Document Title	Acceptance Test Specification of Communication on LIN bus
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	667
Document Classification	Auxiliary
Document Status	Final
Part of AUTOSAR Product	Acceptance Tests for Classic Platform
Part of Product Release	1.1.0

Document Change History		
Release	**Changed by**	**Change Description**
1.1.0	AUTOSAR Release Management	• Checked and adapted to Classic Platform Release 4.2.1 (NumberOfRepetitions set to 0 in ATS_COMLIN_00241) • Formalization of point of control and observation • Added test cases for LIN Transport Protocol • Formal changes
1.0.0	AUTOSAR Release Management	Initial release, including test suites on • RS_BRF_01591 – Data Transfer • RS_BRF_01648 – Large Data Type

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
ECU	Electronic Control Unit
LIN	Local Interconnect Network
LT	Lower Tester
PCO	Point of Control and Observation
PDU	Protocol Data Unit
Rx	Reception
SUT	System Under Test
SWC	Software Component
TCP	Test Coordination Procedures
Tx	Transmission
UT	Upper Tester

2 Scope

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

3 RS_BRF_01592 - Data Transfer

3.1 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:

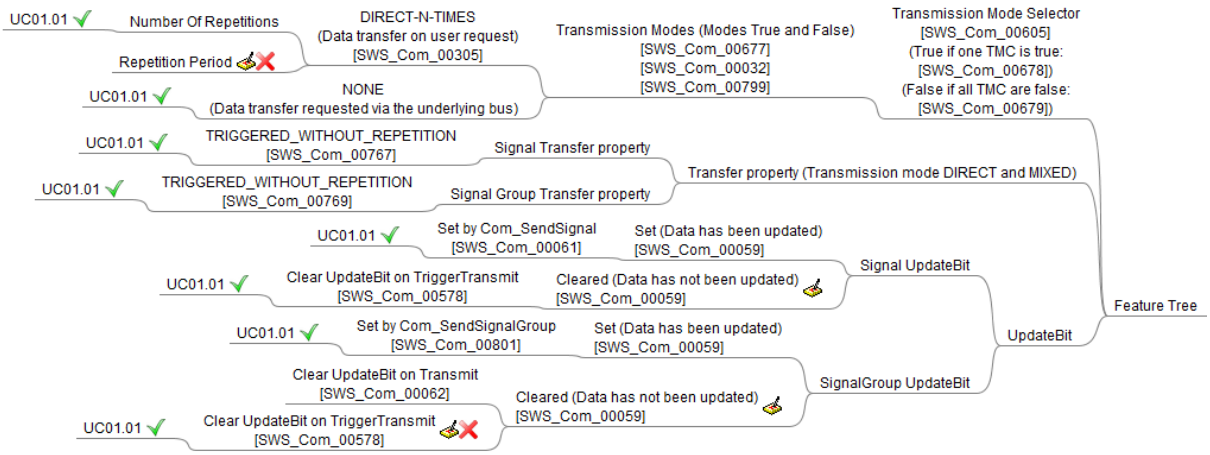

Figure 1 Mindmap of the features covered and not covered in the test cases

** The covered use cases are marked with a green check mark.*

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

3.1.1 Test System

3.1.1.1 Overview on Architecture

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

3.1.1.1.1 Use case 01.01: LIN Bus

For this use case, the aim is to test the data transfer on LIN bus:

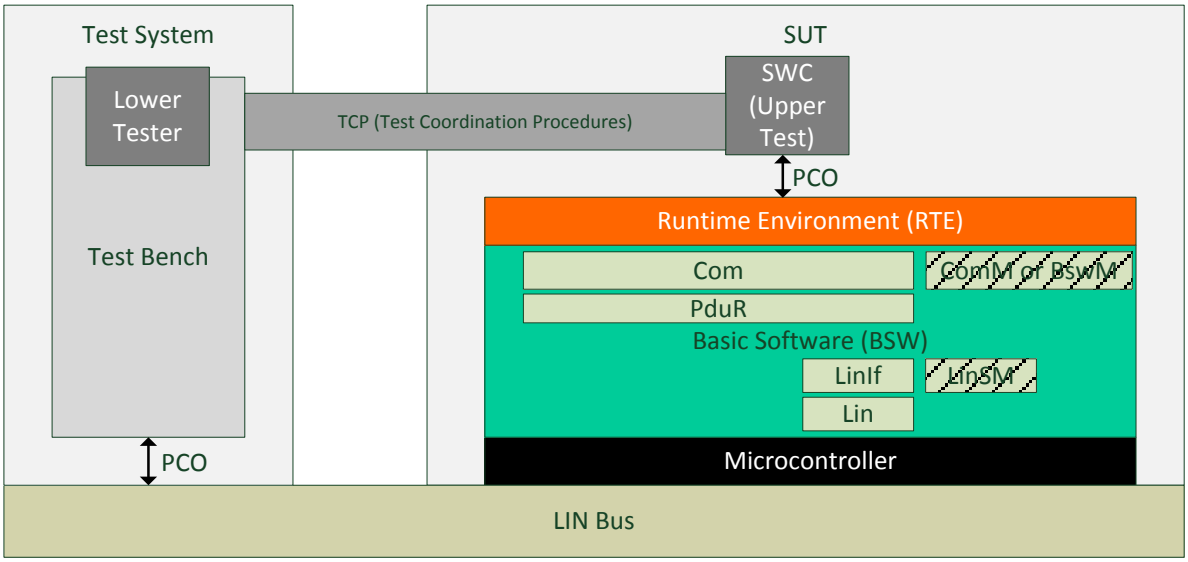

Figure 2 Acceptance test architecture required for the test cases

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

3.1.1.2 Specific Requirements

Not Applicable.

3.1.1.3 Test Coordination Requirements

Not Applicable.

3.1.2 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 suites is implemented.

3.1.2.1 Required ECU Extract of System Description Files

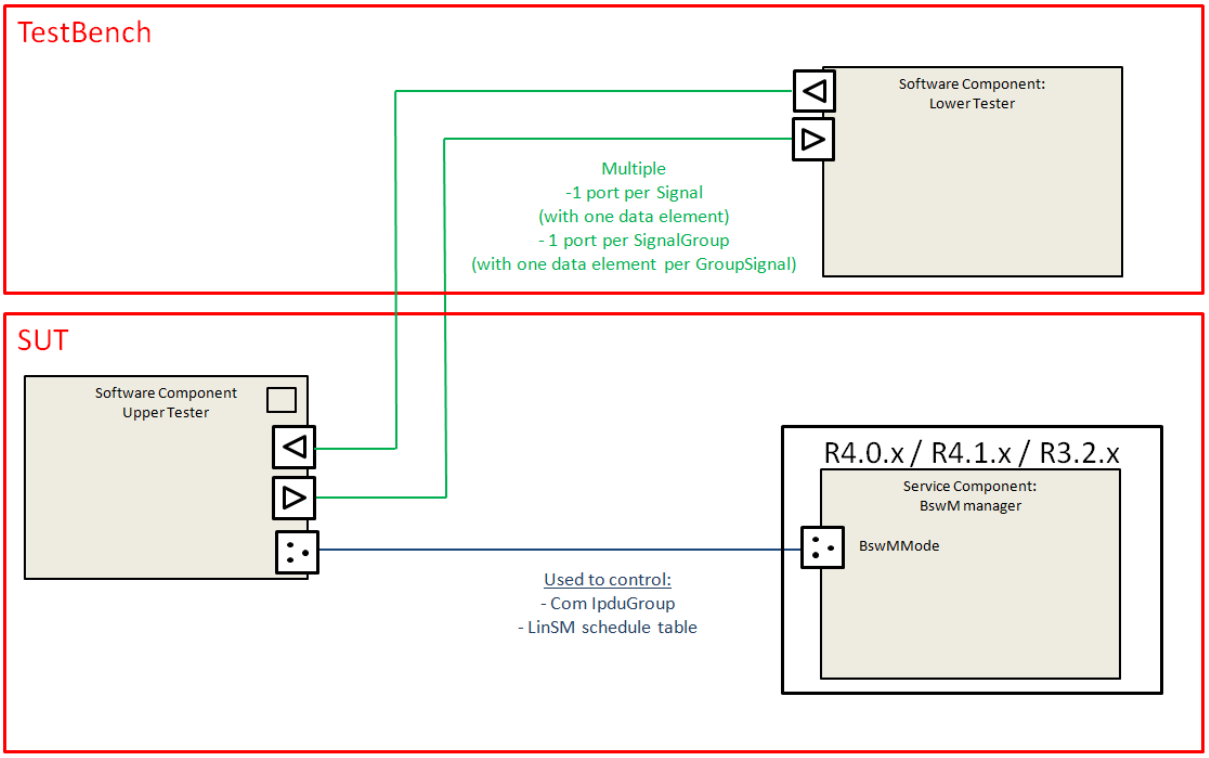

Figure 3 Required SWC description

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 3.3 Test Cases for details):

ModeDeclaration	BswM Actions
IPDU_ACTIVATED	OnEntry: -Start IpduGroup
IPDU_DEACTIVATED	OnEntry: -Stop IpduGroup
IPDU_OFF_ON	OnEntry: -Stop IpduGroup -Re-start IpduGroup
LIN_START_SCHEDULE	OnEntry: -Start LIN Schedule Table
IPDU_ACTIVATED_LIN_START_SCHEDULE	OnEntry: -Start IpduGroup -Start LIN Schedule Table

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

Port name	Data element type	Data element	Mapping	Type
<TestCaseName>_<signalname>	Uint8	<signalname>	<Signalname>	Signal

<TestCaseName>_<signalgroupname>	Struct
	{
	Uint8: groupsignal1;
	…
	Uint8: groupsignalx;
	}
		Groupsignal	Groupsignal1->
			<signal1name>
			Groupsignal2->
			<signal2name>
			<PortName>->
			<signalgroupname>
				Signal Group

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

Unless a different configuration is specified in test case, Sender/Receiver Ports uses for communication “Non queued Data Element” and “Explicit Data access” for associated runnables.

3.1.2.1.1 Use Case UC01.01: LIN Bus

The communication database is depicted below:

IPduGroup	IPdu	SignalGroup	Signal	Tx ECU	Rx ECU
AT_201_IpduGroup	AT_201_Ipdu		AT_201_Sg1	SUT	TestBench
AT_215_IpduGroup	AT_215_Ipdu	AT_215_SgGr1	AT_215_GrSg1	SUT	TestBench
			AT_215_GrSg2		
AT_216_IpduGroup	AT_216_Ipdu		AT_216_Sg1	SUT	TestBench
			AT_216_Sg2		
AT_217_IpduGroup	AT_217_Ipdu	AT_217_SgGr1	AT_217_GrSg1	SUT	TestBench
			AT_217_GrSg2		
AT_218_IpduGroup	AT_218_Ipdu		AT_218_Sg1	SUT	TestBench
AT_219_IpduGroup	AT_219_Ipdu	AT_219_SgGr1	AT_219_GrSg1	SUT	TestBench
			AT_219_GrSg2		

3.1.2.2 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

3.1.2.3 Required Software Component Description Files

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

Refer to Figure 3.

3.1.2.4 Mandatory vs. Customizable Parts

Mandatory parameters are listed in Tests Cases (see 3.3 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
- LIN frames identifiers

3.1.3 Test Case Design

Not Applicable.

3.2 Re-usable Test Steps

Not Applicable.

3.3 Test Cases

3.3.1 [ATS_COMLIN_00201] Signal on Tx requested by underlying bus - unconditional Frame (NONE)

Test Objective	Signal on Tx requested by underlying bus - unconditional Frame (NONE)		
ID	ATS_COMLIN_00201	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1
Affected Modules	Com, PduR, LinIf, Lin, LinSM	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00117		
Trace to SWS Item	COM: SWS_Com_00059 COM: SWS_Com_00061 COM: SWS_Com_00135 COM: SWS_Com_00578		
Requirements / Reference to Test Environment	Use Case UC01.01		
Configuration Parameters	ComIpdu(SignalIPdu): AT_201_Ipdu1(Mapped on LIN Frame=>LinTopology) - ComIPduDirection(CommConnectorPort.communicationDirection) = SEND - ComTxModeTrue (IPduTiming.TransmissionModeDeclaration.transmissionModeTrueTiming) -- NONE(no timing assigned) - No transmissionAcknowledge for signals contained in this Pdu - ComTxIPduClearUpdateBit = TriggerTransmit ComSignal(ISignalToPduMapping): Sg1 - updateIndicationBitPosition is configured - ComSignalInitValue(ISignal.initValue) = Sg1_Value_Init != Sg1_Value_1 LinIfFrame(LinFrameTriggering) (Tx UNCONDITIONAL Frame => Periodic frame on Lin Bus) - LinIfFrameType = UNCONDITIONAL(frame = LinUnconditionalFrame) -- LinIfPduDirection = LinIfTxPdu(FramePort.CommConnectorPort.communicationDirection) LinIfScheduleTable(LinScheduleTable) (slot only set once in RUN_CONTINUOUS Schedule Table) - runMode = RUN_CONTINUOUS - Only 1 LinIfEntry(tableEntry) linked to previously configured FrameTriggering		
Summary	Aim: - Check that sent signal is taken into account in Periodic UNCONDITIONAL Tx LIN frame. Sequence: 1) Action: Start Ipdu Group - Result: Ipdu is not sent (Tx Mode NONE) 2) Action: Start LIN Schedule Table - Result: Ipdu is sent out on associated slot execution (Tx Mode NONE [SWS_Com_00135]) - Result: Signal value is initial value (Value_Init)		

	- Result: Signal update bit is 0 3) Action: Update signal with Value_1 - Result: Periodic Time is not changed (Ipdu is always sent out on associated slot execution) - Result: UpdateBit is set to 1, only in the first send after step 3. After that, it is 0. [SWS_Com_00059][SWS_Com_00061][SWS_Com_00578] - Result: Signal value is changed to Value_1 for all new occurrences of the Tx frame
Needed Adaptation to other Releases	None
Pre-conditions	Com stack is initialized, but ipdu groups are not running Lin schedule table not started

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	***[SWC]*** Request ModeSwitch (call Rte_Switch associated to BswMMode port) to IPDU_ACTIVATED (Start Ipdu Group AT_201_IpduGroup)	***[LT<LIN>]*** AT_201_Ipdu is not sent (Tx Mode NONE)
Step 2	***[SWC]*** Request ModeSwitch (call Rte_Switch associated to BswMMode port) to LIN_START_SCHEDULE (Start LIN Schedule Table)	***[LT<LIN>]*** AT_201_Ipdu is sent out on associated slot execution (Tx Mode NONE) AT_201_Sg1 update bit is 0 AT_201_Sg1 value is initial value (AT_201_Sg1_Value_Init)
Step 3	***[SWC]*** Send signal AT_201_Sg1 with AT_201_Sg1_Value_1 (call Rte_Write() for AT_201_Sg1 Port)	**[LT<LIN>]** AT_201 Periodic Time is not changed AT_201_Sg1 UpdateBit is set to 1 in the first send. AT_201_Sg1 value is now AT_201_Sg1_Value_1
Step 4	-	**[LT<LIN>]** AT_201_Sg1 UpdateBit is set to 0 later after the first send.
Post-conditions	Not Applicable	

3.3.2 [ATS_COMLIN_00215] Signal Group on Tx requested by underlying bus - unconditional Frame (NONE)

Test Objective	Signal Group on Tx requested by underlying bus - unconditional Frame (NONE)		
ID	ATS_COMLIN_00215	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1
Affected	Com, PduR, LinIf, Lin, LinSM	**State**	reviewed

Modules	
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00117
Trace to SWS Item	COM: SWS_Com_00059 COM: SWS_Com_00135 COM: SWS_Com_00578 COM: SWS_Com_00801
Requirements / Reference to Test Environment	Use Case UC01.01
Configuration Parameters	ComIpdu(SignalIPdu):AT_215_Ipdu1(Mapped on LIN Frame=>LinTopology) - ComIPduDirection(CommConnectorPort.communicationDirection)=SEND - ComTxModeTrue (IPduTiming.TransmissionModeDeclaration.transmissionModeTrueTiming) -- NONE (no timing assigned) - ComTxIPduClearUpdateBit = TriggerTransmit ComSignalGroup(ISignalToPduMapping):SgGr1 - updateIndicationBitPosition is configured - ComGroupSignal(ISignalToPduMapping):GrSg1/GrSg2 -- GrSg1: ComSignalInitValue(ISignal.initValue) = GrSg1_Value_Init != GrSg1_Value_1 -- GrSg2: ComSignalInitValue(ISignal.initValue) = GrSg2_Value_Init != GrSg2_Value_1 LinIfFrame(LinFrameTriggering)(Tx UNCONDITIONAL Frame => Periodic) - LinIfFrameType = UNCONDITIONAL(frame = LinUnconditionalFrame) -- LinIfPduDirection = LinIfTxPdu(FramePort.CommConnectorPort.communicationDirection) LinIfScheduleTable(LinScheduleTable) - runMode = RUN_CONTINUOUS - Only 1 LinIfEntry(tableEntry) linked to previously configured FrameTriggering
Summary	Aim: - Check that SignalGroup is taken into account in Periodic UNCONDITIONAL Tx LIN frame. Sequence: 1) Action: Start Ipdu Group - Result: Ipdu is not sent (Tx Mode NONE) 2) Action: Start LIN Schedule Table - Result: Ipdu is sent out on associated slot execution (Tx Mode NONE [SWS_Com_00135]) - Result: GroupSignal values are initial value (Value_Init) - Result: SignalGroup update bit is 0 3) Action: Send SignalGroup with update of GroupSignal to Value_1 - Result: Periodic Time is not changed (Ipdu is always sent out on associated slot execution) - Result: SignalGroup UpdateBit is set to 1, only in the first send after step 3. After that, it is 0. [SWS_Com_00059][SWS_Com_00801][SWS_Com_00578] - Result: GroupSignal values are changed to Value_1 for all new occurrences of the Tx frame
Needed Adaptation to	None

other Releases		
Pre-conditions	Com stack is initialized, but ipdu groups are not running Lin schedule table not started	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	***[SWC]*** Request ModeSwitch (call Rte_Switch associated to BswMMode port) to IPDU_ACTIVATED (Start Ipdu Group AT_215_IpduGroup)	***[LT<LIN>]*** AT_215_Ipdu is not sent (Tx Mode NONE)
Step 2	***[SWC]*** Request ModeSwitch (call Rte_Switch associated to BswMMode port) to LIN_START_SCHEDULE (Start LIN Schedule Table)	***[LT<LIN>]*** AT_215_Ipdu is sent out on associated slot execution (Tx Mode NONE) AT_215_SgGr1 update bit is 0 AT_215_GrSg1 value is initial value (AT_215_GrSg1_Value_Init) AT_215_GrSg2 value is initial value (AT_215_GrSg2_Value_Init)
Step 3	***[SWC]*** AT_215_SgGr1.AT_215_GrSg1=AT_215_GrSg1_Value_1 AT_215_SgGr1.AT_215_GrSg2=AT_215_GrSg2_Value_1 call Rte_Write() for Port AT_215_SgGr1 (Rte will send group signal AT_215_GrSg1 with AT_215_GrSg1_Value_1 send group signal AT_215_GrSg2 with AT_215_GrSg2_Value_1 send signal group AT_215_SgGr1)	***[LT<LIN>]*** AT_215_Ipdu Periodic Time is not changed AT_215_SgGr1 UpdateBit is set to 1 in the first send. AT_215_GrSg1 value is now AT_215_GrSg1_Value_1 AT_215_GrSg2 value is now AT_215_GrSg2_Value_1
Step 4	-	**[LT<LIN>]** AT_215_SgGr1 UpdateBit is set to 0 later after the first send.
Post-conditions	Not Applicable	

3.3.3 [ATS_COMLIN_00216] Signal on Tx Time Base frame - sporadic frame (PERIODIC)

Test Objective	Signal on Tx Time Base frame - sporadic frame (PERIODIC)		
ID	ATS_COMLIN_00216	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1
Affected Modules	Com, PduR, LinIf, Lin, LinSM	**State**	reviewed
Trace to Requirement			

on Acceptance Test Document	
Trace to SWS Item	COM: SWS_Com_00059 COM: SWS_Com_00061 COM: SWS_Com_00222 COM: SWS_Com_00578
Requirements / Reference to Test Environment	Use Case UC01.01
Configuration Parameters	ComIpdu(SignalIPdu): AT_216_Ipdu1 (Mapped on LIN Frame => LinTopology) - ComIPduDirection(CommConnectorPort.communicationDirection) = SEND - ComTxModeTrue (IPduTiming.TransmissionModeDeclaration.transmissionModeTrueTiming) -- PERIODIC (CyclicTiming) --- timePeriod = 2 * Schedule Table Duration - ComTxIPduClearUpdateBit(no upstream template parameter) = TriggerTransmit ComSignal(ISignalToPduMapping): Sg1 - updateIndicationBitPosition is configured - ComSignalInitValue(ISignal.initValue) = Sg1_Value_Init != Sg1_Value_1 LinIfFrame(LinFrameTriggering) (Tx SPORADIC Frame) - LinIfFrameType = SPORADIC(frame = LinSporadicFrame) -- LinIfPduDirection = LinIfTxPdu(FramePort.CommConnectorPort.communicationDirection) LinIfScheduleTable(LinScheduleTable) - runMode = RUN_CONTINUOUS - Only 1 LinIfEntry(tableEntry) linked to previously configured FrameTriggering
Summary	Aim: - Check that sent signal is taken into account in Periodic SPORADIC Tx LIN frame (Period is handled by Com). Sequence: 1) Action: Start Ipdu Group and LIN Schedule Table - Result: Ipdu is sent out every PeriodTime, on next associated slot execution (Tx Mode PERIODIC [SWS_Com_00222]) - Result: Signal value is initial value (Value_Init) - Result: Signal update bit is 0 2) Action: Update signal with Value_1 - Result: Periodic Time is not changed - Result: UpdateBit is set to 1, only in the first send after step 2. After that, it is 0. [SWS_Com_00059][SWS_Com_00061][SWS_Com_00578] - Result: Signal value is changed to Value_1 for all new occurrences of the Tx frame
Needed Adaptation to other Releases	None
Pre-conditions	Com stack is initialized, but ipdu groups are not running Lin schedule table not started

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	***[SWC]***	***[LT<LIN>]***

	Request ModeSwitch (call Rte_Switch associated to BswMMode port) to IPDU_ACTIVATED_LIN_START_SCHEDULE (start Ipdu Group AT_216_IpduGroup and LIN Schedule Table)	AT_216_Ipdu is sent out every PeriodTime, on next associated slot execution (Tx Mode PERIODIC) AT_216_Sg1 update bit is 0 AT_216_Sg1 value is initial value (AT_216_Sg1_Value_Init)
Step 2	***[SWC]*** Send AT_216_Sg1 (call Rte_Write() for Port AT_216_Sg1) with AT_216_Sg1_Value_1	***[LT<LIN>]*** AT_216_Ipdu Periodic Time is not changed AT_216_Sg1 UpdateBit is set to 1 in the first send. AT_216_Sg1 value is now changed AT_216_Sg1_Value_1
Step 3	-	**[LT<LIN>]** AT_216_Sg1 UpdateBit is set to 0 later after the first send.
Post-conditions	Not Applicable	

3.3.4 [ATS_COMLIN_00217] Signal Group on Tx Time Base frame - sporadic frame (PERIODIC)

Test Objective	Signal Group on Tx Time Base frame - sporadic frame (PERIODIC)		
ID	ATS_COMLIN_00217	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1
Affected Modules	Com, PduR, LinIf, Lin, LinSM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COM: SWS_Com_00059 COM: SWS_Com_00222 COM: SWS_Com_00578 COM: SWS_Com_00801		
Requirements / Reference to Test Environment	Use Case UC01.01		
Configuration Parameters	ComIpdu(SignalIPdu): AT_217_Ipdu1(Mapped on LIN Frame => LinTopology) - ComIPduDirection(CommConnectorPort.communicationDirection) = SEND - ComTxModeTrue (IPduTiming.TransmissionModeDeclaration.transmissionModeTrueTiming) -- PERIODIC (CyclicTiming) / timePeriod = 2 * Schedule Table Duration - ComTxIPduClearUpdateBit= TriggerTransmit ComSignalGroup(ISignalToPduMapping):SgGr1 - updateIndicationBitPosition is configured - ComGroupSignal(ISignalToPduMapping):GrSg1/GrSg2		

	-- GrSg1: ComSignalInitValue(ISignal.initValue) = GrSg1_Value_Init != GrSg1_Value_1 -- GrSg2: ComSignalInitValue(ISignal.initValue) = GrSg2_Value_Init != GrSg2_Value_1 LinIfFrame(LinFrameTriggering) (Tx SPORADIC Frame) - LinIfFrameType = SPORADIC(frame = LinSporadicFrame) -- LinIfPduDirection = LinIfTxPdu(FramePort.CommConnectorPort.communicationDirection) LinIfScheduleTable(LinScheduleTable) - runMode = RUN_CONTINUOUS - Only 1 LinIfEntry(tableEntry) linked to previously configured FrameTriggering
Summary	Aim: - Check that sent SignalGroup is taken into account in Periodic SPORADIC Tx LIN frame (Period is handled by Com). Sequence: 1) Action: Start Ipdu Group and LIN Schedule Table - Result: Ipdu is sent out every PeriodTime, on next associated slot execution (Tx Mode PERIODIC [SWS_Com_00222]) - Result: GroupSignal values are initial value (Value_Init) - Result: SignalGroup update bit is 0 2) Action: Send SignalGroup with update of GroupSignal to Value_1 - Result: Periodic Time is not changed - Result: SignalGroup UpdateBit is set to 1, only in the first send after step 2. After that, it is 0. [SWS_Com_00059][SWS_Com_00801][SWS_Com_00578] - Result: GroupSignal values are changed to Value_1 for all new occurrences of the Tx frame
Needed Adaptation to other Releases	None
Pre-conditions	Com stack is initialized, but ipdu groups are not running Lin schedule table not started

Main Test Execution

Test Steps		**Pass Criteria**
Step 1	***[SWC]*** Request ModeSwitch (call Rte_Switch associated to BswMMode port) to IPDU_ACTIVATED_LIN_START_SCHEDULE (start Ipdu Group AT_217_IpduGroup and LIN Schedule Table)	***[LT<LIN>]*** AT_217_Ipdu is sent out every PeriodTime, on next associated slot execution (Tx Mode PERIODIC) AT_217_SgGr1 update bit is 0 AT_217_GrSg1 value is initial value (AT_217_GrSg1_Value_Init) AT_217_GrSg2 value is initial value (AT_217_GrSg2_Value_Init)
Step 2	***[SWC]*** AT_217_SgGr1.AT_217_GrSg1=AT_217_GrSg1_Value_1 AT_217_SgGr1.AT_217_GrSg2=AT_217_GrSg2_Value_1 Call Rte_Write() for Port AT_217_SgGr1 Rte will: - Send group signal AT_217_GrSg1 with AT_217_GrSg1_Value_1 - Send group signal AT_217_GrSg2 with	***[LT<LIN>]*** AT_217_Ipdu Periodic Time is not changed AT_217_SgGr1 UpdateBit is set to 1 in the first send. AT_217_GrSg1 value is now AT_217_GrSg1_Value_1 AT_217_GrSg2 value is now AT_217_GrSg2_Value_1

	AT_217_GrSg2_Value_1 - Send signal group AT_217_SgGr1	
Step 3	-	**[LT<LIN>]** AT_217_SgGr1 UpdateBit is set to 0 later after the first send.
Post-conditions	Not Applicable	

3.3.5 [ATS_COMLIN_00218] Signal on user request frame - sporadic frame (DIRECT)

Test Objective	Signal on user request frame - sporadic frame (DIRECT)		
ID	ATS_COMLIN_00218	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1
Affected Modules	Com, PduR, LinIf, Lin, LinSM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COM: SWS_Com_00767		
Requirements / Reference to Test Environment	Use Case UC01.01		
Configuration Parameters	ComIpdu(SignalIPdu): AT_218_Ipdu1 (Mapped on LIN Frame => LinTopology) - ComIPduDirection(CommConnectorPort.communicationDirection) = SEND - ComTxModeTrue (IPduTiming.TransmissionModeDeclaration.transmissionModeTrueTiming) -- DIRECT (EventControlledTiming) --- NumberOfRepetitions = 2 --- RepetitionPeriod = x ms ComSignal(ISignalToPduMapping): Sg1 - ComTransferProperty (transferProperty) = TRIGGERED_WITHOUT_REPETITION - ComSignalInitValue(ISignal.initValue) = Sg1_Value_Init != Sg1_Value_1 LinIfFrame(LinFrameTriggering) (Tx SPORADIC Frame) - LinIfFrameType = SPORADIC(frame = LinSporadicFrame) -- LinIfPduDirection = LinIfTxPdu(FramePort.CommConnectorPort.communicationDirection) LinIfScheduleTable(LinScheduleTable) - runMode = RUN_CONTINUOUS - Only 1 LinIfEntry(tableEntry) linked to previously configured FrameTriggering		
Summary	Aim: - Check that sent signal is taken into account in SPORADIC Tx LIN frame. Sequence:		

	1) Action: Start Ipdu Group and LIN Schedule Table - Result: Ipdu is not sent out 2) Action: Update signal with Value_1 (Triggered without repetition) [SWS_Com_00767] - Result: Ipdu is sent only one time with Value_1	
Needed Adaptation to other Releases	None	
Pre-conditions	Com stack is initialized, but ipdu groups are not running Lin schedule table not started	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	***[SWC]*** Request ModeSwitch (call Rte_Switch associated to BswMMode port) to IPDU_ACTIVATED_LIN_START_SCHEDULE (start Ipdu Group AT_218_IpduGroup and LIN Schedule Table)	***[LT<LIN>]*** AT_218_Ipdu is not sent out
Step 2	***[SWC]*** Send signal AT_218_Sg1 (call Rte_Write() for Port AT_218_Sg1) with AT_218_Sg1_Value_1 (Triggered without repetition)	***[LT<LIN>]*** AT_218 Ipdu is sent only one time AT_218_Sg1 value is AT_218_Sg1_Value_1
Post-conditions	Not Applicable	

3.3.6 [ATS_COMLIN_00219] Signal Group on user request frame - sporadic frame (DIRECT)

Test Objective	Signal Group on user request frame - sporadic frame (DIRECT)		
ID	ATS_COMLIN_00219	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1
Affected Modules	Com, PduR, LinIf, Lin, LinSM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COM: SWS_Com_00769		
Requirements / Reference to Test Environment	Use Case UC01.01		
Configuration Parameters	ComIpdu(SignalIPdu): AT_219_Ipdu1 (Mapped on LIN Frame => LinTopology) - ComIPduDirection(CommConnectorPort.communicationDirection) = SEND - ComTxModeTrue (IPduTiming.TransmissionModeDeclaration.transmissionModeTrueTiming)		

	-- DIRECT (EventControlledTiming) --- NumberOfRepetitions = 2 --- RepetitionPeriod = x ms ComSignalGroup(ISignalToPduMapping): SgGr1 - ComTransferProperty (transferProperty) = TRIGGERED_WITHOUT_REPETITION - ComGroupSignal(ISignalToPduMapping): GrSg1/GrSg2 -- GrSg1: ComSignalInitValue(ISignal.initValue) = GrSg1_Value_Init != GrSg1_Value_1 -- GrSg2: ComSignalInitValue(ISignal.initValue) = GrSg2_Value_Init != GrSg2_Value_1 LinIfFrame(LinFrameTriggering) (Tx SPORADIC Frame) - LinIfFrameType = SPORADIC(frame = LinSporadicFrame) -- LinIfPduDirection = LinIfTxPdu(FramePort.CommConnectorPort.communicationDirection) LinIfScheduleTable(LinScheduleTable) - runMode = RUN_CONTINUOUS - Only 1 LinIfEntry(tableEntry) linked to previously configured FrameTrigger
Summary	Aim: - Check that sent SignalGroup is taken into account in SPORADIC Tx LIN frame. Sequence: 1) Action: Start Ipdu Group and LIN Schedule Table - Result: Ipdu is not sent out 2) Action: Send SignalGroup (Triggered without repetition) without updating GroupSignal [SWS_Com_00769] - Result: Ipdu is sent only one time with GroupSignal not updated (Value_Init) 3) Action: Send SignalGroup (Triggered without repetition) with update of GroupSignal to Value_1 [SWS_Com_00769] - Result: Ipdu is sent only one time with GroupSignal updated (Value_1)
Needed Adaptation to other Releases	None
Pre-conditions	Com stack is initialized, but ipdu groups are not running Lin schedule table not started

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	***[SWC]*** Request ModeSwitch (call Rte_Switch associated to BswMMode port) to IPDU_ACTIVATED_LIN_START_SCHEDULE (start Ipdu Group AT_219_IpduGroup and LIN Schedule Table)	***[LT<LIN>]*** AT_219_Ipdu is not sent out
Step 2	***[SWC]*** AT_219_SgGr1.AT_219_GrSg1=AT_219_GrSg1_Value_Init AT_219_SgGr1.AT_219_GrSg2=AT_219_GrSg2_Value_Init Call Rte_Write() for Port AT_219_SgGr1 - Send GroupSignal AT_219_GrSg1 with	***[LT<LIN>]*** AT_219_Ipdu is sent only one time AT_219_GrSg1 value is AT_219_GrSg1_Value_Init AT_219_GrSg2 value is AT_219_GrSg2_Value_Init

	AT_219_GrSg1_Value_Init - Send GroupSignal AT_219_GrSg2 with AT_219_GrSg2_Value_Init - Send SignalGroup AT_219_SgGr1 (Triggered without repetition)	
Step 3	***[SWC]*** AT_219_SgGr1.AT_219_GrSg1=AT_219_GrSg1_Value_1 AT_219_SgGr1.AT_219_GrSg2=AT_219_GrSg2_Value_1 Call Rte_Write() for Port AT_219_SgGr1 - Send GroupSignal AT_219_GrSg1 with AT_219_GrSg1_Value_1 - Send GroupSignal AT_219_GrSg2 with AT_219_GrSg2_Value_1 - Send SignalGroup AT_219_SgGr1 (Triggered without repetition)	***[LT<LIN>]*** AT_219_Ipdu is sent only one time AT_219_GrSg1 value is AT_219_GrSg1_Value_1 AT_219_GrSg2 value is AT_219_GrSg2_Value_1
Post-conditions	Not Applicable	

4 RS_BRF_01648 - Large Data Type

4.1 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:

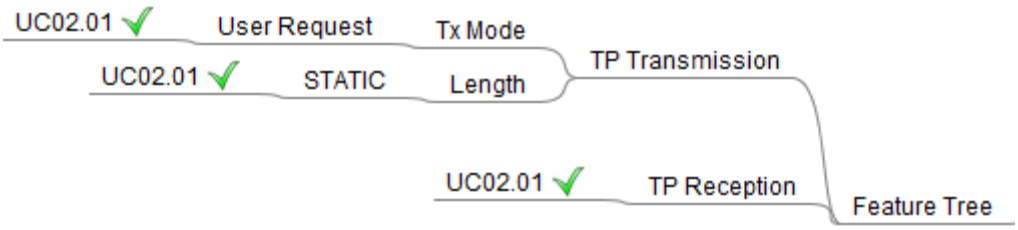

Figure 4 Mindmap of the features covered and not covered in the test cases

** The covered use cases are marked with a green check mark.*

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 coverage, the environment has been separated in several uses case.

4.1.1.1.1 Use case 02.01: LIN Bus

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

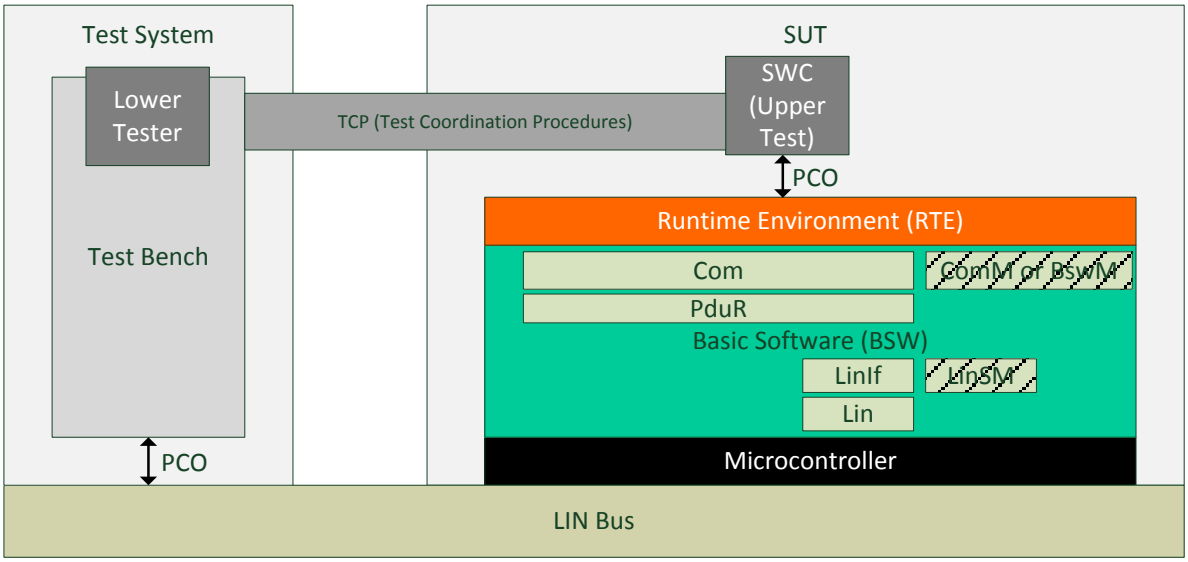

Figure 5 Acceptance test architecture required for the test cases

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 suites is implemented.

4.1.2.1 Required ECU Extract of System Description Files

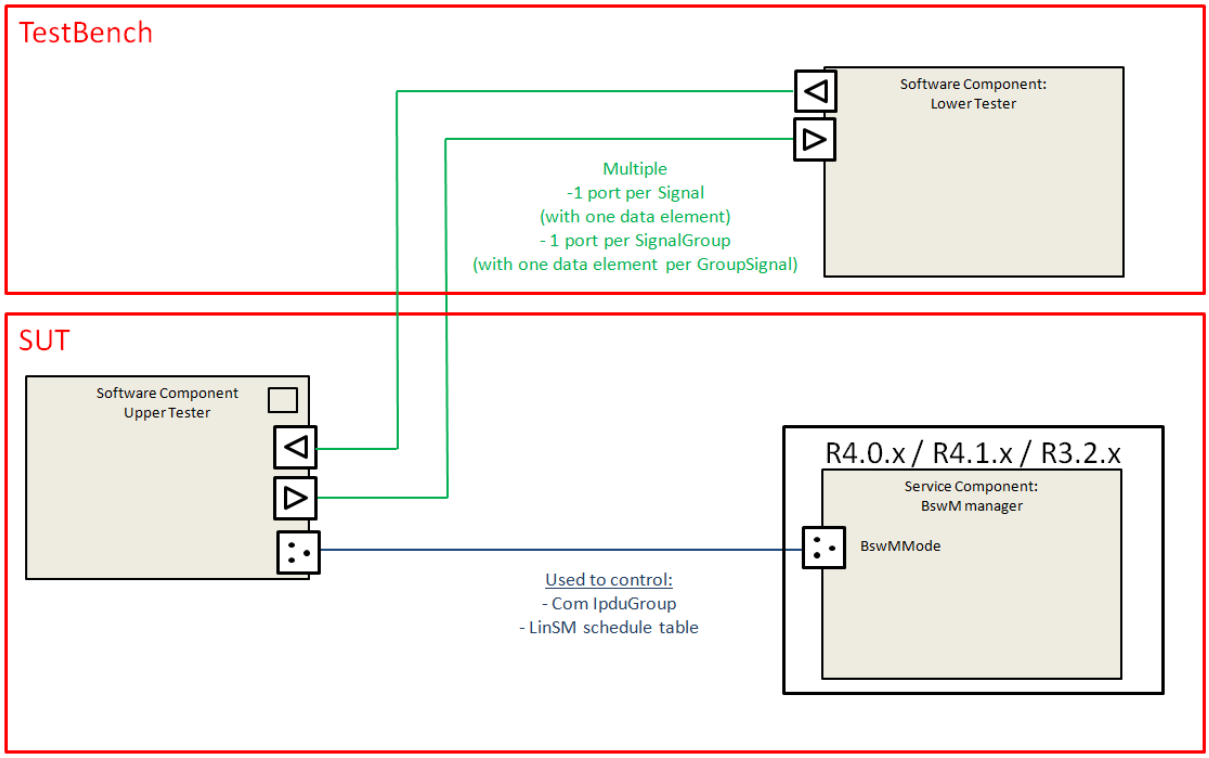

Figure 6 Required SWC description

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 4.3 Test Cases for details):

ModeDeclaration	BswM Actions
IPDU_ACTIVATED	OnEntry: -Start IpduGroup
IPDU_DEACTIVATED	OnEntry: -Stop IpduGroup
IPDU_OFF_ON	OnEntry: -Stop IpduGroup -Re-start IpduGroup
LIN_START_SCHEDULE	OnEntry: -Start LIN Schedule Table
IPDU_ACTIVATED_LIN_START_SCHEDULE	OnEntry: -Start IpduGroup -Start LIN Schedule Table

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

Port name	Data element type	Data element	Mapping	Type
<TestCaseName>_<signalname>	Uint8	<signalname>	<Signalname>	Signal

<TestCaseName>_<signalgroupname>	Struct
	{
	Uint8: groupsignal1;
	…
	Uint8: groupsignalx;
	}
		Groupsignal	Groupsignal1->
			<signal1name>
			Groupsignal2->
			<signal2name>
			<PortName>->
			<signalgroupname>
				Signal Group

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

Unless a different configuration is specified in test case, Sender/Receiver Ports used for communication queued Data Element and Explicit Data access for associated runnables.

4.1.2.1.1 Use Case 02.01: LIN Bus

The communication database is depicted below:

IPduGroup	IPdu	SignalGroup	Signal	Tx ECU	Rx ECU
AT_241_IpduGroup	AT_241_Ipdu		AT_241_Sg1	SUT	TestBench
AT_277_IpduGroup	AT_277_Ipdu		AT_277_Sg1	TestBench	SUT

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 Figure 6.

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
- LIN frames identifiers

4.1.3 Test Case Design

Not Applicable.

4.2 Re-usable Test Steps

Not Applicable.

4.3 Test Cases

4.3.1 [ATS_COMLIN_00241] Large Data TP transmission on LIN (>= 7 bytes)

Test Objective	Large Data TP transmission on LIN (>= 7 bytes)		
ID	ATS_COMLIN_00241	**AUTOSAR Releases**	4.0.3|4.1.1|4.2.1
Affected Modules	Com, PduR, LinTp, LinIf, Lin	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COM: ECUC_Com_00761		
Requirements / Reference to Test Environment	Use Case UC02.01		
Configuration Parameters	ComIpdu(SignalIPdu): AT_241_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 = 0 - ComTxIPduClearUpdateBit = TriggerTransmit 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_LinTp
Summary	Aim: - Check that Application layer can initiate a TP transmission greater than or equal to 7 bytes on LIN 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 shall be removed
Pre-conditions	Com stack is initialized AT_241_IpduGroup is not running LIN schedule table is started

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	***[SWC]*** Request ModeSwitch (call Rte_Switch associated to BswMMode port) to IPDU_ACTIVATED (Start Ipdu Group AT_241_IpduGroup)	***[LT<LIN>]*** AT_241_Ipdu is not sent out
Step 2	***[SWC]*** Call Rte_Send() for Port AT_241_Sg1 with AT_241_Sg1_Value_1 (Send AT_241_Sg1 with AT_241_Sg1_Value_1 (this will initiate a TP transmission with 9 bytes))	***[LT<LIN>]*** First Frame is received frame length is 8 byte, LEN (third frame byte) is 9 bytes Data contained in Bytes 4 to 8
Step 3	***[LT<LIN>]*** Wait Consecutive Frame reception	***[LT<LIN>]*** 1 Consecutive Frame is received frame length is 8 byte Data contained in Bytes 3 to 6 AT_241_Sg1 value is AT_241_Sg1_Value_1
Post-conditions	Not Applicable	

4.3.2 [ATS_COMLIN_00277] Large Data TP reception on LIN (>= 7 bytes)

Test Objective	Large Data TP reception on LIN (>= 7 bytes)		
ID	ATS_COMLIN_00277	**AUTOSAR Releases**	4.0.3|4.1.1|4.2.1
Affected Modules	Com, PduR, LinTp, LinIf, Lin	**State**	reviewed
Trace to Requirement			

on Acceptance Test Document	
Trace to SWS Item	COM: ECUC_Com_00761
Requirements / Reference to Test Environment	Use Case UC02.01
Configuration Parameters	ComIpdu(SignalIPdu): AT_277_Ipdu1 (large I-PDU) - length = 9 (large, greater than a Single Frame) - ComIPduType = TP(TpConfig.TpConnection) - ComIPduDirection(CommConnectorPort.communicationDirection) = RECEIVE - ComTxIPduClearUpdateBit = TriggerTransmit ComSignal(ISignalToPduMapping): Sg1 - dataElement with queued swImplPolicy - DataReceivedEvent mapped on signal reception (ComNotification is configured) PduRRoutingPath: - Routing path for ComIpdu with PduRSrcBswModuleRef = BswMod_LinTp - PduRDestPdu with PduRDestBswModuleRef = BswMod_Com
Summary	Aim: - Check that Application layer can receive a TP Data greater than or equal to 7 bytes on LIN bus
Needed Adaptation to other Releases	Configuration: [n/a] — Large data types and TP for regular COM is not possible in R3.x. Test Steps: [n/a] — This test case shall be removed
Pre-conditions	Com stack is initialized AT_277_IpduGroup is not running LIN schedule table is started

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	***[SWC]*** Request ModeSwitch (call Rte_Switch associated to BswMMode port) to IPDU_ACTIVATED (Start Ipdu Group AT_277_IpduGroup)	***[SWC]*** No DataReceivedEvent for AT_277_Sg1
Step 2	***[LT<LIN>]*** On SRF header reception (Slave Response Frame), Send Signal AT_277_Sg1 with AT_277_Sg1_Value_1 (this will initiate a TP transmission with 9 bytes)	***[LT<LIN>]*** First Frame is sent frame length is 8 byte, LEN (third frame byte) is 9 bytes Data contained in Bytes 4 to 8
Step 3	***[LT<LIN>]*** On next SRF header reception (Slave Response Frame), Send Consecutive Frame with last data bytes	***[LT<LIN>]*** One Consecutive Frame is received

Step 4	***[CP]*** Wait DataReceivedEvent	***[SWC]*** DataReceivedEvent is activated
Step 5	***[SWC]*** Call Rte_Receive() for AT_277_Sg1	***[SWC]*** AT_277_Sg1 value is AT_277_Sg1_Value_1 Return Value of Rte_Receive is RTE_E_OK
Post-conditions	Not Applicable	

4.3.3 [ATS_COMLIN_00750] Testing The Dut For LIN Frame Reception

Test Objective	Testing The Dut For LIN Frame Reception		
ID	ATS_COMLIN_00750	**AUTOSAR Releases**	4.0.3|4.2.1
Affected Modules	LINIF	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	LINInterface: SWS_LinIf_00419 LINInterface: SWS_LinIf_00030 LINInterface: SWS_LinIf_00033 LINInterface: SWS_LinIf_00674		
Requirements / Reference to Test Environment	none		
Configuration Parameters	BswMUserCallout = App_Linif_CurSch LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH05 Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.runMode = RUN_CONTINUOUS ComNotification = App_LinIf_Rte_Com_CbkRxAck CoreTopology::PhysicalChannel::LinframeTriggering.identifier = 0x85 ApplicationSwComponentType_ExplicitInterReceive: PPortPrototype_TC2 VariableDataPrototype_TC2BswMUserCallout = App_Linif_CurSch LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH05 Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.runMode = RUN_CONTINUOUSComNotification = App_LinIf_Rte_Com_CbkRxAck CoreTopology::PhysicalChannel::LinframeTriggering.identifier = 0x85ApplicationSwComponentType_ExplicitInterReceive: PPortPrototype_TC2 VariableDataPrototype_TC2		
Summary	Application Send the request for changing ComM mode to no communication and configure BswM user callout to probe the operational mode of transceiver.		
Needed Adaptation to other Releases			
Pre-conditions	ComM shall be in FULL_COMMUNICATION mode		

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Request for BSWM mode change through runnable entity and request for LIN frame transmission	[SWC] App_LinSM_BswM_CurSch shall be invoked and the current schedule table shall be indicated as LINIF_LINSCH05
Step 2	[LT] The tester shall transmit the Payload after seeing the corresponding frame header	[SWC] App_LinIf_Rte_Com_CbkRxAck shall be invoked
Step 3	[SWC] Invoke Rte_Read for the signal	[SWC] Signal value shall be the same as transmitted
Post-conditions	None	

4.3.4 [ATS_COMLIN_00751] Start Of Reception Is Indicated When Slave Response Frame Is Indicated By First Frame Or Single Frame

Test Objective	Start Of Reception Is Indicated When Slave Response Frame Is Indicated By First Frame Or Single Frame		
ID	ATS_COMLIN_00751	**AUTOSAR Releases**	4.0.3|4.2.1
Affected Modules	LINIF	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	LINInterface: SWS_LinIf_00075 LINInterface: SWS_LinIf_00076 LINInterface: SWS_LinIf_00078		
Requirements / Reference to Test Environment	none		
Configuration Parameters	BswMUserCallout = App_Linif_CurSch LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH06 Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.runMode = RUN_CONTINUOUS ComNotification = App_LinIf_Rte_Com_CbkRxAck ApplicationSwComponentType_ExplicitInterSend: PPortPrototype_TC2 VariableDataPrototype_TC2BswMUserCallout = App_Linif_CurSch LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH06 Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.runMode = RUN_CONTINUOUS ComNotification = App_LinIf_Rte_Com_CbkRxAck ApplicationSwComponentType_ExplicitInterSend: PPortPrototype_TC2 VariableDataPrototype_TC2		
Summary	Send a request for changing BswM mode to switch Lin schedule to LINIF_LINSCH06A requesting slave Response Frame. LinSM shall notify BswM of		

	current schedule table configure BswM user callout to probe the current schedule. The frame reception (SF and FF) shall be verified by configuring a com callback notification for the signal and validating the data in buffer.
Needed Adaptation to other Releases	
Pre-conditions	ComM shall be in Full Communication mode

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Request for BSWM mode change through and request for LIN frame transmission.	[SWC] App_LinSM_BswM_CurSch shall be invoked and the current schedule table shall be indicated as LINIF_LINSCH06
Step 2	[LT] Transmit the TP payload after seeing the corresponding frame header	[SWC] App_LinIf_Rte_Com_CbkRxAck shall be invoked
Step 3	[LT] Observe the consecutive frame configured in Lin schedule table along with rest of data shall transmitted	[SWC] App_LinIf_Rte_Com_CbkRxAck shall be invoked
Post-conditions	None	

4.3.5 [ATS_COMLIN_00752] LIN TP Notifies Upper Layer Of Abortion Of Reception

Test Objective	LIN TP Notifies Upper Layer Of Abortion Of Reception		
ID	ATS_COMLIN_00752	**AUTOSAR Releases**	4.0.3|4.2.1
Affected Modules	LINIF	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	LINInterface: SWS_LinIf_00075 LINInterface: SWS_LinIf_00076 LINInterface: SWS_LinIf_00078		
Requirements / Reference to Test Environment	none		
Configuration Parameters	BswMUserCallout = App_Linif_CurSch LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH07 Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.runMode = RUN_CONTINUOUS ComNotification = App_LinIf_Rte_Com_CbkRxAck ComErrorNotification = App_LinIf_CbkErr_TC_07 ApplicationSwComponentType_ExplicitInterSend: PPortPrototype_TC2		

	VariableDataPrototype_TC2 BswMUserCallout = App_Linif_CurSch LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH07 Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.runMode = RUN_CONTINUOUS ComNotification = App_LinIf_Rte_Com_CbkRxAck ComErrorNotification = App_LinIf_CbkErr_TC_07 ApplicationSwComponentType_ExplicitInterSend: PPortPrototype_TC2 VariableDataPrototype_TC2	
Summary	Send a request for changing BswM mode to switch Lin schedule to LINIF_LINSCH07. In case incorrect PCI is received (SF is received after a CF) LinIf shall abort the TP reception.	
Needed Adaptation to other Releases		
Pre-conditions	ComM shall be in Full communication mode	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Request for BSWM mode change through runnable entity and request for LIN frame transmission.	[SWC] App_LinSM_BswM_CurSch shall be invoked and the current schedule table shall be indicated as LINIF_LINSCH07 Runnable entity shall be invoked.
Step 2	[LT] Monitor and validate the valid LIN frame header and valid LIN frame shall be transmitted.	-
Step 3	[LT] The consecutive frame configured in Lin schedule table shall be transmitted to DUT	[SWC] App_LinIf_Rte_Com_CbkRxAck shall be invoked
Step 4	[SWC] Trigger Rte_Read communication for the signal	[LT] The Data which was transmitted in the previous step shall be observed.
Step 5	[LT] Monitor and validate the valid LIN frame header and valid LIN frame shall be transmitted.	-
Step 6	[LT] The consecutive frame configured in Lin schedule table shall be transmitted to DUT	-
Step 7	[LT] The consecutive frame configured in Lin schedule table with rest of data shall be transmitted to DUT	[SWC] App_LinIf_CbkErr_TC_07shall be invoked indicating reception error.
Post-conditions	None	

4.3.6 [ATS_COMLIN_00753] LIN TP Converts N-Sdu Id To Specific Channel And A Destination Nad For The Slave

Test Objective	LIN TP Converts N-Sdu Id To Specific Channel And A Destination Nad For The Slave		
ID	ATS_COMLIN_00753	**AUTOSAR Releases**	4.0.3|4.2.1
Affected Modules	LINIF	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	LINInterface: SWS_LinIf_00422		
Requirements / Reference to Test Environment	none		
Configuration Parameters	BswMUserCallout = App_Linif_CurSc LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH11 Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.runMode = RUN_CONTINUOUS ApplicationSwComponentType_ExplicitInterSend: PPortPrototype_TC2 VariableDataPrototype_TC2 BswMUserCallout = App_Linif_CurSc LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH11 Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.runMode = RUN_CONTINUOUS ApplicationSwComponentType_ExplicitInterSend: PPortPrototype_TC2 VariableDataPrototype_TC2		
Summary	Send a request for changing BswM mode to switch Lin schedule to LINIF_LINSCH11. The requirement can be verified by monitoring and validating the frames on bus.		
Needed Adaptation to other Releases			
Pre-conditions	ComM shall be in Full communication mode		
Main Test Execution			
Test Steps		**Pass Criteria**	
Step 1	[SWC] Invoke Rte_Write with signal and data	[SWC] Rte_Write shall return E_OK	
Step 2	[SWC] Request for BSWM mode change through runnable entity and request for LIN frame transmission.	[SWC] App_Linif_CurSc shall be invoked and the current schedule table shall be indicated as LINIF_LINSCH11 Runnable entity shall be invoked	
Step 3	[LT] Monitor and validate the frames	[LT] The LIN frame with data transmitted in step 2 shall be observed	
Step 4	[LT] Monitor and validate the frames for the configured CF entry in the schedule table	[LT] Frames shall be observed on the bus with the data transmitted	
Post-conditions	None		

4.3.7 [ATS_COMLIN_00754] Transmission Of Diagnostic Frames (Mrf)

Test Objective	Transmission Of Diagnostic Frames (Mrf)		
ID	ATS_COMLIN_00754	**AUTOSAR Releases**	4.0.3|4.2.1
Affected Modules	LINIF_Conf	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	LINInterface: SWS_LinIf_00066		
Requirements / Reference to Test Environment	none		
Configuration Parameters	BswMUserCallout = App_Linif_CurSch LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH12 ApplicationSwComponentType_ExplicitInterSend:PPortPrototype_TC2 VariableDataPrototype_TC2 BswMUserCallout = App_Linif_CurSch LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH12 ApplicationSwComponentType_ExplicitInterSend: PPortPrototype_TC2 VariableDataPrototype_TC2		
Summary	Send a request for changing BswM mode to switch Lin schedule to LINIF_LINSCH12. The MRF Header and Response frames transmission can be verified by monitoring and validating the frames on bus.		
Needed Adaptation to other Releases			
Pre-conditions	ComM shall be in Full communication mode		
Main Test Execution			
Test Steps		**Pass Criteria**	
Step 1	[SWC] Request for BSWM mode change through runnable entity and request for Diagnostic MRF transmission (Hint: via TP message).	[SWC] App_Linif_CurSc shall be invoked and the current schedule table shall be indicated as LINIF_LINSCH11 Runnable entity shall be invoked	
Step 2	[LT] Monitor and validate the frames	[LT] Observe Master Request Frames on the bus	
Post-conditions	None		

4.3.8 [ATS_COMLIN_00755] Transmission Of Diagnostic Frames (Srf)

Test Objective	Transmission Of Diagnostic Frames (Srf)

ID	ATS_COMLIN_00755	**AUTOSAR Releases**	4.0.3|4.2.1
Affected Modules	LINIF	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	LINInterface: SWS_LinIf_00023		
Requirements / Reference to Test Environment	none		
Configuration Parameters	BswMUserCallout = App_Linif_CurSch LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH13 Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.runMode = RUN_CONTINUOUS ApplicationSwComponentType_ExplicitInterSend:PPortPrototype_TC2 VariableDataPrototype_TC2 BswMUserCallout = App_Linif_CurSch LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH13 Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.runMode = RUN_CONTINUOUS ApplicationSwComponentType_ExplicitInterSend: PPortPrototype_TC2 VariableDataPrototype_TC2		
Summary	Send a request for changing BswM mode to switch Lin schedule to LINIF_LINSCH13. The SRF transmission can be verified by monitoring and validating the slave response frames on bus.		
Needed Adaptation to other Releases			
Pre-conditions	ComM shall be in Full communication mode		
Main Test Execution			
Test Steps		**Pass Criteria**	
Step 1	[SWC] Request for BSWM mode change through runnable entity and request for Diagnostic SRF transmission.	[SWC] App_Linif_CurSc shall be invoked and the current schedule table shall be indicated as LINIF_LINSCH13 Runnable entity shall be invoked	
Step 2	[LT] Observe the LIN frame header with PID value as 0x3D. Then transmit the LIN frame from tester	[LT] LIN frame header shall be observed by tester	
Step 3	[LT] Monitor and validate the Frames	[LT] Slave Request Frames transmitted in step-2 shall be observed on the bus	
Post-conditions	None		

4.3.9 [ATS_COMLIN_00756] Run Once Schedule Table Has Higher Priority Than Run Continuous Schedule Table

Test Objective	Run Once Schedule Table Has Higher Priority Than Run Continuous Schedule Table		
ID	ATS_COMLIN_00756	**AUTOSAR Releases**	4.0.3|4.2.1
Affected Modules	LINIF	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	LINInterface: SWS_LinIf_00393 LINInterface: SWS_LinIf_00397		
Requirements / Reference to Test Environment	none		
Configuration Parameters	BswMUserCallout = App_Linif_CurSch LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH151 Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.runMode = RUN_ONCE LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH152 Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.runMode = RUN_CONTINUOUS CoreTopology::PhysicalChannel::LinframeTriggering.identifier = 0xCA CoreTopology::PhysicalChannel::LinframeTriggering.identifier = 0xBB CoreTopology::PhysicalChannel::LinframeTriggering.identifier = 0x4C ApplicationSwComponentType_ExplicitInterSend:PPortPrototype_TC2 VariableDataPrototype_TC2 BswMUserCallout = App_Linif_CurSch LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH151 Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.runMode = RUN_ONCE LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH152 Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.runMode = RUN_CONTINUOUS CoreTopology::PhysicalChannel::LinframeTriggering.identifier = 0xCA CoreTopology::PhysicalChannel::Linfr		
Summary	Send a request for changing BswM mode to switch Lin schedule and configure a BswM user callout on LinSM current state notification to probe current state of LinSM.		
Needed Adaptation to other Releases			
Pre-conditions	ComM shall be in Full communication mode		
Main Test Execution			
Test Steps		**Pass Criteria**	
Step 1	[SWC] INVOKE Rte_Write with signal and data	[SWC] Rte_Write shall return E_OK	
Step 2	[SWC] Request for BSWM mode change through runnable entity and request for LIN frame transmission.	[SWC] App_Linif_CurSch shall be invoked and the current schedule table shall be indicated as LINIF_LINSCH151	

		Runnable entity shall be invoked
Step 3	[LT] Monitor and validate the frames	[LT] Frame transmitted in step-2 shall be observed
Step 4	[SWC] INVOKE Rte_Write with signal and data	[SWC] Rte_Write shall return E_OK
Step 5	[SWC] Request for BSWM mode change through runnable entity and request for LIN frame transmission.	[SWC] App_Linif_CurSch shall be invoked and the current schedule table shall be indicated as LINIF_LINSCH152 Runnable entity shall be invoked
Step 6	[LT] Monitor and validate the frames	[LT] Frame transmitted in step-5 shall be observed
Post-conditions	None	

4.3.10 [ATS_COMLIN_00757] Schedule Table Change Request After TP Transmission With Parameter LIN TP Diagnostic Response

Test Objective	Schedule Table Change Request After TP Transmission With Parameter LIN TP Diagnostic Response		
ID	ATS_COMLIN_00757	**AUTOSAR Releases**	4.0.3|4.2.1
Affected Modules	LINIF	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	LINInterface: SWS_LinIf_00641 LINInterface: SWS_LinIf_00642		
Requirements / Reference to Test Environment	none		
Configuration Parameters	BswMUserCallout = App_Linif_CurSch LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH16 BswMUserCallout = App_Linif_User_linTpMd ApplicationSwComponentType_ExplicitInterSend:PPortPrototype_TC2 VariableDataPrototype_TC2BswMUserCallout = App_Linif_CurSch LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH16BswMUserCallout = App_Linif_User_linTpMdApplicationSwComponentType_ExplicitInterSend: PPortPrototype_TC2 VariableDataPrototype_TC2		
Summary	Send a request for changing BswM mode to switch Lin schedule and configure BswM user callouts on LinSM current state notification and LinTp request mode to probe current state of LinSM and LinTp request mode.		
Needed Adaptation to other Releases			

Pre-conditions	ComM shall be in Full communication mode	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Request for BSWM mode change through runnable entity and request for diagnostic frame transmission.	[SWC] App_Linif_CurSch shall be invoked indicating Diagnostic request ScheduleApp_Linif_User_linTpMd shall be invoked indicating TP mode as LinTP Diagnostic Request
Step 2	[LT] Frames shall be monitored on the bus	[LT] Master request frame shall be observed on the bus
Step 3	[SWC] Request for BSWM mode change through runnable entity and requestfor diagnostic frame transmission.	[SWC] App_Linif_User_linTpMd shall be invoked indicating Diagnostic response Schedule App_Linif_User_linTpMd shall be invoked indicating TP mode as LinTP Diagnostic Response
Post-conditions	None	

4.3.11 [ATS_COMLIN_00758] Schedule Table Change Request After TP Transmission To LIN TP Applicative Schedule

Test Objective	Schedule Table Change Request After TP Transmission To LIN TP Applicative Schedule		
ID	ATS_COMLIN_00758	**AUTOSAR Releases**	4.0.3|4.2.1
Affected Modules	LINIF	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	LINInterface: SWS_LinIf_00641 LINInterface: SWS_LinIf_00643		
Requirements / Reference to Test Environment	none		
Configuration Parameters	BswMUserCallout = App_Linif_CurSch LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH17 BswMUserCallout = App_Linif_User_linTpMd_19 ApplicationSwComponentType_ExplicitInterSend: PPortPrototype_TC2 VariableDataPrototype_TC2 BswMUserCallout = App_Linif_CurSch LINSM_SCHEDULE_INDEX_REF = LINIF_LINSCH17 BswMUserCallout = App_Linif_User_linTpMd_19 ApplicationSwComponentType_ExplicitInterSend: PPortPrototype_TC2 VariableDataPrototype_TC2		

Summary	Send a request for changing BswM mode to switch Lin schedule and configure BswM user callouts on LinSM current state notification and LinTp request mode to probe current state of LinSM and LinTp request mode.	
Needed Adaptation to other Releases		
Pre-conditions	ComM shall be in Full communication mode	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Request for BSWM mode change through runnable entity and request for frame transmission.	[SWC] App_Linif_CurSch shall be invoked indicating Applicative request Schedule App_Linif_User_linTpMd shall be invoked indicating TP mode as LinTP Diagnostic Request
Step 2	[LT] Frames shall be monitored on the bus	[LT] Master request frame shall be observed on the bus
Step 3	[SWC] Request for BSWM mode change through runnable entity and request for frame transmission.	[SWC] App_Linif_CurSch shall be invoked indicating Diagnostic response Schedule App_Linif_User_linTpMd shall be invoked indicating TP mode as LinTP Diagnostic Response
Post-conditions	None	