTpNbs = 0.1 sec		

	CanTpTxPaddingActivation = CanTpOn CanTpPaddingByte = 0xFF	
Summary	To transmit data having the data length less than or equals to 7 bytes from the test manager software component to the tester. As this is an indirect testing for the transmission confirmation. So the Com notification will be given to the test manager software component about the transmission of the signal.	
Needed Adaptation to other Releases		
Pre-conditions	ComM shall be in full communication	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Triggers Rte_Write for a signal with a value and SduId	[SWC] Rte_Write Returns E_OK
Step 2	[LT] Monitor the frame on the bus and validate the frame on tester	[LT] Frames shall be observed with the value on bus by the DUT
Step 3	-	[SWC] Transmission confirmation for the configured signal shall be invoked
Post-conditions	NONE	

5.3.4 [ATS_COMCAN_00837] Transmission Of The Single Frames And Notification For Pdu Transfer Using Extended Addressing Format

Test Objective	Transmission Of The Single Frames And Notification For Pdu Transfer Using Extended Addressing Format		
ID	ATS_COMCAN_00837	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANTP	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANTransportLayer: SWS_CanTp_00177 CANTransportLayer: SWS_CanTp_00231 CANTransportLayer: SWS_CanTp_00204		
Requirements / Reference to Test Environment	none		
Configuration Parameters	CanTpTxAddressingFormat = CanTpExtended Sample configuration: CanTpNTa = 0x34 CanTpNas = 0.1 sec CanTpNcs = - CanTpNbs = 0.1 sec		

	CanTpTxPaddingActivation = CanTpOn CanTpPaddingByte = 0xFF	
Summary	To transmit data having the data length less than or equals to 6 bytes from the test manager software component to the tester. As this is an indirect testing for the transmission confirmation. So the Com notification will be given to the test manager software component about the transmission of the signal.	
Needed Adaptation to other Releases		
Pre-conditions	ComM shall be in full communication	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Trigger Rte_Write API for a signal	[SWC] E_OK shall be returned.
Step 2	[LT] Monitor the frame on the bus and validate the frame on tester	[LT] Frames shall be observed with the value on bus by the DUT
Step 3	-	[SWC] Transmission confirmation for the configured signal shall be invoked
Post-conditions	NONE	

5.3.5 [ATS_COMCAN_00838] Transmission Of The Multi-PDU Frames And Notification For Pdu Transfer Using Standard Addressing Format

Test Objective	Transmission Of The Multi-PDU Frames And Notification For Pdu Transfer Using Standard Addressing Format		
ID	ATS_COMCAN_00838	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANTP	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANTransportLayer: SWS_CanTp_00177 CANTransportLayer: SWS_CanTp_00232 CANTransportLayer: SWS_CanTp_00204		
Requirements / Reference to Test Environment	none		
Configuration Parameters	CanTpTxAddressingFormat = CanTpStandard Sample configuration: CanTpNas = 0.1 sec CanTpNcs = - CanTpNbs = 0.1 sec CanTpTxPaddingActivation = CanTpOn CanTpPaddingByte = 0xFF		

Summary	To transmit data having the data length more than 7 bytes from the test manager software component to the tester. As this is an indirect testing for the transmission confirmation. So the Com notification will be given to the test manager software component about the transmission of the signal.	
Needed Adaptation to other Releases		
Pre-conditions	ComM shall be in full communication	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Triggers Rte_Write API for a signal Sg1 with a value and SduId	[SWC] E_OK shall be returned.
Step 2	[LT] Monitor the frame on the bus and validate the frame on tester	[LT] Frames shall be observed with the value on bus by the DUT Flow Control frame with value expected to be received in the DUT
Step 3	[LT] Monitor the frame on the bus and validate the frame on tester after the reception of flow control frame in the DUT	[LT] Frames shall be observed with the value on bus by the DUT
Step 4	-	[SWC] Transmission confirmation for the configured signal shall be invoked
Post-conditions	NONE	

5.3.6 [ATS_COMCAN_00839] Transmission Of The Multi-PDU Frames And Notification For Pdu Transfer Using Extended Addressing Format

Test Objective	Transmission Of The Multi-PDU Frames And Notification For Pdu Transfer Using Extended Addressing Format		
ID	ATS_COMCAN_00839	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANTP	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANTransportLayer: SWS_CanTp_00177 CANTransportLayer: SWS_CanTp_00232 CANTransportLayer: SWS_CanTp_00204		
Requirements / Reference to Test Environment	none		
Configuration	CanTpTxAddressingFormat = CanTpExtended		

Parameters	Sample configuration: CanTpNTa = 0x36 CanTpNas = 0.1 sec CanTpNcs = - CanTpNbs = 0.1 sec CanTpTxPaddingActivation = CanTpOn CanTpPaddingByte = 0xFF	
Summary	To transmit data having the data length more than 6 bytes from the test manager software component to the tester. As this is an indirect testing for the transmission confirmation. So the Com notification will be given to the test manager software component about the transmission of the signal.	
Needed Adaptation to other Releases		
Pre-conditions	ComM shall be in full communication	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Triggers Rte_Write for a signal Sg1 with value and SduId	[SWC] E_OK shall be returned.
Step 2	[LT] Monitor the frame on the bus and validate the frame on tester	[LT] Frames shall be observed with the value on bus by the DUT Flow Control frame with value is expected to be received in the DUT
Step 3	[LT] Monitor the frame on the bus and validate the frame on tester after the reception of flow control frame in the DUT	[LT] Frames shall be observed with the value on bus by the DUT
Step 4	-	[SWC] Transmission confirmation for the configured signal shall be invoked
Post-conditions	NONE	

5.3.7 [ATS_COMCAN_00840] Reception Of The Single Frames With SDU Padding Off

Test Objective	Reception Of The Single Frames With SDU Padding Off		
ID	ATS_COMCAN_00840	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANTP	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANTransportLayer: SWS_CanTp_00079 CANTransportLayer: SWS_CanTp_00084		

	CANTransportLayer: SWS_CanTp_00098 CANTransportLayer: SWS_CanTp_00116
Requirements / Reference to Test Environment	none
Configuration Parameters	CanTpRxAddressingFormat = CanTpStandard Sample configuration: CanTpNas = 0.1 sec CanTpNbr = 0.1 sec CanTpNcr = 1 sec CanTpRxDl = 3 CanTpRxPaddingActivation = CanTpOff CanTpPaddingByte = 0xFF
Summary	The data will be sent from the tester to the DUT to check the reception process and will be notified to the upper layer (PduR). As the CanTpRxPaddingActivation parameter is set to OFF the CanTp module shall check the frame data length. If a frame is received with an unexpected data length (check only for too short DLCs) the frame shall be ignored otherwise it is passed to the upper layer. As this is an indirect testing i.e. the Com notification will be given to the test manager software component about the reception and the data will be read by the RTE.
Needed Adaptation to other Releases	
Pre-conditions	ComM shall be in full communication
Main Test Execution	
Test Steps	**Pass Criteria**
Step 1	[LT] Send a frame with Can-Id and Data to the DUT from tester [SWC] Com notification for the configured signal shall be invoked
Step 2	[SWC] Trigger Rte_Read API to read the signal [SWC] Data shall be updated similar to what is observed on the bus
Post-conditions	None

5.3.8 [ATS_COMCAN_00841] Behaviour Of CanTp When A Frame With Unexpected Data Length Is Received With SDU Padding Off

Test Objective	Behaviour Of CanTp When A Frame With Unexpected Data Length Is Received With SDU Padding Off		
ID	ATS_COMCAN_00841	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANTP	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS	CANTransportLayer: SWS_CanTp_00079		

Item	CANTransportLayer: SWS_CanTp_00084 CANTransportLayer: SWS_CanTp_00098 CANTransportLayer: SWS_CanTp_00116	
Requirements / Reference to Test Environment	none	
Configuration Parameters	CanTpRxAddressingFormat = CanTpStandard Sample configuration: CanTpNas = 0.1 sec CanTpNbr = 0.1 sec CanTpNcr = 1 sec CanTpRxDl = 8 CanTpRxPaddingActivation = CanTpOff CanTpPaddingByte = 0xFF	
Summary	The data will be sent from the tester to the DUT to check the reception process and will be notified to the upper layer (PduR). As the CanTpRxPaddingActivation parameter is set to OFF the CanTp module shall check the frame data length. If a frame is received with an unexpected data length (check only for too short DLCs) the frame shall be ignored otherwise it is passed to the upper layer. As this is an indirect testing i.e. the Com notification will be given to the test manager software component about the reception and the data will be read by the RTE.	
Needed Adaptation to other Releases		
Pre-conditions	ComM shall be in full communication	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] Transmit a frame with canId and data with wrong/unexpected data length to the DUT from tester	[SWC] Com notification for the configured signal shall be not invoked
Post-conditions	NONE	

5.3.9 [ATS_COMCAN_00842] Reception Of The Multi-PDU Frames With SDU Padding On

Test Objective	Reception Of The Multi-PDU Frames With SDU Padding On		
ID	ATS_COMCAN_00842	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANTP	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANTransportLayer: SWS_CanTp_00079 CANTransportLayer: SWS_CanTp_00084 CANTransportLayer: SWS_CanTp_00116		

Requirements / Reference to Test Environment	none
Configuration Parameters	CanTpRxAddressingFormat = CanTpStandard Sample configuration: CanTpNas = 0.1 sec CanTpNbr = 0.1 sec CanTpNcr = 1 sec CanTpRxDl = 8 CanTpRxPaddingActivation = CanTpOn CanTpPaddingByte = 0xFF
Summary	The data will be sent from the tester to the DUT to check the reception process and will be notified to the upper layer (PduR). While receiving the frames from the tester if the CanTpRxPaddingActivation parameter is set to ON then a received N-PDU shorter than 8 bytes will be considered corrupt by CanTp. As this is an indirect testing i.e. the Com notification will be given to the test manager software component about the reception and the data will be read by the RTE.
Needed Adaptation to other Releases	
Pre-conditions	ComM shall be in full communication

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] Transmit a frame with Can-Id and Data with lesser than 8 bytes to the DUT from tester	[LT] Flow Control frame with value is expected to be given to the tester in response to the frame received in the DUT
Step 2	[LT] Monitor the frame on the bus and validate the frame on tester and flow control frames are expected from the tester	[LT] Frames shall be observed with the value on bus by the DUT
Step 3	[SWC] Triggers Rte_Read API to read the signal	[SWC] Data shall be updated.
Post-conditions	NONE	

5.3.10 [ATS_COMCAN_00843] Behaviour Of CanTp When Flow Control Frames Are Not Received After A Certain Amount Of Time During Transmission Of Multi-PDU Frames

Test Objective	Behaviour Of CanTp When Flow Control Frames Are Not Received After A Certain Amount Of Time During Transmission Of Multi-PDU Frames		
ID	ATS_COMCAN_00843	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANTP	**State**	reviewed

Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANTransportLayer: SWS_CanTp_00316		
Requirements / Reference to Test Environment	none		
Configuration Parameters	CanTpRxAddressingFormat = CanTpStandard Sample configuration: CanTpNas = 0.1 sec CanTpNbr = 0.1 sec CanTpNcr = 1 sec CanTpNbs = 1 sec CanTpRxPaddingActivation = CanTpOn CanTpPaddingByte = 0xFF		
Summary	To transmit data having the data length more than 7 bytes from the test manager software component to the tester. After the first frame is transmitted the wait for the flow control frame has to be deliberately extended beyond the timer N_Bs. When the timer expires CanTp will abort the current transmission.		
Needed Adaptation to other Releases			
Pre-conditions	ComM shall be in full communication		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[SWC] Triggers Explicit Inter Rte_Write API for a signal Sg1 with value and SduId		[SWC] E_OK shall be returned.
Step 2	[LT] Monitor the frame on the bus and validate the frame on tester and flow control frames are expected from the tester		[LT] Frames shall be observed with the value on bus by the DUT
Step 3	[LT] After the expiry of the Timer N_Bs monitor and validate the frame on tester		[LT] Frames shall not be observed on bus by the DUT
Post-conditions	NONE		

5.3.11 [ATS_COMCAN_00844] Behaviour Of CanTp When An Application Tries To Send Another Segmented Frame During The Time CanTp Waits For The FC Frame Fo An Ongoing Transmission

Test Objective	Behaviour Of CanTp When An Application Tries To Send Another Segmented Frame During The Time CanTp Waits For The FC Frame Fo An Ongoing Transmission		
ID	ATS_COMCAN_00844	**AUTOSAR**	4.0.3|4.2.1|4.2.2

		Releases	
Affected Modules	CANTP	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANTransportLayer: SWS_CanTp_00096		
Requirements / Reference to Test Environment	none		
Configuration Parameters	CanTpRxAddressingFormat = CanTpStandard Sample configuration: CanTpNas = 0.1 sec CanTpNbr = 0.1 sec CanTpNcr = 1 sec CanTpNbs = 1 sec CanTpRxPaddingActivation = CanTpOn CanTpPaddingByte = 0xFF		
Summary	To transmit data having the data length more than 7 bytes from the test manager software component to the tester. The second frame has to be deliberately introduced in-between the already initiated transmission of the first frame.		
Needed Adaptation to other Releases			
Pre-conditions	ComM shall be in full communication		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[SWC] Triggers Explicit Inter Rte_Write API for a signal Sg1 with value and SduId		[SWC] Rte_Write shall return RTE_E_OK.
Step 2	[LT] Monitor the frame on the bus and validate the frame on tester and flow control frames are expected from the tester		[LT] Frames shall be observed with the value on bus by the DUT
Step 3	[SWC] Before the expiry of timer N_Bs test riggers Explicit Inter Rte_Write API for a signal Sg2 with value and SduId		[SWC] Rte_Write shall return RTE_E_OK.
Step 4	[LT] Monitor the frame on the bus and validate the frame on tester and flow control frames are expected from the tester		[LT] Frames shall be observed with the value on bus by the DUT Flow Control frame is expected to be received in the DUT
Step 5	[LT] Monitor the frame on the bus and validate the frame on tester after the reception of flow control frame in the DUT		[LT] Frames shall be observed with the value on bus by the DUT

Step 6	-	[SWC] Transmission confirmation for the configured signal shall be invoked
Post-conditions	NONE	

5.3.12 [ATS_COMCAN_00845] Transmission Of A Large N-SDU

Test Objective	Transmission Of A Large N-SDU		
ID	ATS_COMCAN_00845	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANTP	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANTransportLayer: SWS_CanTp_00232		
Requirements / Reference to Test Environment	none		
Configuration Parameters	CanTpRxAddressingFormat = CanTpStandard Sample configuration: CanTpNas = 0.1 sec CanTpNbr = 0.1 sec CanTpNcr = 1 sec CanTpNbs = 1 sec CanTpNcs = 0.1 sec CanTpBs = 2 CanTpSTmin = 10 ms CanTpRxPaddingActivation = CanTpOn CanTpPaddingByte = 0xFF		
Summary	To transmit a large N-SDU having data length as 32 bytes from the test manager software component to the tester.		
Needed Adaptation to other Releases			
Pre-conditions	ComM shall be in full communication		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[SWC] Triggers Explicit Inter Rte_Write API to write a signal Sg1 with value and SduId		[SWC] E_OK shall be returned.
Step 2	[LT] Monitor the frame on the bus and validate the frame on tester and flow control frame is expected from the tester		[LT] Frames shall be observed with the value on bus by the DUT Flow Control frame with value is expected to be received in the DUT

Step 3	[LT] Monitor the frame on the bus and validate the frame on tester after the reception of flow control frame in the DUT	[LT] Frames shall be observed with the value on bus by the DUT
Step 4	[LT] Monitor the frame on the bus and validate the frame on tester for consecutive frames	[LT] Frames shall be observed with the value on bus by the DUT
Step 5	[LT] Monitor the frame on the bus and validate the frame on tester for the second flow control frame	[LT] Flow Control frame with value is expected for the second time as Block Size
Step 6	[LT] Monitor the frame on the bus and validate the frame on tester after the reception of second flow control frame in the DUT	[LT] Frames shall be observed with the value on bus by the DUT
Step 7	[LT] Monitor the frame on the bus and validate the frame on tester for second consecutive frame	[LT] Frames shall be observed with the value on bus by the DUT
Step 8	-	[SWC] Transmission confirmation for the configured signal shall be invoked
Post-conditions	NONE	

6 RS_BRF_01707 – CAN Bus Off handling

6.1 General Test Objective and Approach

The “CAN Bus-Off” feature is tested by setting the conditions which should trigger Bus-Off, transitions between internal states, Bus-Off recovery and then checking whether the transitions are performed correctly, following the right timing constraints.

6.1.1 Test System

6.1.1.1 Overview on Architecture

The basic test setup is depicted in the following figure:

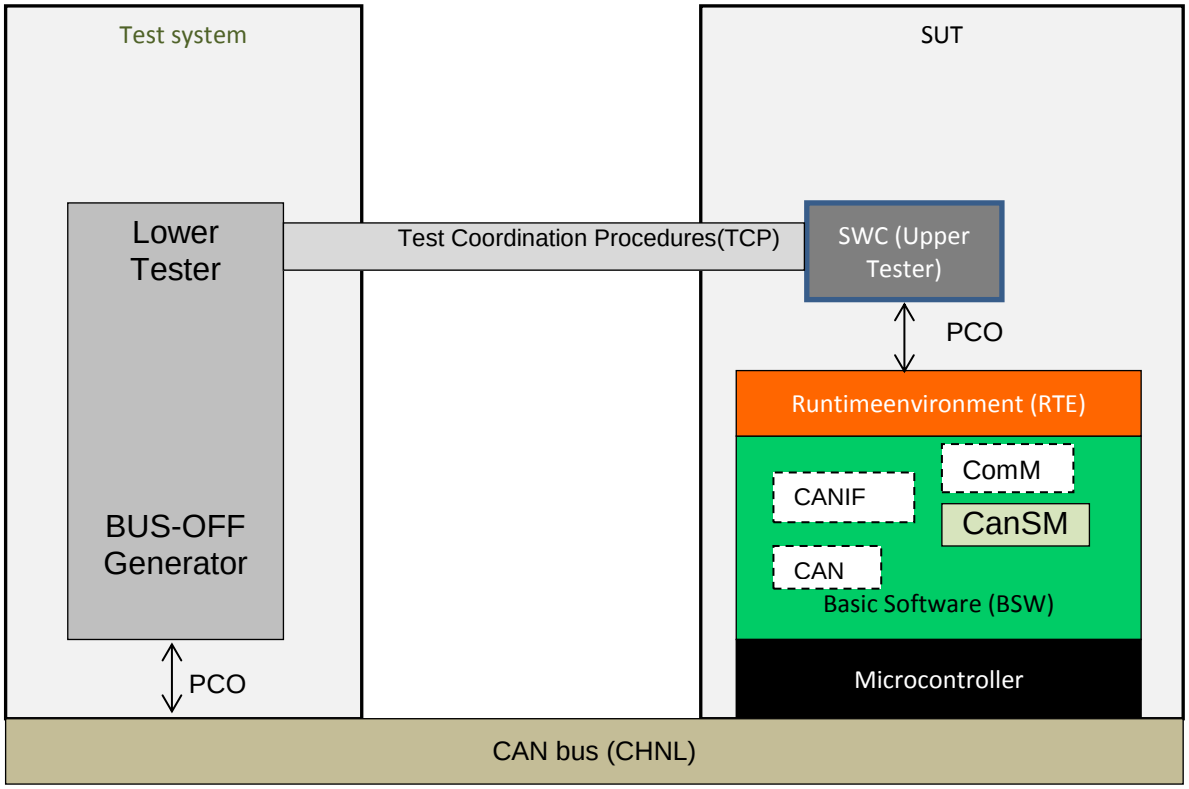

Fig G: Test System Architecture for BUS-OFF.

Figure1: Test Architecture

6.1.1.2 Specific Requirements

Lower Tester simulates Bus-Off with the help of a custom made tool which can generate Bus-Off in CHNL either by

a. Creating a short circuit and then by transmitting a message from the SUT
b. A disturbance in the CAN messages. Sent by the SUT

It is up to the test system designer/ implementer to decide how to generate Bus-Off.

Internal states of SUT shall be verified by checking the invocation of Rte_Mode API. A tester shall be connected to the test ECU CHNL for reading the logged Events.

An internal timer TMR-1 shall be used to verify the timing requirements.

6.1.1.3 Test Coordination Requirements

“Test Coordination Procedures” are needed to collect the test results of the SWC and the Remaining bus simulation at one central place, in order to derive the test verdict. It is up to the test system designer/implementer to define that “central place” and to design/implement the test coordination functionality.

6.1.2 Test Configuration

This section describes sets of requirements on configuration.
These sets are later referenced by test cases.
No configuration files are provided. They need to be developed when the test suite is implemented.

6.1.2.1 Required ECU Extract of System Description Files

The section describes the common EcuC parameters between test cases that are required by the implementer of the test cases.

No specific configuration requirements for ECU Configuration files as they can be derived from EcuExtract.

6.1.2.2 Required ECU Configuration Description Files

Each test case requires some configuration parameters of the SUT to be set with a specific value or within a given range. The test cases below are then provided along with 2 configuration sets (BUSOFF_PS001, BUSOFF_PS002). Each test case description includes a field mentioning the configuration sets which are applicable to that test case among these proposals.

6.1.2.2.1 Parameter Set [BUSOFF_PS001]

SUT configuration parameters	

Parameter name	Value
CanSMMainFunctionTimePeriod	10 msec
CanSMBorCounterL1ToL2	20
CanSMBorTimeL1	2000 msec
CanSMBorTimeL2	2000 msec
CanSMBorTimeTxEnsured	1500 msec
CanSMBorTxConfirmationPolling	false

Table 5:

6.1.2.2.2 Parameter Set [BUSOFF_PS002]

SUT configuration parameters	

Parameter name	Value
CanSMMainFunctionTimePeriod	10 msec
CanSMBorCounterL1ToL2	2
CanSMBorTimeL1	4000 msec
CanSMBorTimeL2	8000 msec

	CanSMBorTimeTxEnsured	1500 msec	
	CanSMBorTxConfirmationPolling	true	

Table 6:

6.1.2.3 Mandatory vs. Customizable Parts

Timing's (CanSMBorTimeL1, CanSMBorTimeL2 and CanSMBorTimeTxEnsured) and counter (CanSMBorCounterL1ToL2) values may be changed to the user's requirements or typical values.

6.1.3 Test Case Design

The test cases check that the SUT follows the state transitions defined in CanSM SWS with the required behavior. States and behavior can only be observed indirectly because of the ICC1 approach of acceptance testing. Thus the behavior on the bus, on the RTE and the diagnostic modules will be observed. State changes can be triggered only from outside of the SUT thus the bus has to be disturbed directly.
The test cases cover

- state change transitions triggered by Bus-Off generation and release
- behavior to the bus
- behavior to the RTE
- behavior to Events (Behaviour related to DEM Event)
- timing behaviour (Behaviour related to Configurable timing parameters – Ref sec 5.1.2.2)

6.2 Re-usable Test Steps

Creation of BUS-OFF scenario can be re-used in all the test cases.

6.3 Test Cases

6.3.1 [ATS_COMCAN_00269] Switching of communication mode during Bus-Off

Test Objective	Switching of communication mode during Bus-Off		
ID	ATS_COMCAN_00269	**AUTOSAR Releases**	4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CanSM, ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00101 ATR: ATR_ATR_00104		
Trace to SWS Item	CANStateManager: SWS_CanSM_00496 CANStateManager: SWS_CanSM_00498 CANStateManager: SWS_CanSM_00521 CANStateManager: SWS_CanSM_00522 COMManager: SWS_ComM_00091 COMManager: SWS_ComM_00778		

<table>
<tr><td>Requirements / Reference to Test Environment</td><td colspan="2">Test environment shall be able to generate a Bus-Off in the Test ECU</td></tr>
<tr><td>Configuration Parameters</td><td colspan="2">See [BUSOFF_PS001]</td></tr>
<tr><td>Summary</td><td colspan="2">Test whether CanSM is able to perform state transition based on Bus-Off notification and release.

Bus-Off mode switch and its release is observed by

• events retrieved through diagnostic interface
• the Rte_Mode API
• the CAN bus under test itself</td></tr>
<tr><td>Needed Adaptation to other Releases</td><td colspan="2">Needed Adaptation for Release [3.2.2]

<table><tr><td>Configuration: none

Test Steps: low</td><td>Same requirements on configuration

DEM events for bus off have fixed name in R3.2

Same test step sequence</td></tr></table></td></tr>
<tr><td>Pre-conditions</td><td colspan="2">All the communication channels are initialized</td></tr>
<tr><td colspan="3">Main Test Execution</td></tr>
<tr><td colspan="2">Test Steps</td><td>Pass Criteria</td></tr>
<tr><td>Step 1</td><td>[SWC]

Requests ComM to switch to COMM_FULL_COMMUNICATION using Rte_Call_comRequest_RequestComMode</td><td>[SWC]

Call to Rte_Call_comRequest_RequestComMode returns E_OK

Rte_Mode API is invoked with mode as RTE_MODE_ComMode_COMM_FULL_COMMUNICATION</td></tr>
<tr><td>Step 2</td><td>[SWC]

Trigger a communication sequence in the SUT- Example - ComIPduGroup start</td><td>[LT<CAN>]

Valid frames are observed in the Bus</td></tr>
<tr><td>Step 3</td><td>[CP]

WAIT till (CanSMBorTimeTxEnsured + 1s) time

WHILE WAITING, DO nothing</td><td>-</td></tr>
</table>

	Note: This delay is to provide enough time for the SUT to log the Events.	
Step 4	[SWC] Check the Events	[SWC] Event CANSM_E_BUS_OFF is logged with status as DEM_EVENT_STATUS_PASSED [SWS_CanSM_00496][SWS_CanSM_00498]
Step 5	[LT<CAN>] Generate Bus-Off in the Test ECU	-
Step 6	[CP] WAIT till (CanSMBorTimeL1 / 2) WHILE WAITING, DO nothing	-
Step 7	[SWC] Trigger a communication sequence in the SUT- Example - IPDU group start	[LT<CAN>] - No valid frames are observed in the Bus
Step 8	[SWC] Check the Events	[SWC] Event CANSM_E_BUS_OFF is logged with status as DEM_EVENT_STATUS_PREFAILED [SWS_CanSM_00522]
Step 9	[SWC] Check whether the SUT is in COMM_SILENT_COMMUNICATION	[SWC] Rte_Mode API is invoked with mode as RTE_MODE_ComMode_COMM_SILENT_COMMUNICATION [SWS_CanSM_00521][SWS_ComM_00091][SWS_ComM_00778]
Step 10	[LT<CAN>] End generation of Bus-Off in the Test ECU	-
Step 11	[CP] WAIT till (CanSMBorTimeL1 + 1.5s) WHILE WAITING, DO nothing	-
Step 12	[SWC] Check the Events	[SWC] Event CANSM_E_BUS_OFF is logged with status as DEM_EVENT_STATUS_PASSED [SWS_CanSM_00498]

Step 13	[SWC] Check whether the SUT is in COMM_FULL_COMMUNICATION	[SWC] Rte_Mode API is invoked with mode as RTE_MODE_ComMode_COMM_FULL_COMMUNICATION [SWS_ComM_00091][SWS_ComM_00778]
Step 14	[LT<CAN>] BUS-OFF should be re-covered	[LT<CAN>] Valid frames are observed in the Bus
Post-conditions	-	

6.3.2 [ATS_COMCAN_00270] Retaining FULL com in case of no BusOff with disabled CanSMBorTxConfirmationPolling

Test Objective	Retaining FULL com in case of no BusOff with disabled CanSMBorTxConfirmationPolling		
ID	ATS_COMCAN_00270	**AUTOSAR Releases**	4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CanSM	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00101		
Trace to SWS Item	CANStateManager: SWS_CanSM_00496 CANStateManager: SWS_CanSM_00498		
Requirements / Reference to Test Environment	-		
Configuration Parameters	See [BUSOFF_PS001]		
Summary	This test cases tests the ability of retaining in the FULL communication mode in case of no Bus-Off event when CanSMBorTxConfirmationPolling is disabled Test whether CanSM is able to enter and stay in FULL communication mode in case of no Bus-Off event. The CAN-Bus under test and Events are observed.		
Needed Adaptation to other Releases	**Needed Adaptation for Release [3.2.2]**		
	Configuration: none Test Steps: low	Same requirements on configuration DEM events for bus off have fixed name in R3.2 Same test step sequence	

Pre-conditions	All the communication channels are initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Requests ComM to switch to COMM_FULL_COMMUNICATION using Rte_Call_comRequest_RequestComMode	[SWC] Call to Rte_Call_comRequest_RequestComMode returns E_OK Rte_Mode API is invoked with mode as RTE_MODE_ComMode_COMM_FULL_COMMUNICATION
Step 2	[SWC] Trigger a communication sequence in the SUT- Example - ComIPduGroup start	[LT<CAN>] Valid frames are observed in the Bus Note: To ensure that the channel is in COMM_FULL_COMMUNICATION
Step 3	[CP] WAIT till (CanSMBorTimeTxEnsured + 1s) time WHILE WAITING, DO nothing	-
Step 4	[SWC] Check the Events	[SWC] Event CANSM_E_BUS_OFF is logged with status as DEM_EVENT_STATUS_PASSED [SWS_CanSM_00496][SWS_CanSM_00498]
Step 5	[CP] WAIT for 1 sec WHILE WAITING, DO: Check the invocation of Rte_Mode API	[SWC] Rte_Mode API is not invoked Note: This ensures that no mode switches are observed during the test cycle.
Post-conditions	-	

6.3.3 [ATS_COMCAN_00271] Retaining FULL com in case of no BusOff with enabled CanSMBorTxConfirmationPolling

Test Objective	Retaining FULL com in case of no BusOff with enabled CanSMBorTxConfirmationPolling		
ID	ATS_COMCAN_00271	**AUTOSA**	4.0.3|4.1.1|4.2.1|4.2.2

		R Releases	
Affected Modules	CanSM	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00101		
Trace to SWS Item	CANStateManager: SWS_CanSM_00497 CANStateManager: SWS_CanSM_00498		
Requirements / Reference to Test Environment	-		
Configuration Parameters	See [BUSOFF_PS002]		
Summary	This test case tests the ability of retaining in the FULL communication mode in case of no Bus-Off event when CanSMBorTxConfirmationPolling is enabled Test whether CanSM is able to enter and stay in FULL communication mode in case of no Bus-Off event. The CAN-Bus under test and Events are observed.		
Needed Adaptation to other Releases	**Needed Adaptation for Release [3.2.2]** Configuration: none Same requirements on configuration Test Steps: low DEM events for bus off have fixed name in R3.2 Same test step sequence		
Pre-conditions	All the communication channels in SUT are initialized		
Main Test Execution			
Test Steps		**Pass Criteria**	
Step 1	[SWC] Requests ComM to switch to COMM_FULL_COMMUNICATION using Rte_Call_comRequest_RequestComMode	[SWC] Call to Rte_Call_comRequest_RequestComMode returns E_OK Rte_Mode API is invoked with mode as RTE_MODE_ComMode_COMM_FULL_COMMUNICATION	
Step 2	[SWC] Trigger a communication sequence in the SUT- Example -	[LT<CAN>] Valid frames are observed in the Bus	

	ComIPduGroup start	Note: To ensure that the channel is in COMM_FULL_COMMUNICATION
Step 3	[SWC] Check the Events	[SWC] Event CANSM_E_BUS_OFF is logged with status as DEM_EVENT_STATUS_PASSED [SWS_CanSM_00497][SWS_CanSM_00498]
Step 4	[CP] WAIT for 1 sec WHILE WAITING, DO: Check the invocation of Rte_Mode API	[SWC] Rte_Mode API is not invoked Note: This ensures that no mode switches are observed during the test cycle.
Post-conditions	-	

6.3.4 [ATS_COMCAN_00272] Behavior of SUT during short recovery time

Test Objective	Behavior of SUT during short recovery time		
ID	ATS_COMCAN_00272	**AUTOSAR Releases**	4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CanSM	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00101 ATR: ATR_ATR_00104		
Trace to SWS Item	CANStateManager: SWS_CanSM_00375 CANStateManager: SWS_CanSM_00498 CANStateManager: SWS_CanSM_00522		
Requirements / Reference to Test Environment	Test environment shall be able to generate a Bus-Off in the Test ECU		
Configuration Parameters	See [BUSOFF_PS002]		
Summary	This test cases tests the behavior of the SUT during short recovery time. Test the behavior of SUT in: a. handling the application requests during the short recovery time		

	b. handling the received messages during the Bus-Off recovery cycle The test procedure generates Bus-Off, waits short Bus-Off recovery time and releases Bus-Off again. The correct behavior in the respective states is observed • on the bus (transmission of frames and acknowledgement of received frames) • on the RTE (application requests) • by events retrieved through diagnostic interface
Needed Adaptation to other Releases	**Needed Adaptation for Release [3.2.2]** Configuration: none — Same requirements on configuration Test Steps: low — DEM events for bus off have fixed name in R3.2 Same test step sequence
Pre-conditions	All the communication channels are initialized

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Requests ComM to switch to COMM_FULL_COMMUNICATION using Rte_Call_comRequest_RequestComMode	[SWC] Call to Rte_Call_comRequest_RequestComMode returns E_OK Rte_Mode API is invoked with mode as RTE_MODE_ComMode_COMM_FULL_COMMUNICATION
Step 2	[LT<CAN>] Generate Bus-Off in the Test ECU	[LT<CAN>] No valid frames are observed in the Bus
Step 3	[SWC] Check the Events	[SWC] Event CANSM_E_BUS_OFF is logged with status as DEM_EVENT_STATUS_PREFAILED [SWS_CanSM_00522]
Step 4	[SWC] Requests ComM to switch to COMM_NO_COMMUNICATION using Rte_Call_comRequest_RequestComMode	[SWC] Call to Rte_Call_comRequest_RequestComMode returns E_NOT_OK [SWS_CanSM_00375]
Step 5	[SWC] Requests ComM to switch to COMM_FULL_COMMUNICATION using Rte_Call_comRequest_RequestComMode	[SWC] Call to Rte_Call_comRequest_RequestComMode returns E_NOT_OK [SWS_CanSM_00375]

Step 6	[CP] WAIT till (CanSMBorTimeL1 / 2). WHILE WAITING, DO nothing	-
Step 7	[SWC] Trigger a communication sequence in the SUT- Example - ComIPduGroup start	[LT<CAN>] No valid frames are observed in the Bus
Step 8	[LT<CAN>] End generation of Bus-Off in the Test ECU	-
Step 9	[LT<CAN>] Send one message from the Tester to the test ECU	[LT<CAN>] Test ECU acknowledges the message and no error frames are observed in the Bus
Step 10	[CP] WAIT till (CanSMBorTimeL1 + 1s) WHILE WAITING, DO nothing	-
Step 11	[SWC] Check the Events	[SWC] Event CANSM_E_BUS_OFF is logged with status as DEM_EVENT_STATUS_PASSED [SWS_CanSM_00498]
Step 12	[SWC] Check whether the SUT is in COMM_FULL_COMMUNICATION	[SWC] Rte_Mode API is invoked with mode as RTE_MODE_ComMode_COMM_FULL_COMMUNICATION
Step 13	[LT<CAN>] BUS-OFF should be re-covered	[LT<CAN>] Valid frames are observed in the Bus
Post-conditions	-	

6.3.5 [ATS_COMCAN_00273] Behavior of SUT during long recovery time

Test Objective	Behavior of SUT during long recovery time		
ID	ATS_COMCAN_00273	**AUTOSAR Releases**	4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CanSM, ComM	**State**	reviewed
Trace to Require	ATR: ATR_ATR_00101 ATR: ATR_ATR_00104		

ment on Acceptance Test Document	
Trace to SWS Item	CANStateManager: SWS_CanSM_00376 CANStateManager: SWS_CanSM_00498 CANStateManager: SWS_CanSM_00522 CANStateManager: SWS_CanSM_00515 CANStateManager: SWS_CanSM_00518 COMManager: SWS_ComM_00091 COMManager: SWS_ComM_00778
Requirements / Reference to Test Environment	Test environment shall be able to generate a Bus-Off in the Test ECU
Configuration Parameters	See [BUSOFF_PS002]
Summary	This test case tests the behavior of the SUT during long recovery time. Test the behavior of SUT in: a. handling the application requests during the long recovery time b. handling the received messages during the Bus-Off recovery cycle The test procedure generates Bus-Off, waits short plus long Bus-Off recovery time and releases Bus-Off again. The correct behavior in the respective states is observed • on the bus (transmission of frames and acknowledgement of received frames) • on the RTE (application requests) • Events retrieved through diagnostic interface
Needed Adaptation to other Releases	**Needed Adaptation for Release [3.2.2]** Configuration: none — Same requirements on configuration Test Steps: low — DEM events for bus off have fixed name in R3.2 Same test step sequence
Pre-conditions	All the communication channels are initialized

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Requests ComM to switch to COMM_FULL_COMMUNICA	[SWC] Call to Rte_Call_comRequest_RequestComMode returns

	TION using Rte_Call_comRequest_RequestComMode	E_OK Rte_Mode API is invoked with mode as RTE_MODE_ComMode_COMM_FULL_COMMUNICATION
Step 2	[LT<CAN>] Generate Bus-Off in the Test ECU	-
Step 3	[CP] WAIT till (CanSMBorTimeL1 + 1s) WHILE WAITING, DO check frames on the bus	[LT<CAN>] No valid frames are observed in the Bus
Step 4	[SWC] Check the Events	[SWC] Event CANSM_E_BUS_OFF is logged with status as DEM_EVENT_STATUS_PREFAILED [SWS_CanSM_00522]
Step 5	[SWC] Requests ComM to switch to COMM_NO_COMMUNICATION using Rte_Call_comRequest_RequestComMode	[SWC] Call to Rte_Call_comRequest_RequestComMode returns E_NOT_OK [CANSM376]
Step 6	[SWC] Requests ComM to switch to COMM_FULL_COMMUNICATION using Rte_Call_comRequest_RequestComMode	[SWC] Call to Rte_Call_comRequest_RequestComMode returns E_NOT_OK [SWS_CanSM_00376]
Step 7	[CP] WAIT till (CanSMBorTimeL2 / 2) WHILE WAITING, DO nothing	-
Step 8	[SWC] Trigger a communication sequence in the SUT-Example -ComIPduGroup start	[LT<CAN>] No valid frames are observed in the Bus
Step 9	[LT<CAN>] End generation of Bus-Off in the Test ECU	-
Step 10	[LT<CAN>] Send one message from the	[LT<CAN>] Test ECU acknowledges the message and no error frames

	Tester to the test ECU	are observed in the Bus
Step 11	[CP] WAIT till (CanSMBorTimeL2 + 1s) WHILE WAITING, DO nothing	-
Step 12	[SWC] Check the Events	[SWC] Event CANSM_E_BUS_OFF is logged with status as DEM_EVENT_STATUS_PASSED [CANSM498]
Step 13	[SWC] Check whether the SUT is in COMM_FULL_COMMUNICATION	[SWC] Rte_Mode API is invoked with mode as RTE_MODE_ComMode_COMM_FULL_COMMUNICATION [SWS_CanSM_00515][SWS_CanSM_00518][SWS_ComM_00091][SWS_ComM_00778]
Step 14	[LT<CAN>] BUS-OFF should be re-covered	[LT<CAN>] Valid frames are observed in the Bus
Post-conditions	-	

6.3.6 [ATS_COMCAN_00274] Ensure the correct duration of Bus-Off recovery delay time

Test Objective	Ensure the correct duration of Bus-Off recovery delay time		
ID	ATS_COMCAN_00274	**AUTOSAR Releases**	4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CanSM, ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00101 ATR: ATR_ATR_00104		
Trace to SWS Item	CANStateManager: SWS_CanSM_00375 CANStateManager: SWS_CanSM_00376 CANStateManager: SWS_CanSM_00498 CANStateManager: SWS_CanSM_00521 CANStateManager: SWS_CanSM_00522 CANStateManager: SWS_CanSM_00514		

	CANStateManager: SWS_CanSM_00518 COMManager: SWS_ComM_00091 COMManager: SWS_ComM_00778
Requirements / Reference to Test Environment	Test environment shall be able to generate a Bus-Off in the Test ECU
Configuration Parameters	See [BUSOFF_PS002] ComIpdu(SignalIPdu): AT_274_Ipdu1 (Mapped on CAN Frame => CanTopology) - ComIPduDirection(CommConnectorPort.communicationDirection) = SEND - ComTxModeTrue (IPduTiming.TransmissionModeDeclaration.transmissionModeTrueTiming) -- PERIODIC (CyclicTiming) --- timePeriod > 0
Summary	This test case tests the correct duration of Bus-Off recovery delay time. Test whether the correct time is ensured for short and long recovery time. The test procedure generates bus off, releases Bus-Off and waits for valid messages. The time between Bus-Off generation and messages on the bus shall be the short respectively long bus off recovery time. RTE callbacks and Events are observed additionally.
Needed Adaptation to other Releases	**Needed Adaptation for Release [3.2.2]** Configuration: none Same requirements on configuration DEM events for bus off have fixed name in R3.2 Test Steps: low Same test step sequence
Pre-conditions	All the communication channels are initialized

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Requests ComM to switch to COMM_FULL_COMMUNICATION using Rte_Call_comRequest_RequestComMode	[SWC] Call to Rte_Call_comRequest_RequestComMode returns E_OK Rte_Mode API is invoked with mode as RTE_MODE_ComMode_COMM_FULL_COMMUNICATION
Step 2	[LT<CAN>] Generate Bus-Off in the Test ECU	-
Step 3	[LT<CAN>] Start the measurement timer TMR-1	-

Step 4	[CP] WAIT till (CanSMBorTimeL1 / 2) WHILE WAITING, DO check frames on the bus	[LT<CAN>] No valid frames are observed in the Bus
Step 5	[SWC] Check whether the SUT is in COMM_SILENT_COMMUNICATION	[SWC] Rte_Mode API is invoked with mode as RTE_MODE_ComMode_COMM_SILENT_COMMUNICATION [SWS_CanSM_00521][SWS_ComM_00091][SWS_ComM_00778]
Step 6	[LT<CAN>] End generation of Bus-Off in the Test ECU	-
Step 7	[LT<CAN>] BUS-OFF should be re-covered	[LT<CAN>] Valid frames are observed in the Bus
Step 8	[LT<CAN>] Stop the measurement timer TMR-1 with the first reception of I-PDUs	-
Step 9	[LT<CAN>] Check the calculated elapsed time from the timer TMR-1	[LT<CAN>] The Elapsed time is within the permissible range of CanSMBorTimeL1 [SWS_CanSM_00375][ECUC_CanSM_00128] Note: Time base of calculated Elapsed time should be in-line with CanSMMainFunctionTimePeriod
Step 10	[SWC] Check whether the SUT is in COMM_FULL_COMMUNICATION	[SWC] Rte_Mode API is invoked with mode as RTE_MODE_ComMode_COMM_FULL_COMMUNICATION [SWS_CanSM_00514][SWS_CanSM_00518][SWS_ComM_00091][SWS_ComM_00778]
Step 11	[SWC] Check the Events	[SWC] Event CANSM_E_BUS_OFF is logged with status as DEM_EVENT_STATUS_PASSED [SWS_CanSM_00498]
Step 12	[LT<CAN>]	-

	Generate Bus-Off in the Test ECU	
Step 13	[LT<CAN>] WAIT till CanSMBorTimeL1 WHILE WAITING, DO check frames on the bus	[LT<CAN>] No valid frames are observed in the Bus
Step 14	[SWC] Check the Events	[SWC] Event CANSM_E_BUS_OFF is logged with status as DEM_EVENT_STATUS_PREFAILED [SWS_CanSM_00522]
Step 15	[LT<CAN>] Start the measurement timer TMR-1	-
Step 16	[LT<CAN>] WAIT till (CanSMBorTimeL1 / 2) WHILE WAITING, DO check frames on the bus	[LT<CAN>] No valid frames are observed in the Bus
Step 17	[LT<CAN>] End generation of Bus-Off in the Test ECU	-
Step 18	[LT<CAN>] BUS-OFF should be re-covered	[LT<CAN>] Valid frames are observed in the Bus
Step 19	[LT<CAN>] Stop the measurement timer TMR-1 with the first reception of I-PDUs	-
Step 20	[LT<CAN>] Check the calculated elapsed time from the timer TMR-1	[LT<CAN>] The Elapsed time is within the permissible range of CanSMBorTimeL2 [SWS_CanSM_00376][ECUC_CanSM_00129] Note: Time base of calculated Elapsed time should be in-line with CanSMMainFunctionTimePeriod
Post-conditions	-	

7 RS_BRF_01649 - LdCom Large Data Transfer

7.1 General Test Objective and Approach

This Test Specification intends to cover communication transfer of array type signals, using LdCom as Interaction Layer on CAN bus as described in AUTOSAR Feature [RS_BRF_01649].

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

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

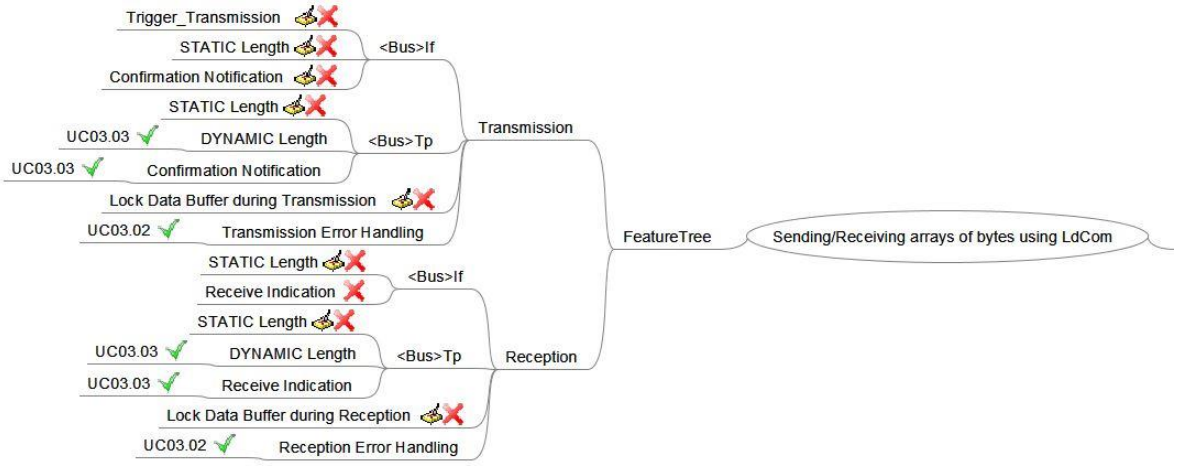

Fig H: Mindmap of features covered and not covered in the test cases

This specification gives the description of required test environment (Test Bench, Use cases, arxml files) and detailed test case for executing tests.

7.1.1 Test System

7.1.1.1 Overview on Architecture

In order to cover the required features/sub-features, the environment has been separated into several Use cases.

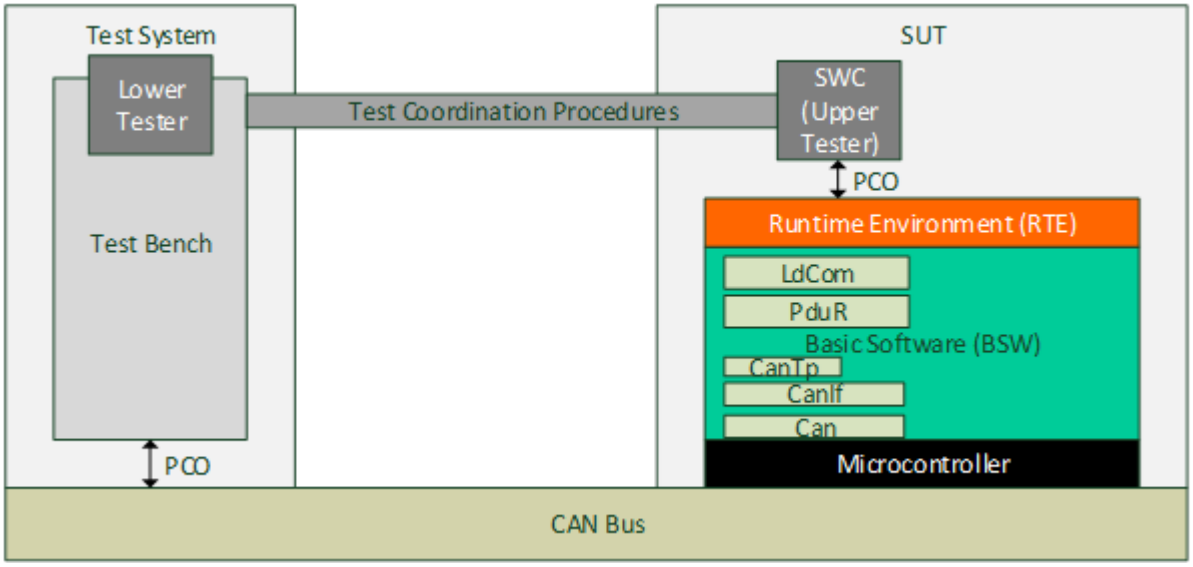

Fig I: Test System architecture

The Test System architecture consists of Test Bench that executes only test sequences and gives action request through test coordination procedures to embedded SWC.

7.1.1.1.1 Use case 03.02: Data Transfer of Arrays Signal of size more than <BUS> Capability

For this use case, the aim is to test data transfer features of LdCom, for array signal of length greater than underlying CAN bus capability.

7.1.1.1.2 Use case 03.03: Data Transfer of Dynamic Array size Signals

For this use case, the aim is to test data transfer features of LdCom, for dynamic array signal on CAN bus.

7.1.1.2 Specific Requirements

Not Applicable.

7.1.1.3 Test Coordination Requirements

Not Applicable.

7.1.2 Test Configuration

This section describes sets of requirements on configuration. These sets are later referenced by test cases. No configuration files are provided, they need to be developed when the test suite is implemented.

7.1.2.1 Required ECU Extract of System Description Files

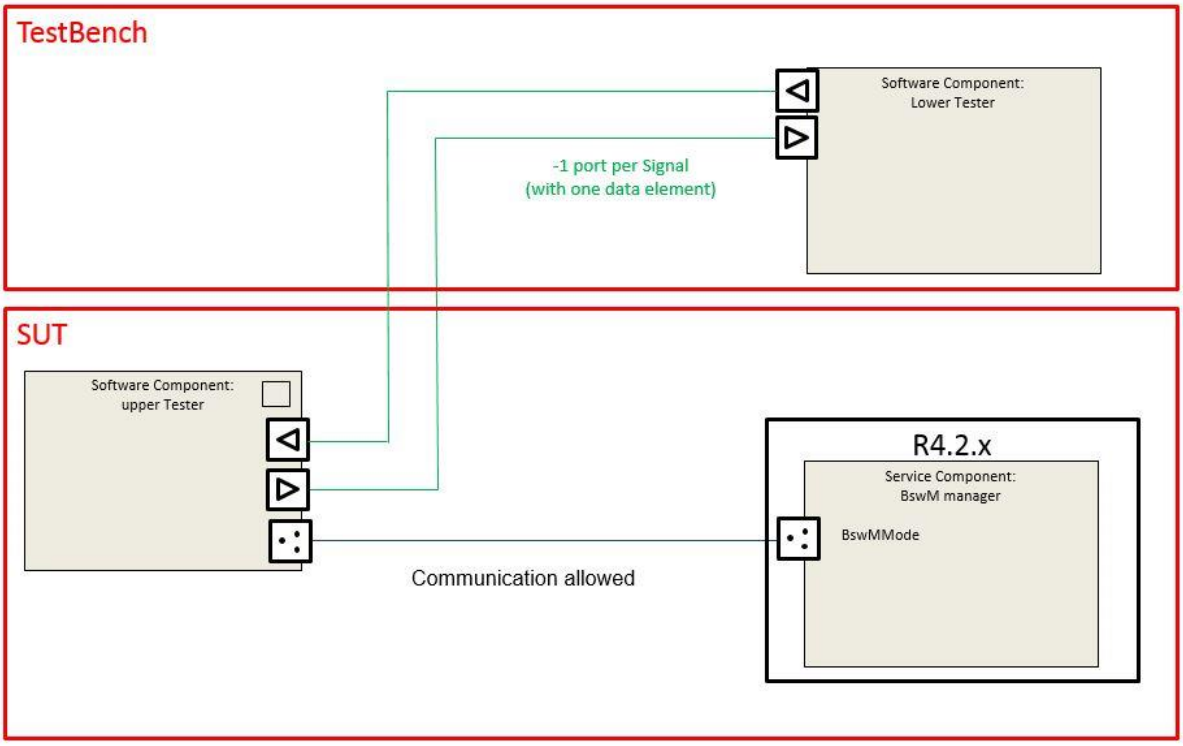

Fig J: Required SWC description

From Software Component point of view, for each test case, the communication interfaces are defined as follows:

Port name	Data element type	Data element	Mapping	Type
<TestCaseName>_<signalname>	UINT8_N	<signalname>	<signalname>	signal
<TestCaseName>_<signalname>	UINT8_DYN	<signalname>	<signalname>	signal

Table 7: SWC Interfaces used

Therefore ports and signals names change according to Test Case number, but the building rule is the same.

For API calls Rte_Write(), Rte_Send(), Rte_Read(),Rte_Receive(), Rte_Feedback() Refer [2] of Section 2.1 Input Documents.

For API calls Rte_LdComCbkTpRxIndication, Rte_LdComCbkTxConfirmation, Rte_LdComRxIndication, Rte_LdComCbkTpTxConfirmation Refer [1] of Section 2.1 Input Documents.

7.1.2.1.1 Use case 03.02: Data Transfer of Array Signal of size more than <BUS> Capability

The communication database is depicted below:

IPdu	Signal	Tx ECU	Rx ECU
AT_1483_IPdu	AT_1483_Sg1	SUT	TestBench
AT_1484_IPdu	AT_1484_Sg1	TestBench	SUT

Table 8: Communication Database

7.1.2.1.2 Use case 03.03: Data Transfer of Dynamic Array size Signals

The communication database is depicted below:

IPdu	Signal	Tx ECU	Rx ECU
AT_1479_IPdu	AT_1479_Sg1	SUT	TestBench
AT_1480_IPdu	AT_1480_Sg1	SUT	TestBench
AT_1481_IPdu	AT_1481_Sg1	TestBench	SUT
AT_1482_IPdu	AT_1482_Sg1	TestBench	SUT

Table 9: Communication Database

7.1.2.2 Required ECU Configuration Description Files

No specific configuration requirements for ECU Configuration files, as they can be derived from EcuExtract.

7.1.2.3 Required Software Component Description Files

No specific configuration requirements for Software Components.

7.1.2.4 Mandatory vs. Customizable Parts

Mandatory parameters are:

- ISignalToIPduMapping.startPosition => 0
- ISignalToIPduMapping.packingByteOrder =>Opaque
- ISignalToIPduMapping.transferProperty =>triggered/ triggeredWithoutRepetition
  See 7.3 Test Cases for further details.

Customizable parameters are (these values are test case independent):

- CANframes identifiers

7.1.3 Test Case Design

Not Applicable.

7.2 Re-usable Test Steps

Not Applicable.

7.3 Test Cases

7.3.1 [ATS_COMCAN_01479] LdCom Transmission using CanTp API for Dynamic Array Size with in Single Frame (<64 bytes for CAN-FD) and Notification for PDU transfer

Test Objective	LdCom Transmission using CanTp API for Dynamic Array Size with in Single Frame (<64 bytes for CAN-FD) and Notification for PDU transfer		
ID	ATS_COMCAN_01479	**AUTOSAR Releases**	4.2.1|4.2.2
Affected Modules	LdCom, PduR, CanTp, CanIf, Can	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00121 ATR: ATR_ATR_00127		
Trace to SWS Item	LargeDataCOM: SWS_LDCOM_00012 LargeDataCOM: SWS_LDCOM_00013		
Requirements / Reference to Test Environment	Use Case UC03.03		
Configuration Parameters	LdComIPdu(SignalIPdu): AT_1479_IPdu(normal I-PDU) -ISignal.length(for CAN 2.0) = 6(< 8 bytes) -ISignal.length(for CAN-FD) = 60(< 64 bytes) -LdComApiType = LDCOM_TP -LdComIPduDirection(CommConnectorPort.communicationDirection) = LDCOM_SEND -LdComTxCopyTxData = Rte_LdComCbkCopyTxData_Sg1 -LdComTxConfirmation = Rte_LdComCbkTpTxConfirmation_Sg1 LdComSignal(ISignalToPduMapping): Sg1 -dataElement with queued swImplPolicy -DataSendCompletedEvent mapped on TxConfirmation -SystemSignal.dynamicLength = true		
Summary	To check the LdCom transmission through CanTp for Single frame of dynamic array data type. Signal length shall be configured to a value less than 64 bytes for a CAN-FD frame or to a value less than 8 bytes for CAN 2.0 frame. As this is an indirect testing for LdCom transmission confirmation, notification is given to the software component of Upper Tester about transmission of the signal.		
Needed Adaptation to other Releases	n/a		
Pre-conditions	Com stack is initialized		
Main Test Execution			
Test Steps		**Pass Criteria**	
Step 1	*[SWC]* Trigger Rte_Send() for dynamic Signal AT_1479_Sg1 with signal length 6 bytes (This will initiate TP transmission).	*[SWC]* Rte_Send() shall return RTE_E_OK.	
Step 2	*[LT<CAN>]* Monitor and validate the frame on bus.	*[LT<CAN>]* Frame shall be observed with data	

		transmitted by SUT.
Step 3	-	*[SWC]* Rte_LdComCbkTpTxConfirmation API shall be invoked for the signal. DataSendCompleted event is activated.
Post-conditions	None	

7.3.2 [ATS_COMCAN_01480] LdCom Transmission using CanTp API for Dynamic Array Size with Multiple PDU (>64 bytes for CAN-FD) and Notification for PDU transfer

Test Objective	LdCom Transmission using CanTp API for Dynamic Array Size with Multiple PDU (>64 bytes for CAN-FD) and Notification for PDU transfer		
ID	ATS_COMCAN_01480	**AUTOSAR Releases**	4.2.1|4.2.2
Affected Modules	LdCom, PduR, CanTp, CanIf, Can	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00121 ATR: ATR_ATR_00127		
Trace to SWS Item	LargeDataCOM: SWS_LDCOM_00012 LargeDataCOM: SWS_LDCOM_00013		
Requirements / Reference to Test Environment	Use Case UC03.03		
Configuration Parameters	LdComIPdu(SignalIPdu): AT_1480_IPdu(large I-PDU) -ISignal.length(for CAN 2.0) = 30(>= 8 bytes) -ISignal.length(for CAN-FD) = 100(>= 64 bytes) -LdComApiType = LDCOM_TP -LdComIPduDirection(CommConnectorPort.communicationDirection) = LDCOM_SEND -LdComTxCopyTxData = Rte_LdComCbkCopyTxData_Sg1 -LdComTxConfirmation = Rte_LdComCbkTpTxConfirmation_Sg1 LdComSignal(ISignalToPduMapping): Sg1 -SystemSignal.dynamicLength = true -dataElement with queued swImplPolicy -DataSendCompletedEvent mapped on TxConfirmation		
Summary	-To check LdCom transmission for multiple Pdu through CanTP for a signal of type dynamic array. Signal length needs to be configured to a value greater than or equal to 64 bytes for a CAN-FD frame and to a value greater than or equal to 8 bytes for CAN 2.0 frame. As this is an indirect testing for transmission confirmation, notification will be given to software component of Upper Tester about transmission of the signal.		
Needed Adaptation to other Releases	n/a		
Pre-conditions	Com stack is initialized.		
Main Test Execution			

Test Steps		Pass Criteria
Step 1	*[SWC]* Trigger Rte_Send() for a dynamic signal AT_1480_Sg1 with signal length 30 bytes.	*[SWC]* Rte_Send() shall return RTE_E_OK.
Step 2	*[SWC]* First frame is observed on the bus. Wait for Flow Control frame with Flow Status ClearToSend.	*[SWC]* Flow Control with Flow Status ClearToSend is received.
Step 3	*[SWC]* Consecutive frames are sent by SWC until all the data has been transmitted on reception of Flow Control frame.	*[LT<CAN>]* Consecutive frames are received with data transmitted by SUT.
Step 4	-	*[SWC]* Rte_LdComCbkTpTxConfirmation API is invoked for the signal and DataSendCompleted event is activated for the same..
Post-conditions	None	

7.3.3 [ATS_COMCAN_01481] LdCom Reception using CanTp API for Dynamic Array Size with in Single Frame (<64 bytes for CAN-FD)

Test Objective	LdCom Reception using CanTp API for Dynamic Array Size with in Single Frame (<64 bytes for CAN-FD)		
ID	ATS_COMCAN_01481	**AUTOSAR Releases**	4.2.1|4.2.2
Affected Modules	LdCom, PduR, CanTp, CanIf, Can	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00121 ATR: ATR_ATR_00127		
Trace to SWS Item	LargeDataCOM: SWS_LDCOM_00014 LargeDataCOM: SWS_LDCOM_00015 LargeDataCOM: SWS_LDCOM_00016		
Requirements / Reference to Test Environment	Use Case UC03.03		
Configuration Parameters	LdComIPdu(SignalIPdu): AT_1481_IPdu(normal I-PDU) -ISignal.length(for CAN 2.0) = 6(< 8 bytes) -ISignal.length(for CAN-FD) = 60(< 64 bytes) -LdComApiType = LDCOM_TP -LdComIPduDirection(CommConnectorPort.communicationDirection) = LDCOM_RECEIVE -LdComRxIndication = Rte_LdComCbkTpRxIndication_Sg1 -LdComRxStartOfReception = Rte_LdComCbkStartOfReception_Sg1		

	-LdComRxCopyRxData = Rte_LdComCbkCopyRxData_Sg1 LdComSignal(ISignalToPduMapping): Sg1 -SystemSignal.dynamicLength = true -dataElement with queued swImplPolicy -DataReceivedEvent mapped on RxIndication
Summary	- To check that application can receive LdCom data through TP for a dynamic signal of length less than 8 bytes in case of CAN 2.0 frame or of length lesser than 64 bytes in case of CAN-FD frame.
Needed Adaptation to other Releases	n/a
Pre-conditions	Com stack is initialized.
Main Test Execution	

Test Steps		**Pass Criteria**
Step 1	*[LT<CAN>]* Send the dynamic signal AT_1481_Sg1 with value AT_1481_Sg1_Value_1 and signal length 6 bytes.	*[SWC]* Rte_LdComCbkTpRxIndication API for the signal is invoked. DataReceivedEvent is activated.
Step 2	*[SWC]* Call Rte_Receive() for AT_1481_Sg1.	*[SWC]* Return value of Rte_Receive() is RTE_E_OK AT_1481_Sg1_value is AT_1481_Sg1_Value_1.
Post-conditions	NONE	

7.3.4 [ATS_COMCAN_01482] LdCom Reception using CanTp API for Dynamic Array Size with Multiple PDU (>64 bytes for CAN-FD)

Test Objective	LdCom Reception using CanTp API for Dynamic Array Size with Multiple PDU (>64 bytes for CAN-FD)		
ID	ATS_COMCAN_01482	**AUTOSAR Releases**	4.2.1|4.2.2
Affected Modules	LdCom, PduR, CanTp, CanIf, Can	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00121 ATR: ATR_ATR_00127		
Trace to SWS Item	LargeDataCOM: SWS_LDCOM_00015 LargeDataCOM: SWS_LDCOM_00016 LargeDataCOM: SWS_LDCOM_00017		
Requirements / Reference to Test Environment	Use Case UC03.03		
Configuration Parameters	LdComIPdu(SignalIPdu): AT_1482_IPdu(large I-PDU) -ISignal.length(for CAN 2.0) = 30(>= 8 bytes) -ISignal.length(for CAN-FD) = 100(>= 64 bytes)		

	-LdComApiType = LDCOM_TP -LdComIPduDirection(CommConnectorPort.communicationDirection) = LDCOM_RECEIVE -LdComRxIndication = Rte_LdComCbkTpRxIndication_Sg1 -LdComRxStartOfReception = Rte_LdComCbkStartOfReception_Sg1 -LdComRxCopyRxData = Rte_LdComCbkCopyRxData_Sg1 LdComSignal(ISignalToPduMapping): Sg1 -SystemSignal.dynamicLength = true -dataElement with queued swImplPolicy -DataReceivedEvent mapped on RxIndication
Summary	-To check that application can receive LdCom data through CanTP for a dynamic signal of length greater than or equal to 8 bytes for CAN 2.0 frame and length greater than or equal to 64 bytes for CAN-FD.
Needed Adaptation to other Releases	n/a
Pre-conditions	Com stack is initialized.
Main Test Execution	
Test Steps	**Pass Criteria**
Step 1	*[LT<CAN>]* Send the frame with dynamic signal AT_1482_Sg1 with value AT_1482_Sg1_Value_1 and signal length 30 bytes.
	[SWC] First frame is observed on the bus with Frame length 8 bytes and FF_DL 30 bytes. Flow Control with Flow Status ClearToSend is sent.
Step 2	*[LT<CAN>]* Consecutive frames are sent by LT until all the data has been transmitted on reception of Flow Control frame.
	[SWC] DataReceivedEvent is activated on successful reception of all Consecutive frames.
Step 3	*[SWC]* Call Rte_Receive() to read AT_1482_Sg1.
	[SWC] Rte_Receive() returns RTE_E_OK. AT_1482_Sg1_value is AT_1482_Sg1_Value_1.
Post-conditions	NONE

7.3.5 [ATS_COMCAN_01483] LdCom Behavior in case of CanTp communication failures during multiple PDU Transmission

Test Objective	LdCom Behavior in case of CanTp communication failures during multiple PDU Transmission		
ID	ATS_COMCAN_01483	**AUTOSAR Releases**	4.2.1|4.2.2
Affected Modules	RTE, LdCom	**State**	reviewed
Trace to Requirement	ATR: ATR_ATR_00127 ATR: ATR_ATR_00128		

on Acceptance Test Document		
Trace to SWS Item	RTE: SWS_Rte_01380	
Requirements / Reference to Test Environment	Use Case UC03.02	
Configuration Parameters	LdComIPdu(SignalIPdu): AT_1483_IPdu(large I-PDU) -LdComApiType = LDCOM_TP -LdComIPduDirection(CommConnectorPort.communicationDirection) = LDCOM_SEND -LdComTxCopyTxData = Rte_LdComCbkCopyTxData_Sg1 -LdComTxConfirmation = Rte_LdComCbkTpTxConfirmation_Sg1 LdComSignal(ISignalToPduMapping): Sg1 -ISignal.length > Size of <BUS> capability	
Summary	- LdCom shall return an appropriate error result to RTE on CanTp transmission failure. Note: Errors may occur due to OverFlow, N_Bs timeout, N_As Timeout, N_Cs timeout and so on. A Rte_Feedback call to RTE shall return transmission failure status to Application.	
Needed Adaptation to other Releases	n/a	
Pre-conditions	Com stack is initialized.	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	*[SWC]* Trigger Rte_Write() for signal AT_1483_Sg1 with signal length greater than single frame (This initiates TP transmission).	*[SWC]* Rte_Write() shall return RTE_E_OK.
Step 2	*[LT<CAN>]* Monitor and validate the frame on bus.	*[LT<CAN>]* Frames shall be observed with value on bus by SUT.
Step 3	*[SWC]* TP transmission is terminated because of an error. Note: Errors are listed in the Summary of this TestCase.	*[SWC]* Rte_LdComCbkTpTxConfirmation API for the signal is invoked with result E_NOT_OK.
Step 4	*[SWC]* Call Rte_Feedback() for AT_1483_Sg1.	*[SWC]* Rte_Feedback() shall return RTE_E_NO_DATA.
Post-conditions	None	

7.3.6 [ATS_COMCAN_01484] LdCom Behavior in case of CanTp communication failure during multiple PDU Reception

Test Objective	LdCom Behavior in case of CanTp communication failure during multiple PDU Reception		
ID	ATS_COMCAN_01484	**AUTOSAR Releases**	4.2.1|4.2.2
Affected Modules	RTE, LdCom	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00127 ATR: ATR_ATR_00128		
Trace to SWS Item	RTE: SWS_Rte_01387 RTE: SWS_Rte_01388		
Requirements / Reference to Test Environment	Use Case UC03.02		
Configuration Parameters	LdComIPdu(SignalIPdu): AT_1484_IPdu(large I-PDU) -LdComApiType = LDCOM_TP -LdComIPduDirection(CommConnectorPort.communicationDirection) = LDCOM_RECEIVE -LdComRxIndication = Rte_LdComCbkTpRxIndication_Sg1 -LdComRxStartOfReception = Rte_LdComCbkStartOfReception_Sg1 -LdComRxCopyRxData = Rte_LdComCbkCopyRxData_Sg1 LdComSignal(ISignalToPduMapping): Sg1 -SignalLength(baseTypeSize) > Size of <BUS> capability		
Summary	- LdCom shall return appropriate error result to RTE when CanTp reception fails during Multi PDU reception. Note: Error may occur due to Buffer unavailability, N_Ar timeout, N_Cr timeout and so on. Reception Failure is notified to Application on read request.		
Needed Adaptation to other Releases	n/a		
Pre-conditions	Com stack is initialized		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	*[LT<CAN>]* Send the signal AT_1484_Sg1 of length greater than single frame.		*[LT<CAN>]* Frames is observed on the bus with Frame length 8 bytes and FF_DL greater than single frame length.
Step 2	*[SWC]* TP reception fails because of an error. Note: Errors are listed in the Summary of this TestCase.		*[SWC]* Rte_LdComCbkTpRxIndication API for the signal with E_NOT_OK.
Step 3	*[SWC]*		*[SWC]*

	Call Rte_Read() for AT_1484_Sg1.	AT_1484_Sg1 Signal value is set to invalid value.
Post-conditions	NONE	