Document Title	Acceptance Test Specification of Communication Management
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	628
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 • Updated following Test cases: 00189,00190,00781,00760 to 00764,00191,00816,00819,00794,00809, 00812,00802,00776,00777,00180,00182, 00792,00799,00800,00806,00807,00813, 00818. • Removed Test case 00763.
2015-10-31	1.1.0	AUTOSAR Release Management	• Checked and adapted to Classic Platform Release 4.2.1 • Formalized point of control and observation • Added 62 test cases • Formal Changes
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 Scope

The following test cases are used to verify the correct behavior of all the communication Management features.

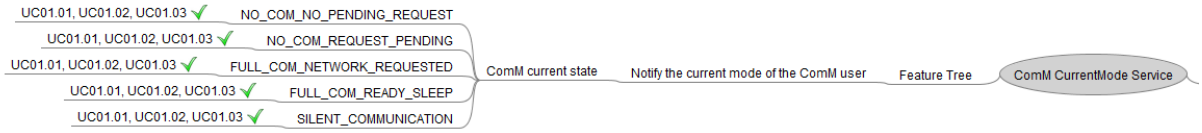
Each test case documents for which releases of the AUTOSAR software specification it can be used:

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

3 RS_BRF_01448 - ComM Current Mode

3.1 General Test Objective and Approach

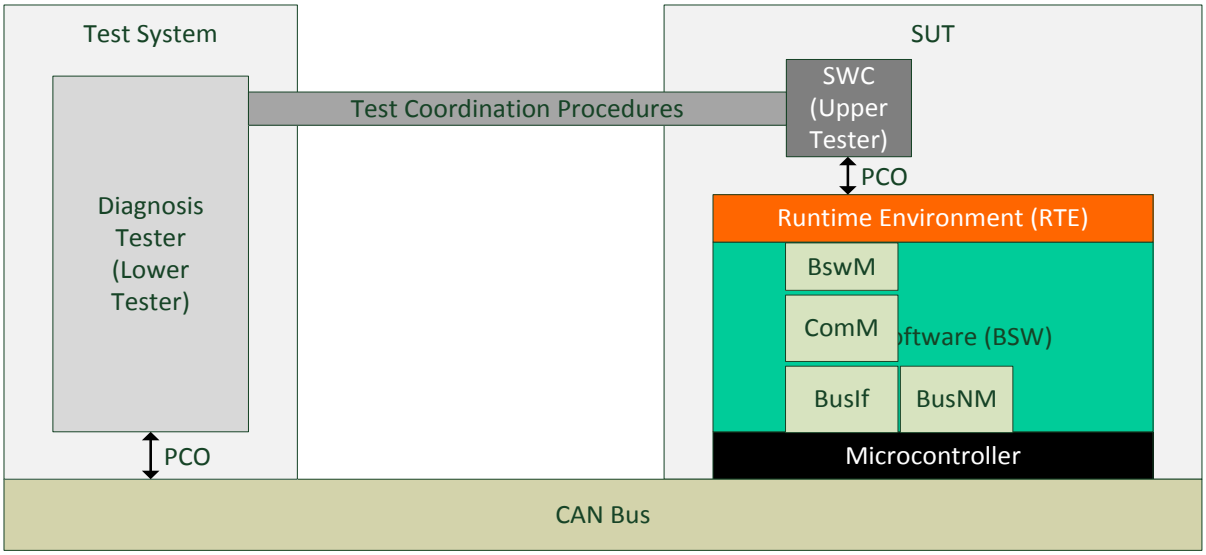
This test specification intends to cover the ComM services features.
This test case document has been established to cover the following features:



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

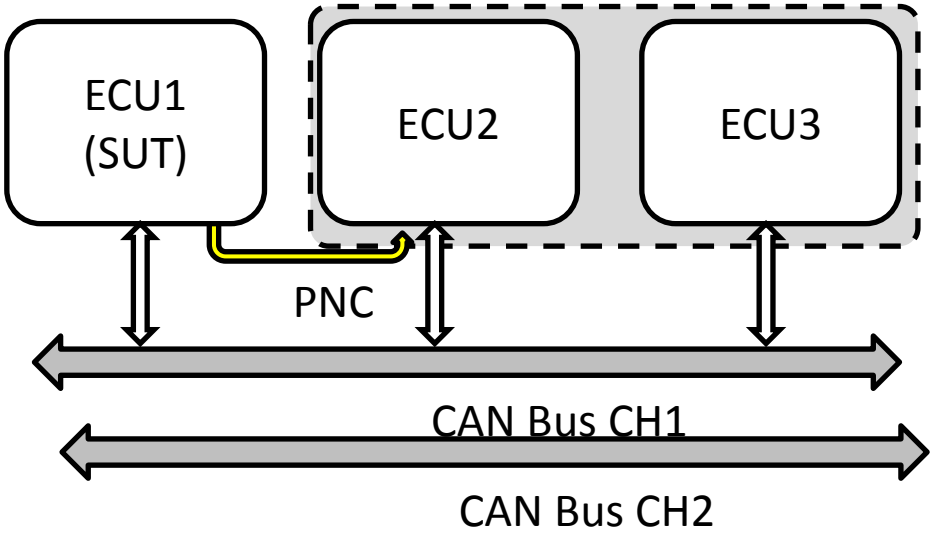
3.1.1.1 Overview on Architecture



The test system architecture consists of testing the two state machine of the ComM: Channel state machine and Pnc state machine.

To perform this, the Test Bench will simulate 2 ECUs and SUT produces either signals to Partial Network Clusters and Full network.

The infrastructure is described below:



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 suite is implemented.

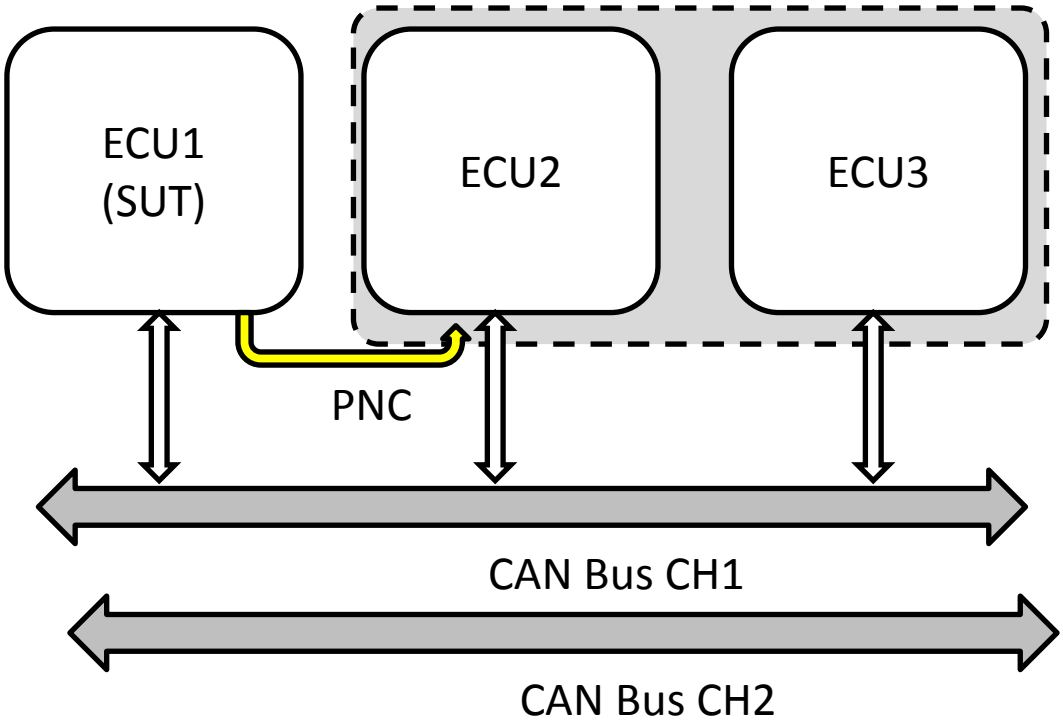
3.1.2.1 Required ECU Extract of System Description Files

Regarding BswM and Mode definitions, the configuration depends on use case:

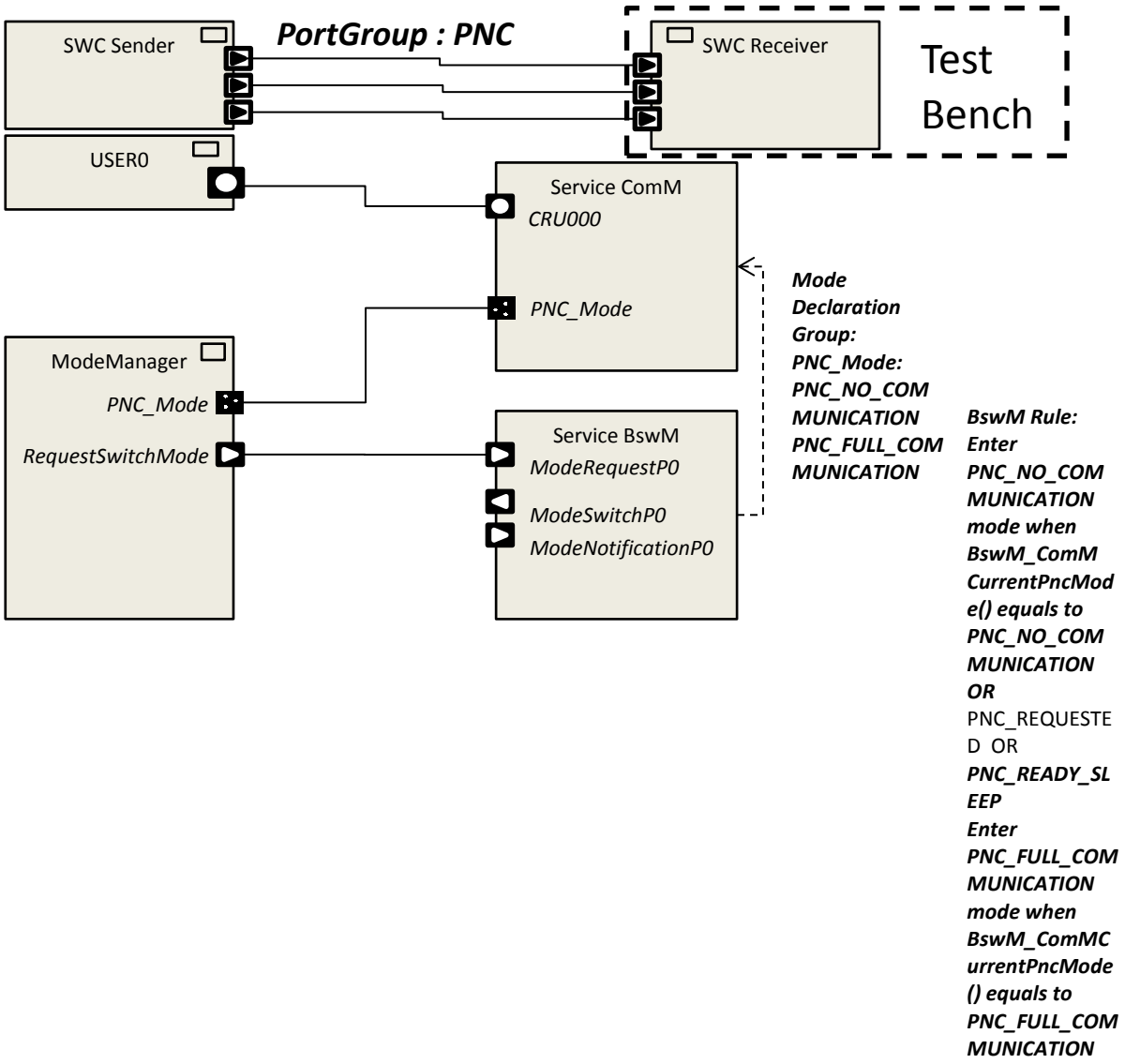
3.1.2.1.1 Use Case 01.01 : One User, One Partial Network, One Channel

The network configuration is described below:

Use case 1 – Partial



The SWC configuration is described below:



The SWC User should be defined as follow:

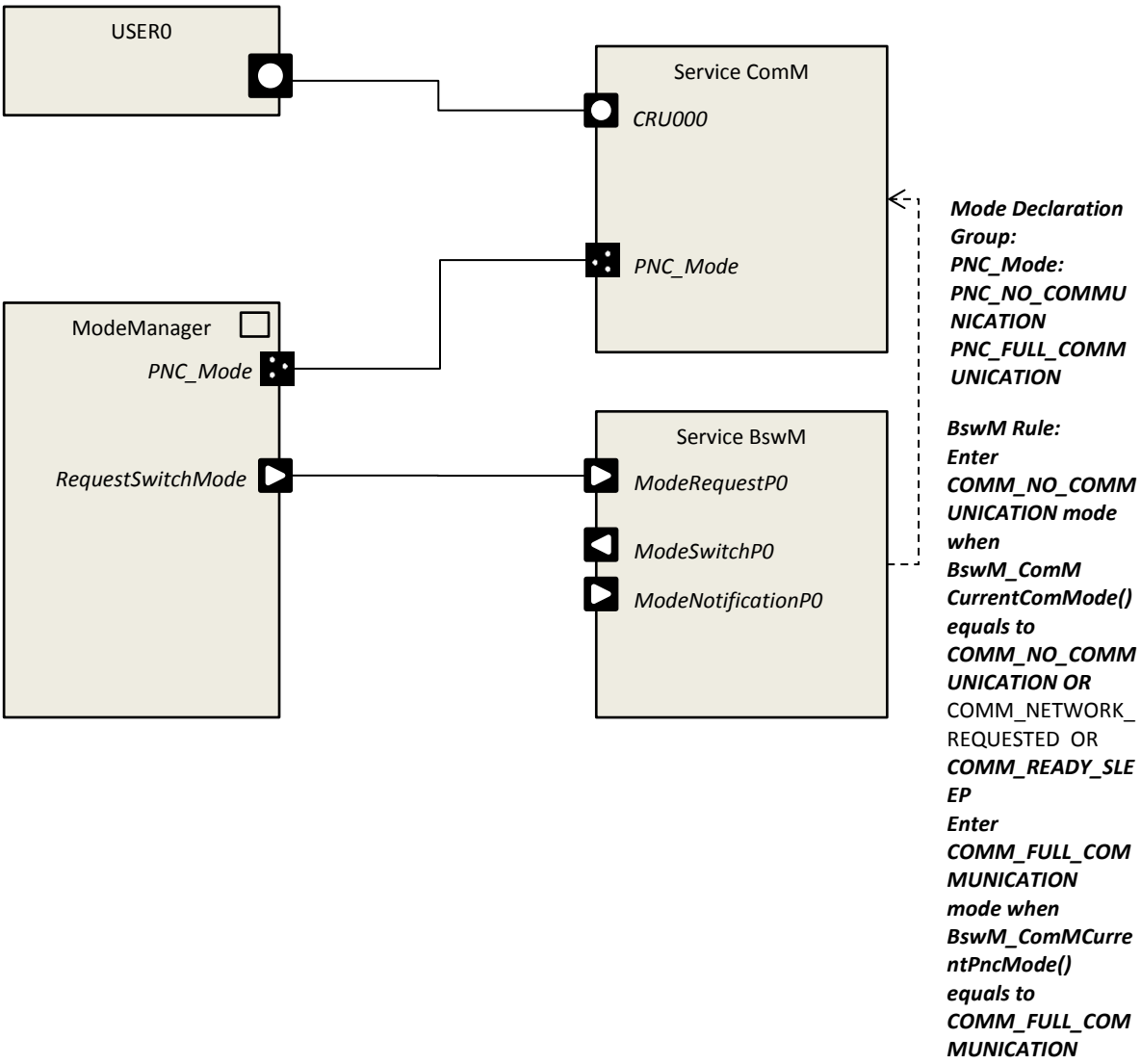
RUN_NO_COM activated by ModeSwitchEvent defined onEntry of Mode PNC_NO_COMMUNICATION

RUN_FULL_COM activated by ModeSwitchEvent defined onEntry of Mode PNC_FULL_COMMUNICATION

RUN_Ctrl: Requests communication to ComM Service.

3.1.2.1.2 Use Case 01.02 : One User, One Channel

The configuration is described below:



The SWC User should be defined as follow:

RUN_NO_COM activated by ModeSwitchEvent defined onEntry of Mode COMM_NO_COMMUNICATION

RUN_FULL_COM activated by ModeSwitchEvent defined onEntry of Mode COMM_FULL_COMMUNICATION

RUN_Ctrl: Requests communication to ComM Service (call to ComM_RequestComMode)

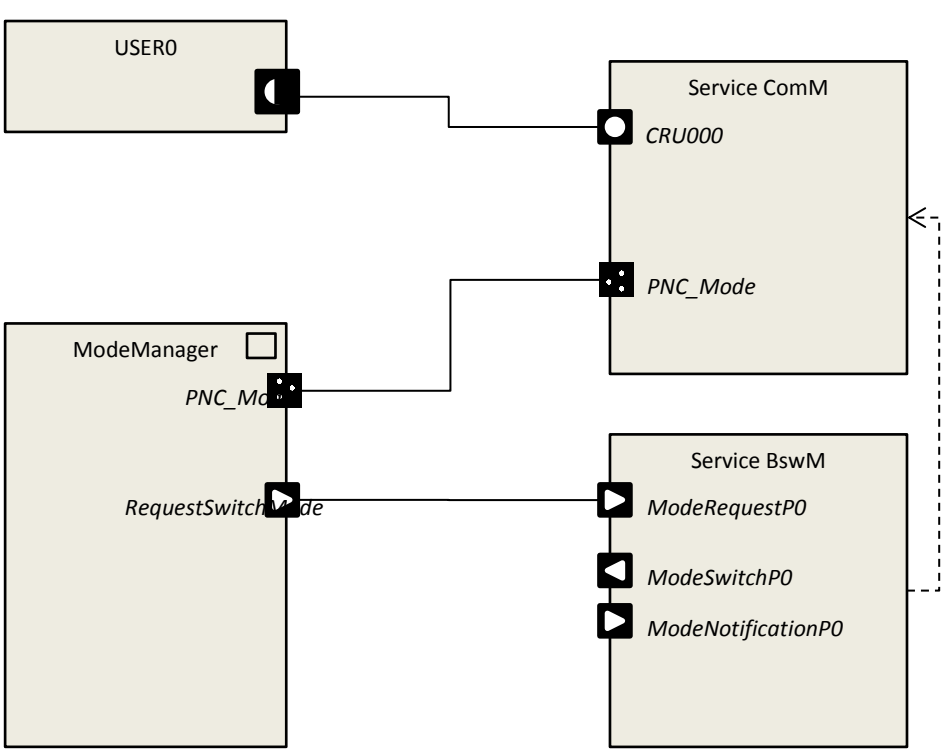
3.1.2.1.3 Use Case 01.03 : One User, 3 Channels

For this use case, the SUT shall be configured to use 3 networks:

- 1 channel in variant FULL
- 1 channel in variant LIGHT
- 1 channel in variant PASSIVE

Same user is configured for the 3 channels.

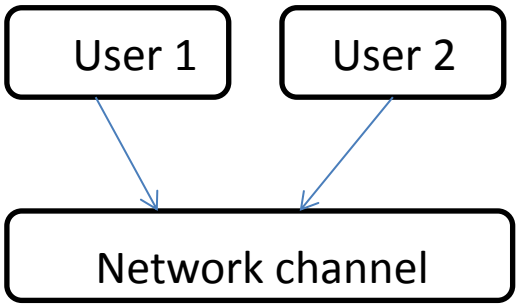
SWC configuration is described below:



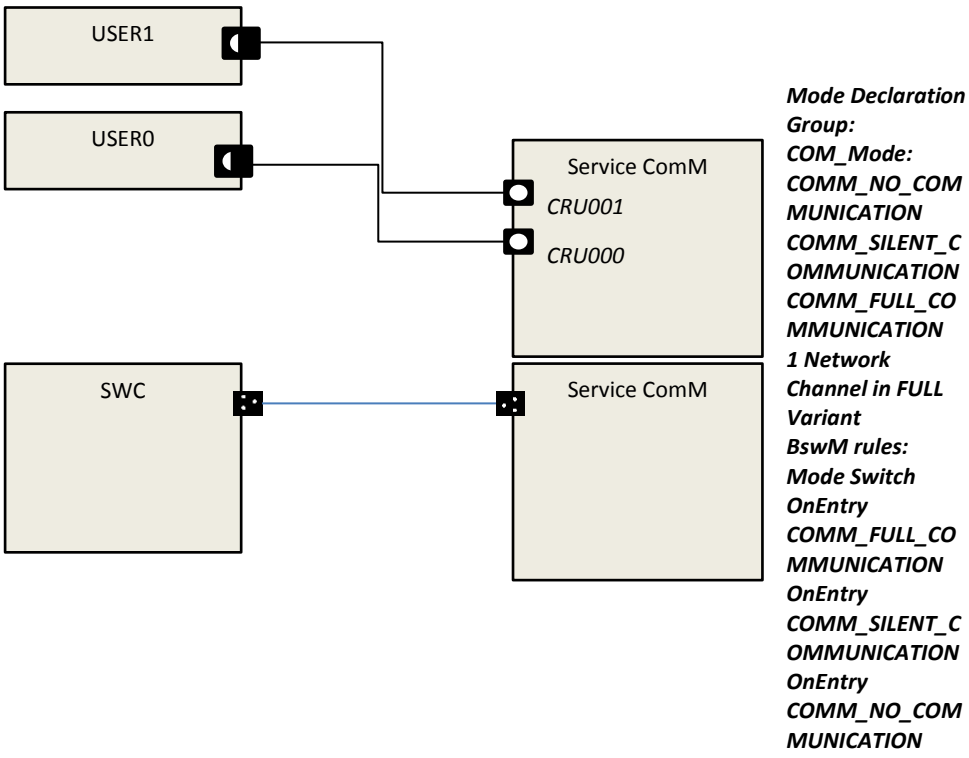
Mode Declaration Group:
PNC_Mode:
PNC_NO_COMMUNICATION
BswM Rule:
Enter PNC_FULL_COMMUNICATION mode when BswM_ComM CurrentComMode() equals to COMM_NO_COMMUNICATION OR COMM_NETWORK_REQUESTED OR ***COMM_READY_SLEEP for the 3 channels***
Enter ALLCHANNELS_FULL_COMMUNICATION mode when BswM_ComMCurrentPncMode() equals to COMM_FULL_COMMUNICATION for the 3 channels
Enter: CHANNEL1&CHANNEL2_FUILL_COMMUNICATION when BswM_CurrentComMode equals to COMM_FULL_COMMUNICATION for Channel1 & channel 2

3.1.2.1.4 Use Case 01.04: Two Users, 1 Channel

For this use case, the configuration focuses on two separate users that request access to the same network channel:



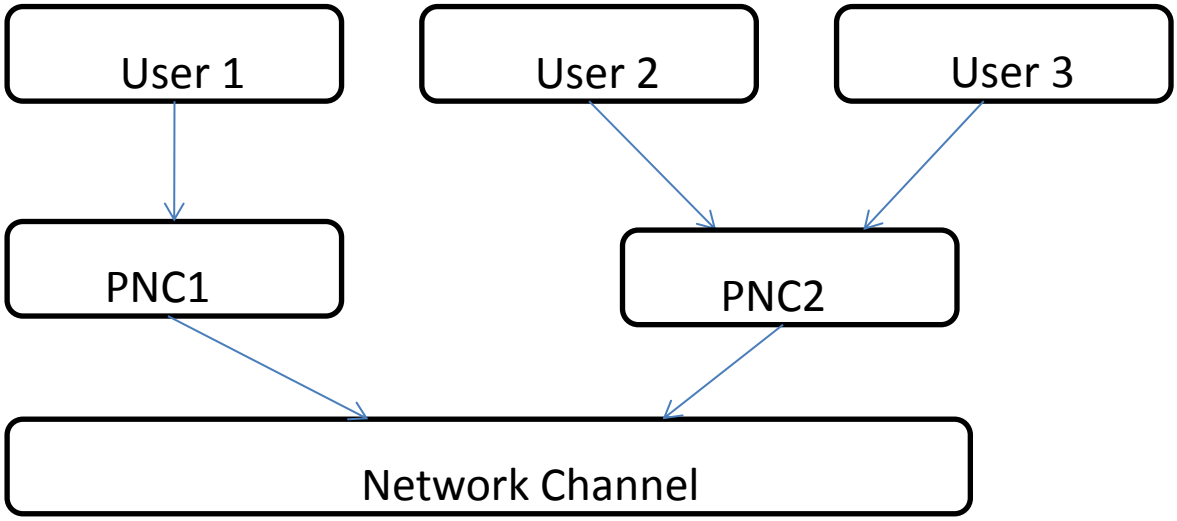
So, the configuration at SWC level is described below:



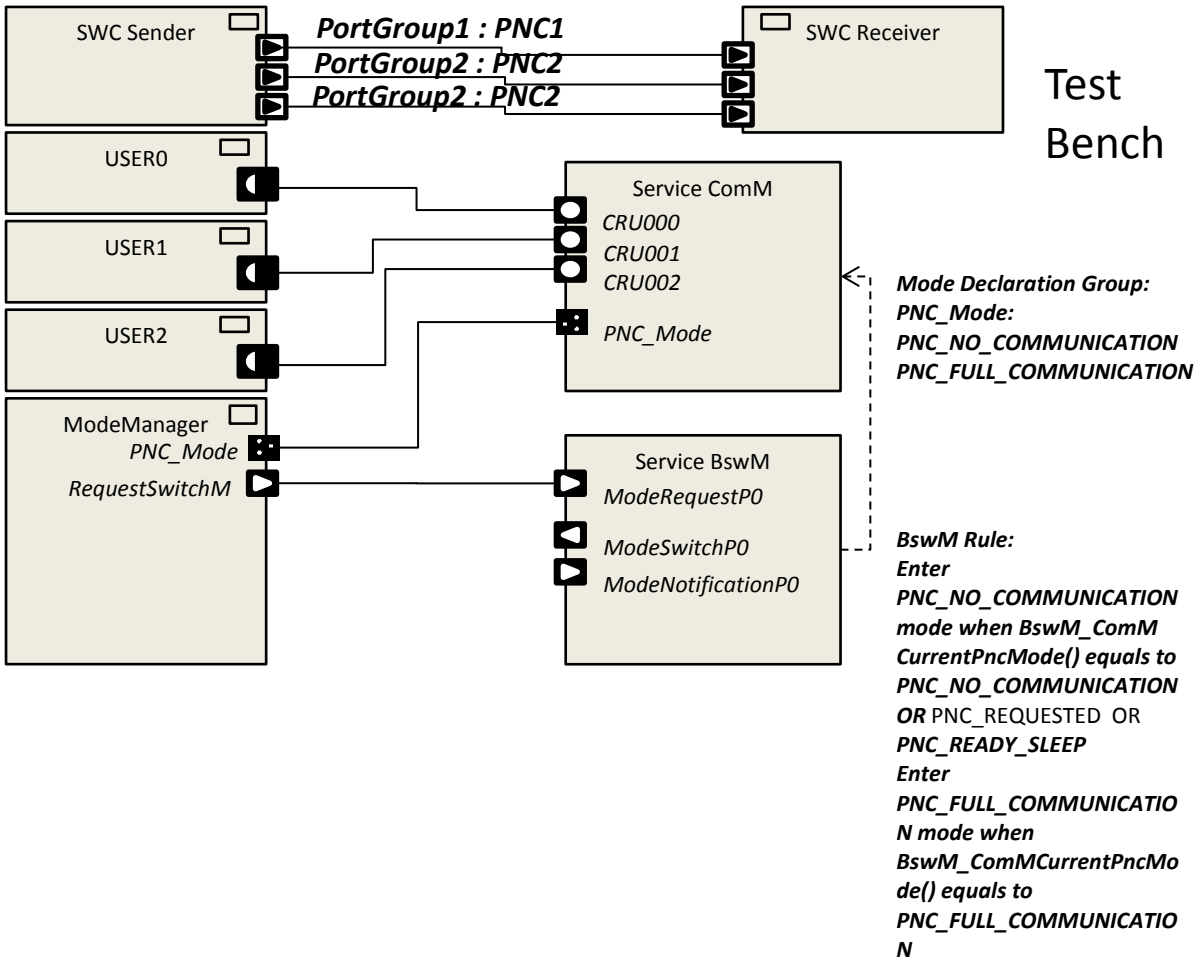
The SWC is notified at each ModeSwitch by ModeSwitchEvents on Entry each ComMode.

3.1.2.1.5 Use Case 01.05 : Three Users, 2 users for 1 PartialNetworkCluster, 1 user for 1 PartialNetworkCluster, 1 Channel

For this use case, the configuration will focus on 3 separate users that request 2 partial network clusters allocated to one specific network channel:



The SWC description used is the following:



For this use case, application messages should be sent periodically.

A SWC is defined with Runnables activated by ModeSwitchEvent:

- OnEntry COMM_NO_COMMUNICATION
- OnEntry COMM_SILENT_COMMUNICATION
- OnEntry COMM_NO_COMMUNICATION

3.1.2.2 Required ECU Configuration Description Files

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

3.1.2.3 Required Software Component Description Files

No specific configuration requirements for Software Components.

3.1.2.4 Mandatory vs. Customizable Parts

In the configuration set, the only mandatory definition is to have at least 2 User, one with PNC, one other without PNC.

3.1.3 Test Case Design

Not Applicable

3.2 Re-usable Test Steps

Not Applicable

3.3 Test Cases

3.3.1 [ATS_COMM_00192] Notifying the current mode of the ComM user when Partial Networking is activated

Test Objective	Notifying the current mode of the ComM user when Partial Networking is activated		
ID	ATS_COMM_00192	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	ComM, BswM	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00111 ATR: ATR_ATR_00038 ATR: ATR_ATR_00114		
Trace to SWS Item	COMManager: SWS_ComM_00847 COMManager: SWS_ComM_00733		
Requirements / Reference to Test Environment	Configuration use case : UC01.01		
Configuration Parameters	ModeDeclarationGroup: PNC_NO_COMMUNICATION PNC_FULL_COMMUNICATION one SWC Mode User: RUN_FULL_COM activated on ModeSwitchEvent (OnEntry mode PNC_FULL_COMMUNICATION) RUN_NO_COM activated on ModeSwitchEvent (OnEntry mode PNC_NO_COMMUNICATION) 1 Network on channel 1 1 Partial Network on Channel 1 BswM rule is configured to generate a mode switch event when PNC change its modes. A mode declaration group is defined: PNC_NO_COMMUNICATION PNC_FULL_COMMUNICATION		
Summary	The aim of this test is to verify that the SUT is able to notify a mode switch when the state of the ComM changes. This test must be done with a SUT configured with the Partial Networking activated (one User, one Partial Network Cluster, one Channel).		
Needed Adaptation to other Releases	None		
Pre-conditions	SUT is started		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[CP] Power on SUT		[LT] BUS is not active
Step 2	[SWC] Call ComM_RequestComMode() to COMM_FULL_COMMUNICATION (with configuration switch ComMPncGatewayEnabled = TRUE, PNC bits within ERAn = 1)		[SWC] The Runnable RUN_FULL_COM should be activated by the ModeSwitchEvent on entering Mode : PNC_FULL_COMMUNICATION

Step 3	[CP] WAIT n seconds (wait for the network to be started - Bus dependent time)	-
Step 4	[SWC] Call ComM_RequestComMode() to COMM_NO_COMMUNICATION	[SWC] The Runnable RUN_NO_COM should be activated by the ModeSwitchEvent OnEntry Mode : PNC_NO_COMMUNICATION
Step 5	[CP] WAIT 24.2 seconds (20s (NM Timeout Timer expiration) + 1s (Wait Bus Sleep Timer expiration) + 2 * 100ms (ComM main function) + 3s (ComMPncPrepareSleepTimer))	-
Step 6	[LT] Start Sending NM Frames corresponding to Partial Network Cluster (PNV bit is set)	[SWC] The Runnable RUN_FULL_COM should be activated by the ModeSwitchEvent OnEntry Mode : PNC_FULL_COMMUNICATION
Post-conditions	None	

3.3.2 [ATS_COMM_00193] Notifying the current mode of the ComM user when ComM is in variant FULL

Test Objective	Notifying the current mode of the ComM user when ComM is in variant FULL		
ID	ATS_COMM_00193	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00111 ATR: ATR_ATR_00038		
Trace to SWS Item	COMManager: SWS_ComM_00847 COMManager: SWS_ComM_00733		
Requirements / Reference to Test Environment	Configuration use case : UC01.02		
Configuration Parameters	ModeDeclarationGroup: COMM_NO_COMMUNICATION COMM_FULL_COMMUNICATION one SWC Mode User: RUN_FULL_COM activated on ModeSwitchEvent (OnEntry mode COMM_FULL_COMMUNICATION) RUN_NO_COM activated on ModeSwitchEvent (OnEntry mode COMM_NO_COMMUNICATION) one channel no Partial Networking		
Summary	The aim of this test is to verify that the SUT is able to notify the RTE when the state of the ComM changes. This test must be done with a SUT configured with		

	NmVariant = FULL.	
Needed Adaptation to other Releases	None	
Pre-conditions	None	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[CP] Power on SUT	-
Step 2	[SWC] Call ComM_RequestComMode() to COMM_FULL_COMMUNICATION	[SWC] The Runnable RUN_FULL_COM should be activated by ModeSwitchEvent OnEntry Mode COMM_FULL_COMMUNICATION
Step 3	[CP] WAIT n seconds (wait for the network to be started - Bus dependent time)	-
Step 4	[SWC] Call ComM_RequestComMode() to COMM_NO_COMMUNICATION	[SWC] The Runnable RUN_NO_COM should be activated by ModeSwitchEvent OnEntry Mode COMM_NO_COMMUNICATION
Step 5	[CP] WAIT 21.2 seconds (20s (NM Timeout Timer expiration) + 1s (Wait Bus Sleep Timer expiration) + 2 * 100ms (ComM main function))	-
Step 6	[LT] Start Sending NM Frames to SUT	[SWC] The Runnable RUN_FULL_COM should be activated by ModeSwitchEvent OnEntry Mode COMM_FULL_COMMUNICATION
Post-conditions	None	

3.3.3 [ATS_COMM_00194] Notifying the current mode of the ComM user for 3 channels / 1 ComM user

Test Objective	Notifying the current mode of the ComM user for 3 channels / 1 ComM user		
ID	ATS_COMM_00194	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	ComM, BswM	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00111 ATR: ATR_ATR_00038		

Trace to SWS Item	COMManager: SWS_ComM_00847 COMManager: SWS_ComM_00733
Requirements / Reference to Test Environment	Configuration use case : UC01.03
Configuration Parameters	No Partial Networks One SWC User: one RUN_FULL_COM activated on ModeSwitchEvent OnEntry COMM_FULL_COMMUNICATION one RUN_NO_COM activated on ModeSwitchEvent OnEntry COMM_NO_COMMUNICATION 3 channels : one channel defined in FULL variant (not ICC1 configuration) one channel defined in LIGHT variant (not ICC1 configuration) one channel defined in PASSIVE variant (not ICC1 configuration)
Summary	The aim of this test is to verify that the SUT is able to notify the RTE when the state of the ComM changes. This test focuses on one ComM User using 3 channels configured in different ComM variants (FULL, PASSIVE, LIGHT).
Needed Adaptation to other Releases	None
Pre-conditions	None

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[CP] Power on SUT	
Step 2	[SWC] Call ComM_RequestComMode() with COMM_FULL_COMMUNICATION	[LT]: Check that channel 1 (FULL variant) transmit periodically Nm Messages Check that Channel 2 and 3 do not transmit Nm Messages Check that Application Messages are visible on the bus level for the 3 channels
Step 3	-	[SWC] Check that Runnable RUN_FULL_COM has been activated by ModeSwitchEvent OnEntry ALL_CHANNELS_FULL_COMMUNICATION
Step 4	[SWC] Call ComM_RequestComMode() to COMM_NO_COMMUNICATION to release the buses	[SWC] Check that SUT has entered the COMM_FULL_COM_READY_SLEEP state for each channel
Step 5	-	[LT] Check that all channels stop to send NM frames Check that periodic application messages sent by SUT are visible on the bus level for the 3 channels
Step 6	[CP] Wait <ComMTMinFullComModeDuration> time (delay configured to force	[SWC] (check that channel in LIGHT variant enters the COMM_NO_COMMUNICATION state after

	channel to keep the FULL_COMMUNICATION after no active network requests)	expiration on ComMTMinFullComModeDuration timer expires) Check that CHANNEL_2 is in COMM_NO_COMMUNICATION & CHANNEL_1 & CHANNEL_3 are in COMM_FULL_COMMUNICATION
Step 7	-	[LT] Check that periodic application messages are still present only for Channel in FULL Variant and Channel in PASSIVE Variant
Step 8	[LT] Stop sending Nm Messages for 3 channels	[SWC] Check that Channel in FULL variant and channel in PASSIVE variants are in COMM_SILENT_COMMUNICATION state.
Step 9	[CP] Wait MAX (< configured Nm Timeout >)	[SWC] (check that PASSIVE and FULL variant channels enter in COMM_NO_COM_NO_PENDING_REQUEST state) Check that CHANNEL_1, CHANNEL_2 and CHANNEL_3 are in COMM_NO_COMMUNICATION state.
Step 10	-	[LT] Check that periodic application messages are NOT present for any channel
Step 11	[LT] Start sending NM Frames on 3 channels	[SWC] (check that PASSIVE and FULL variant channels enters in COMM_FULL_COM_NETWORK_REQUESTED state) Check that Runnable RUN_CH1_CH2_FULL_COM is activated by ModeSwitch Event OnEntry CHANNEL1&CHANNEL3_FULL_COMMUNICATION (Mode entered when both Channel1 & Channel3 in FULL_COMMUNICATION)
Step 12	-	[LT] Check that Application Messages are visible on the bus level for the channel 1 & Channel 3
Post-conditions	None	

3.3.4 [ATS_COMM_00267] Notifying the current COM mode, 2 users for one channel

Test Objective	Notifying the current COM mode, 2 users for one channel		
ID	ATS_COMM_00267	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	Com, ComM	**State**	reviewed

Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00111 ATR: ATR_ATR_00038
Trace to SWS Item	COMManager: SWS_ComM_00995 COMManager: SWS_ComM_00051 COMManager: SWS_ComM_00073 COMManager: SWS_ComM_00869 COMManager: SWS_ComM_00888
Requirements / Reference to Test Environment	Configuration use case : UC01.04
Configuration Parameters	2 SWC ComM Users 2 users allocated to the same network channel
Summary	This test case validates the notification of mode changes and com requests when 2 users are connected to the same network channel. Test sequence will request to go in FULL_COMMUNICATION, the test case will check that the second user will not change the communication for the channel (as it is already in FULL_COMMUNICATION state). Then, the test case request to go in NO_COMMUNICATION state for the 2 users. The SUT shall enter in NO_COMMUNICATION after both users have request to go in NO_COMMUNICATION.
Needed Adaptation to other Releases	None
Pre-conditions	SUT is not started

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[CP] Power ON ECU	
Step 2	[SWC<User1>] Call ComM_RequestComMode(<user1>, COMM_FULL_COMMUNICATION)	-
Step 3	[SWC<User1>] Call ComM_GetCurrentComMode()	[SWC] Check that runnable called on ModeSwitchEvent OnEntry COMM_FULL_COMMUNICATION is called Check that CurrentComMode equals to COMM_FULL_COMMUNICATION
Step 4	-	[LT] Check that Nm Messages are sent by SUT
Step 5	[SWC<User2>] Call ComM_RequestComMode(<user2>, COMM_FULL_COMMUNICATION)	
Step 6	[SWC<User2>] Call ComM_GetCurrentComMode()	[SWC] Check that CurrentComMode equals to COMM_FULL_COMMUNICATION
Step 7	-	[LT] Check that Nm Messages are sent by SUT

Step 8	[SWC<User1>] Call ComM_RequestComMode(<user1>, COMM_NO_COMMUNICATION)	-
Step 9	[SWC<User1>] Call ComM_GetCurrentComMode()	[SWC] Check that CurrentComMode equals to COMM_FULL_COMMUNICATION (as network is still requested by user2)
Step 10	-	[LT] Check that Nm Messages are sent by SUT
Step 11	[SWC<User2>] Call ComM_RequestComMode(<user2>, COMM_NO_COMMUNICATION)	-
Step 12	[SWC<User2>] Call to ComM_GetCurrentComMode()	[SWC] Check that runnable activated on ModeSwitchEvent OnEntry COMM_NO_COMMUNICATION is called Check that CurrentComMode equals to COMM_NO_COMMUNICATION
Step 13	-	[LT] Check that NO Nm Messages are sent by SUT
Post-conditions	None	

3.3.5 [ATS_COMM_00268] Notifying the current COM mode, 3 users, 2 Partial Networks, 1 channel

Test Objective	Notifying the current COM mode, 3 users, 2 Partial Networks, 1 channel		
ID	ATS_COMM_00268	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	Rte, Com, ComM, BswM	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00111 ATR: ATR_ATR_00038		
Trace to SWS Item	COMManager: SWS_ComM_00995 COMManager: SWS_ComM_00051 COMManager: SWS_ComM_00073 COMManager: SWS_ComM_00869 COMManager: SWS_ComM_00888		
Requirements / Reference to Test Environment	Configuration use case : UC01.05		
Configuration Parameters	1 Swc User 1 allocated to a partial network cluster pnc1 1 Swc User 2 and User 3 allocated to a partial network cluster pnc2 Application messages are sent periodically on PortGroup1 (for PartialNetwork1) Application messages are sent periodically on PortGroup2 (for PartialNetwork2)		

	Timeout values shall be different from 0 to be able to detect the mode transitions
Summary	This test case validates network requests and associated notifications for 3 users and 2 partial networks on the same channel. The main goal of this test consist in request to go in full communication for each partial networks users (pnc are allocated to the same physical bus), then request each user to go in NO_COMMUNICATION. The test shall demonstrate that Physical channel go in NO_COMMUNICATION only when all partial networks allocated to the physical channel are released.
Needed Adaptation to other Releases	None
Pre-conditions	SUT is powered off

Main Test Execution

Test Steps		**Pass Criteria**
Step 1	[CP] Power ON ECU	
Step 2	[SWC<USER1>] (for PNC1): Call to ComM_RequestComMode(<user1_pnc1>, COMM_FULL_COMMUNICATION)	[SWC] Check that runnable activated on ModeSwitchEvent OnEntry COMM_FULL_COMMUNICATION is called
Step 3	-	[LT] Check that Nm Messages with PartialNetworkVector = PNC1 is sent by SUT Check that application periodic messages of PortGroup1 (mapped on PNC1) are sent
Step 4	[SWC] Call to ComM_GetCurrentComMode(<user1_pnc1>)	[SWC] Check that CurrentComMode equals to COMM_FULL_COMMUNICATION
Step 5	[SWC<USER2>] (for PNC2): Call to ComM_RequestComMode(<user2_pnc2>, COMM_FULL_COMMUNICATION)	[LT] Check that Nm Messages with PartialNetworkVector = PNC1&PNC2 is sent by SUT Check that application periodic messages of PortGroup1 (mapped on PNC1) & PortGroup2 (mapped on PNC2) are sent
Step 6	[SWC<USER3>] (for PNC2): Call to ComM_RequestComMode(<user3_pnc2>, COMM_FULL_COMMUNICATION)	[LT] Check that Nm Messages with PartialNetworkVector = PNC1&PNC2 is sent by SUT Check that application periodic messages of PortGroup1 (mapped on PNC1) & PortGroup2 (mapped on PNC2) are sent
Step 7	[SWC<USER2>] (for PNC2):Call to ComM_RequestComMode(<user2_pnc2>, COMM_NO_COMMUNICATION)	[LT] Check that Nm Messages with PartialNetworkVector = PNC1&PNC2 is sent by SUT Check that application periodic messages of PortGroup1 (mapped on

		PNC1) & PortGroup2 (mapped on PNC2) are sent
Step 8	[SWC<USER1>] (for PNC1): Call to ComM_RequestComMode(<user1_pnc1>, COMM_NO_COMMUNICATION)	[LT] Check that Nm Messages with PartialNetworkVector = PNC2 is sent by SUT Check that application periodic messages of PortGroup1 (mapped on PNC1) & PortGroup2 (mapped on PNC2) are sent
Step 9	[SWC] Call to ComM_GetCurrentComMode(<user1_pnc1>)	[SWC] Check that ComMCurrentComMode equals to COMM_FULL_COMMUNICATION
Step 10	[CP] Wait (<NmTimeoutTime> configured in BusNm Module)	[LT] Check that Nm Messages with PartialNetworkVector = PNC2 is sent by SUT Check that application periodic messages of PortGroup2 (mapped on PNC2) are sent
Step 11	[SWC<USER3>] (for PNC2): Call to ComM_RequestComMode(<user3_pnc2>, COMM_NO_COMMUNICATION)	[LT] Check that No Nm Messages are sent by the SUT Check that application periodic messages of PortGroup2 (mapped on PNC2) are sent
Step 12	-	[SWC] Checks that runnable activated on ModeSwitchEvent OnEntry mode COMM_SILENT_COMMUNICATION is called.
Step 13	[SWC] Call ComM_GetCurrentComMode(<user3_pnc2>)	[SWC] Checks that CurrentComMode equals to COMM_SILENT_COMMUNICATION
Step 14	[CP] Wait (<NmWaitBusSleepTime> configured in BusNm Module)	[LT] Check that No Nm Messages are sent by SUT Check that no application periodic messages are sent
Step 15	-	[SWC] Checks that runnable activated on ModeSwitchEvent OnEntry mode COMM_NO_COMMUNICATION is called.
Post-conditions	None	

3.3.6 [ATS_COMM_00771] Different Users Mapped To Same Channel Which Is In FULL Communication and All Users Requesting For NO Communication

Test Objective	Different Users Mapped To Same Channel Which Is In FULL Communication and All Users Requesting For NO Communication

ID	ATS_COMM_00771	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00686		
Requirements / Reference to Test Environment	none		
Configuration Parameters	ComM user1 = SWC<1> Channel 0 ComM user2 = SWC<2> Channel1 LinChannelId = Ex: LinChannelId 0 CANChannelId = Ex: CANChannelId 1		
Summary	If two different users are mapped to same channel then the status of the channel shall be the highest communication mode.		
Needed Adaptation to other Releases			
Pre-conditions	Channel 0 and 1 are in Full Communication and message transmission is normal		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[SWC] Request ComM to be in No Communication mode		[SWC] RTE_E_OK should be returned
Step 2	[SWC] Request ComM to be in No Communication mode		[SWC] RTE_E_OK shall be returned
Step 3	[SWC] Request ComM for the current mode for the associated channel		[SWC] COMM_NO_COMMUNICATION shall be returned
Step 4	[SWC] Request ComM for the current mode for the associated channel		[SWC] COMM_NO_COMMUNICATION shall be returned
Step 5	-		[LT] No messages shall be observed on CAN and LIN channels
Post-conditions	NONE		

3.3.7 [ATS_COMM_00772] FULL Communication Mode Request By Diagnostic Module

Test Objective	FULL Communication Mode Request By Diagnostic Module		
ID	ATS_COMM_00772	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed

Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00866		
Requirements / Reference to Test Environment	none		
Configuration Parameters	User defined configuration reference: CANChannel= Ex:1 Transmission frame: Data= 0x02 0x10 0x01 0x00 0x00 0x00 0x00 and 0x00 CANId= Ex: 0x200 and 0x201 Response frame: data=0x02 0x50 0x01 CANId= Ex: 0x200 and 0x201		
Summary	When a diagnostic message is received the DCM module will request ComM_DCM_ActiveDiagnostic of ComM. Then ComM will change the status of that channel to FULL communication if communication can be allowed. Application message can be observed on the bus.		
Needed Adaptation to other Releases			
Pre-conditions	DUT shall be initialized		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[LT] Transmit a valid diagnostic session request to the DUT		[LT] Positive response for the request shall be observed on the bus
Step 2	[SWC] Check the current ComM mode for the channel		[SWC] COMM_FULL_COMMUNICATION shall be returned
Post-conditions	NONE		

3.3.8 [ATS_COMM_00773] Behaviour of DUT If Silent Communication Mode Is Requested By SW-C

Test Objective	Behaviour of DUT If Silent Communication Mode Is Requested By SW-C		
ID	ATS_COMM_00773	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	BswM, ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	BSWModeManager: SWS_BswM_00038 COMManager: SWS_ComM_00151		

Requirements / Reference to Test Environment	none	
Configuration Parameters		
Summary	COMM_SILENT_COMMUNICATION is an internal state of ComM which shall not be requested by the user. If in case user requests for COMM_SILENT_COMMUNICATION mode ComM will report development error to DET.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Request ComM to switch to SILENT communication mode	[SWC] RTE_E_NOT_OK shall be returned
Step 2	-	[CP] If DET is enabled COMM_E_WRONG_PARAMETERS shall be reported to DET
Post-conditions	NONE	

3.3.9 [ATS_COMM_00812] At Least One ComM User Assigned To The PNC Requests “Full Communication”

Test Objective	At Least One ComM User Assigned To The PNC Requests “Full Communication”		
ID	ATS_COMM_00812	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00932 COMManager: SWS_ComM_00993		
Requirements / Reference to Test Environment	none		
Configuration Parameters	COMM_USER = 1 COMM_PNC_ENABLED = TRUE		
Summary	Configure SWC0 and SWC1 (User 0 and User 1) to the same PNC which is mapped to ComM Channel 0 and one RX and TX Ipdu-Group is mapped to this particular PNC. In NM Frame 2nd bytes first bit is mapped to this particular PNC.		
Needed Adaptation to other Releases			

Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Send the request for changing COM mode to Full communication for User0.	[SWC] RTE shall return E_OK
Step 2	[SWC] Send the request for the current mode of ComM for User0.	[SWC] ComM mode shall be in Full communication
Step 3	[SWC] Send the request for changing COM mode to No communication for User0.	[SWC] RTE shall return E_OK BswM user callout shall be invoked for transition to PNC_PREPARE_SLEEP State
Step 4	[SWC] Check the PNC state	[SWC] PNC shall be in PNC_READY_SLEEP State
Step 5	[SWC] Send the request for changing COM mode to Full communication for User1.	[SWC] RTE shall return E_OK BswM user callout shall be invoked for transition to PNC_REQUESTED State
Post-conditions	NONE	

3.3.10 [ATS_COMM_00813] Notification Of State Changes Of PNC To BswM From ComM As PNC Prepare Sleep, When Bit Representing This PNC In EIRA Changes To Zero

Test Objective	Notification Of State Changes Of PNC To BswM From ComM As PNC Prepare Sleep, When Bit Representing This PNC In EIRA Changes To Zero		
ID	ATS_COMM_00813	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00940		
Requirements / Reference to Test Environment	none		
Configuration Parameters	COMM_PNC_ENABLED = TRUE ComMPncComSignalKind = EIRA		
Summary	Configure SWC0 with one PNC which is mapped to ComM Channel 0 and one RX and TX Ipdu-Group is mapped to this particular PNC. In NM Frame 2 bytes first bit is mapped to this particular PNC. When User request NO COMMUNICATION, ComM mode shall change from PNC_REQUESTED to PNC_READY_SLEEP state and then from		

	PNC_READY_SLEEP to PNC_PREPARE_SLEEP thereby ComM will intimate BswM about state changes.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Send the request for changing COM mode to Full communication for User0.	[SWC] RTE shall return E_OK
Step 2	[SWC] Send the request for the current mode of ComM for User0.	[SWC] ComM mode shall be in Full communication
Step 3	[SWC] Send the request for changing COM mode to No communication for User0.	[SWC] RTE shall return E_OK BswM user callout shall be invoked for transition to PNC_PREPARE_SLEEP State
Step 4	[SWC] Check the PNC state	[SWC] PNC shall be in PNC_READY_SLEEP State
Post-conditions	NONE	

3.3.11 [ATS_COMM_00814] Behaviour Of The PNC State Machine When It Is In PNC_NO_COMMUNICATION And Parameter ComMPncGatewayEnabled = True

Test Objective	Behaviour Of The PNC State Machine When It Is In PNC_NO_COMMUNICATION And Parameter ComMPncGatewayEnabled = True		
ID	ATS_COMM_00814	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00934		
Requirements / Reference to Test Environment	none		
Configuration Parameters	COMM_PNC_GW_ENABLED = TRUE		
Summary	Stack is configured as Passive Mode. Configure SWC0 with one PNC which is mapped to ComM Channel 0 and one RX and TX Ipdu-Group is mapped to this particular PNC. In NM Frame 2 bytes first bit is mapped to this particular PNC.		

Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Request ComM for the current mode	[SWC] ComM shall be in No communication mode
Step 2	[LT] Send a valid frame from Tester to DUT(The DUT shall wake up passively)	-
Step 3	[LT] Send Nm message with 2nd byte's first bit is set periodically to DUT	[SWC] User Callout shall be invoked on ComM current Pnc state notification indicating PNC_REQUESTED
Post-conditions	NONE	

3.3.12 [ATS_COMM_00817] Behaviour Of The ComM Channel State Machine When More Than One Channel Is Linked To One User Request And The Modes Of The Channels Are Different

Test Objective	Behaviour Of The ComM Channel State Machine When More Than One Channel Is Linked To One User Request And The Modes Of The Channels Are Different		
ID	ATS_COMM_00817	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00995 COMManager: SWS_ComM_00085 COMManager: SWS_ComM_00374 COMManager: SWS_ComM_00176 COMManager: SWS_ComM_00663		
Requirements / Reference to Test Environment	none		
Configuration Parameters	COMM_USER = 1 COMM_CHANNEL_ID = 1		
Summary	Configure two users and two channels. Both the channels are mapped to one of the user. And only one channel is mapped to the other user. Send the No Communication request to the user which is mapped to both the channels. Send the Full communication request to the other user. Now send the request to the user (which is mapped to both the channels) for getting maximum allowed communication mode. The response for that request will be No Communication.		
Needed Adaptation to other Releases			

Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Send the request for changing COM mode to No communication for user1	[SWC] RTE shall return E_OK
Step 2	[SWC] Send the request for changing COM mode to Full communication for user0	[SWC] RTE shall return E_OK
Step 3	[SWC] Send the request for the current mode of ComM for user1	[SWC] ComM shall be in No communication mode
Step 4	[SWC] Send the request to query the maximum allowed Communication Mode of User1	[SWC] ComM shall be in No communication mode
Step 5	-	[LT] Frames shall not be observed
Post-conditions	NONE	

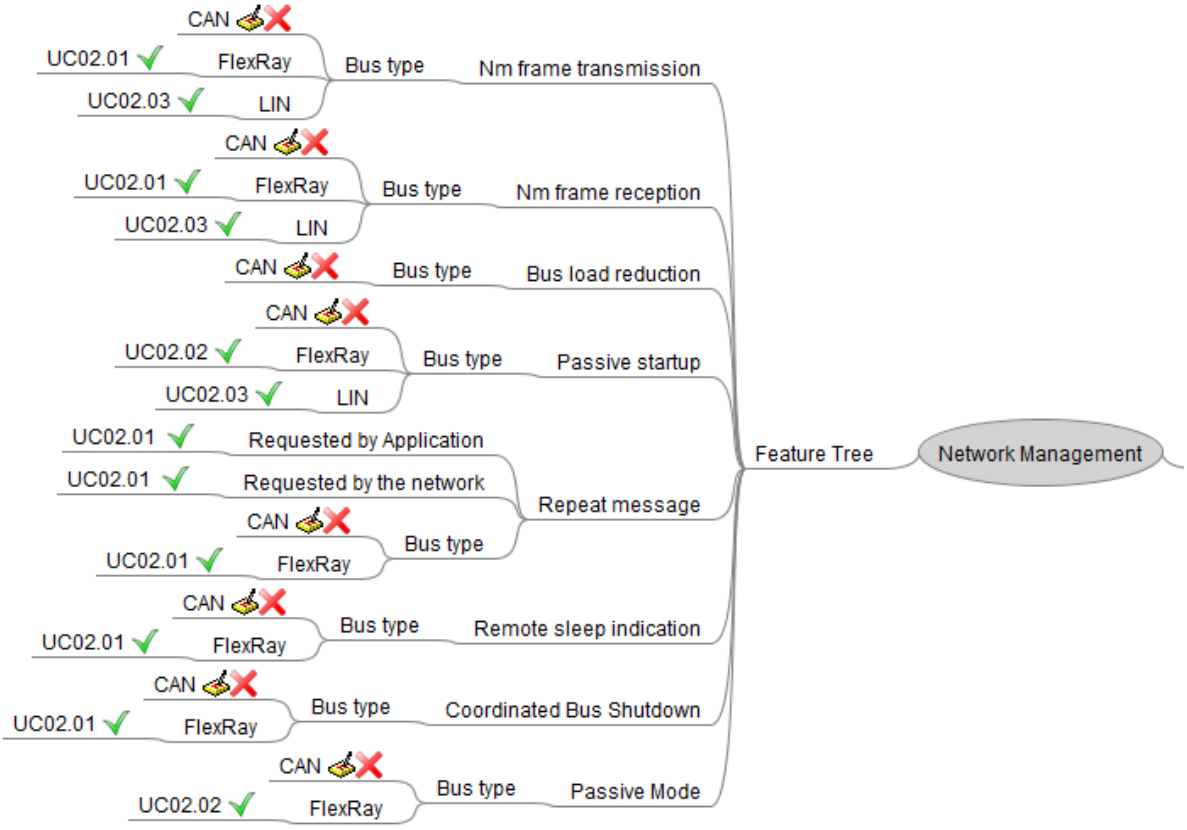
4 RS_BRF_01680 - Network management

4.1 General Test Objective and Approach

This Test Specification intends to cover the Network Management feature as described in the AUTOSAR Feature [RS_BRF_01680].

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:



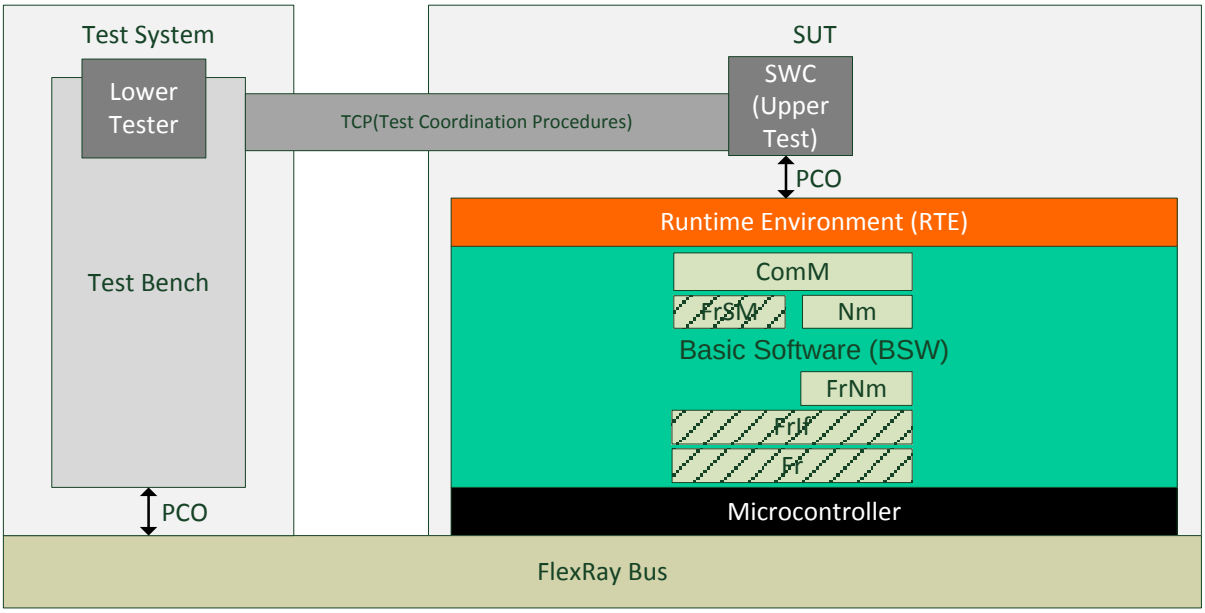
This specification gives the description of required tests environments (test bench, uses case, arxml files) and detailed tests cases for executing tests.

4.1.1 Test System

4.1.1.1 Overview on Architecture

4.1.1.1.1 Use case 02.01: FlexRay bus with SUT in Active Node

For this use case, the aim is to test the network management feature on a FlexRay bus when the SUT is an Active Node.



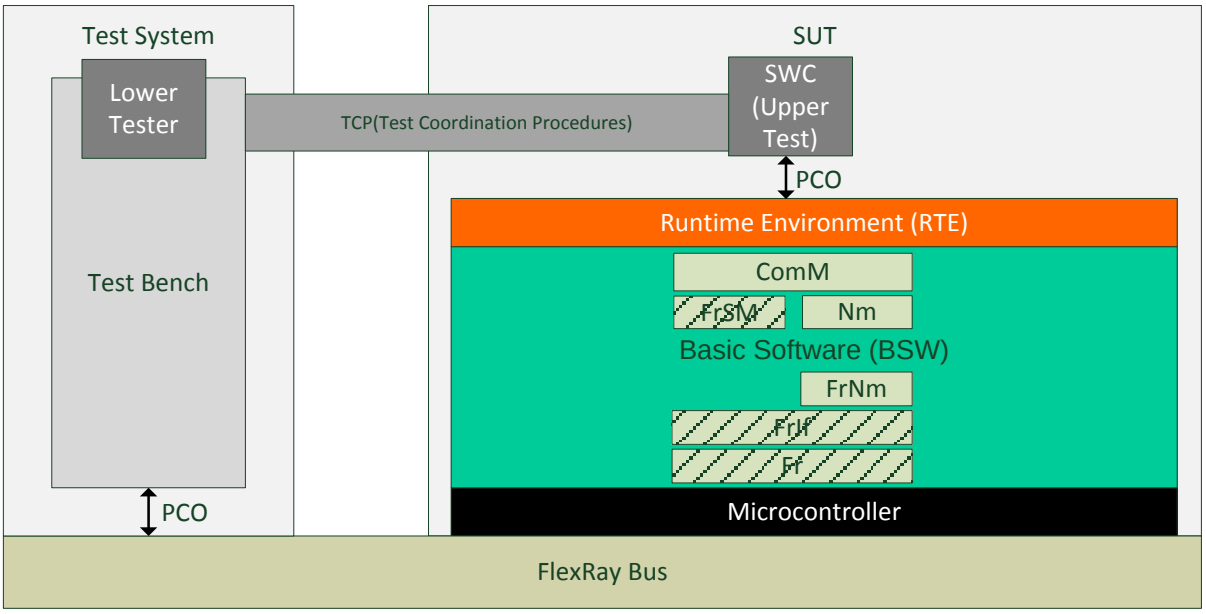
The test system architecture consists of Test Bench that executes only test sequencer and gives actions request through Test coordination Procedures to embedded SWC.

Three modules are only used for test but they are not in the scope of the tests:

- FrSM: Used to start and stop the FlexRay bus
- FrIf: Used to transmit the PDUs over the FlexRay bus
- Fr: Used to drive the FlexRay controller

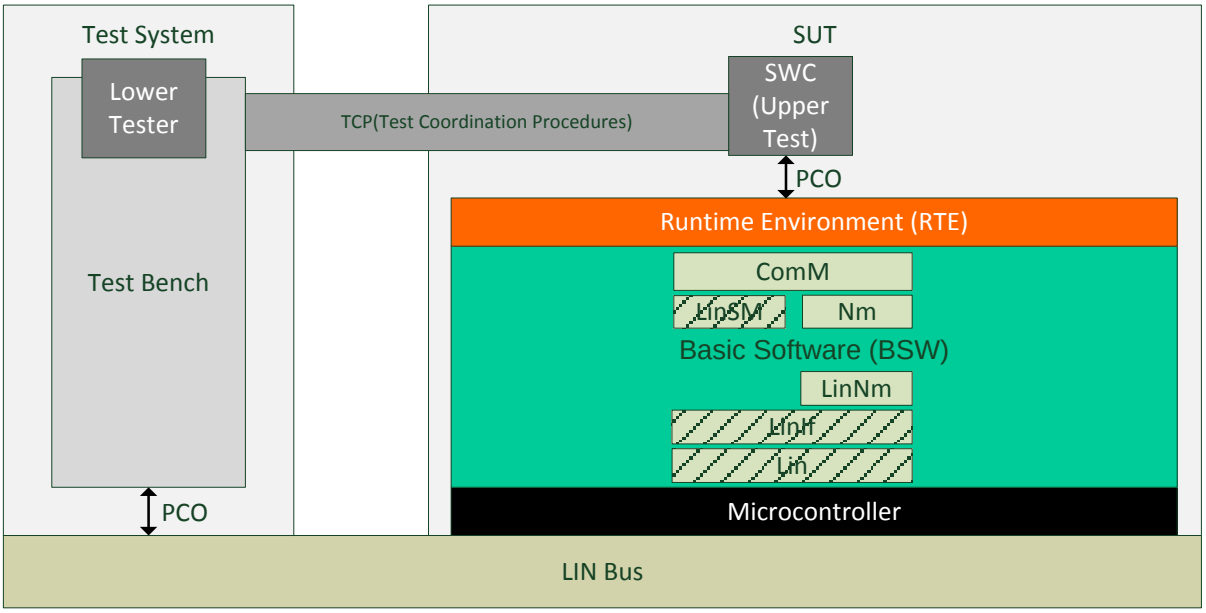
4.1.1.1.2 Use case 02.02: FlexRay bus with SUT in Passive Node

For this use case, the aim is to test the network management feature on a FlexRay bus when the SUT is a Passive Node.



4.1.1.1.3 Use case 02.03: LIN bus

For this use case, the aim is to test the network management feature on a LIN bus when the SUT is an Active Node.



Three modules are only used for test but they are not in the scope of the tests:

- LinSM: Used to start and stop the LIN bus
- LinIf: Used to transmit the PDUs over the LIN bus
- Lin: Used to drive the LIN controller

4.1.1.2 Specific Requirements

Not Applicable

4.1.1.3 Test Coordination Requirements

Not Applicable

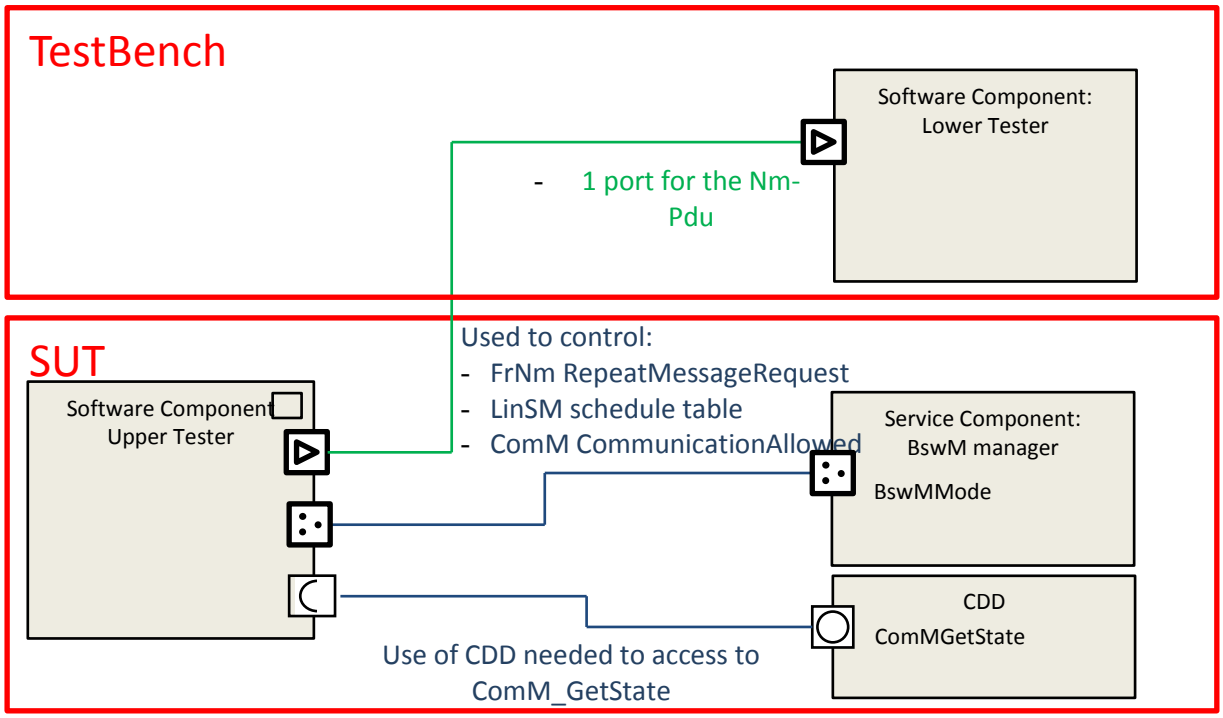
4.1.2 Test Configuration

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

For each use case, the same configuration set can be used. The only difference will be on the mapping of the Nm-Data signal on the bus.

The SWC description is defined below:



4.1.2.1.1 FlexRay Specific Tests

See specific Tests cases configuration parameters description.

4.1.2.1.2 LIN Specific Tests

The LIN Specific tests require 2 Schedule tables with the following properties:

LIN_ScheduleTable1:

General Properties	
Resume Position	START-FROM-BEGINNNING
Run Mode	RUN-CONTINUOUS
Position	**Assigned Frame**
0	FRAME1
1	FRAME2

LIN_ScheduleTable2:

General Properties	
Resume Position	START-FROM-BEGINNNING
Run Mode	RUN-CONTINUOUS
Position	**Assigned Frame**
0	FRAME1
1	FRAME2
2	FRAME3

The Bsw shall be configured with ModeDeclarationGroup as follow:

LIN_SCHEDULE_TABLE1_MODE -> Request Action BswMLinScheduleSwitch to LIN_ScheduleTable1
LIN_SCHEDULE_TABLE2_MODE -> Request Action BswMLinScheduleSwitch to LIN_ScheduleTable2

4.1.2.2 Required ECU Configuration Description Files

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

4.1.2.3 Required Software Component Description Files

No specific configuration requirements for Software Components.

4.1.2.4 Mandatory vs. Customizable Parts

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

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

- The different timeouts
- The activation of the Source node identifier
- The activation of the Car Wakeup feature
- CAN, LIN and FlexRay 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_COMM_00147] Sending "positive" Nm vote periodically when the network is requested by the application

Test Objective	Sending "positive" Nm vote periodically when the network is requested by the application		
ID	ATS_COMM_00147	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	FrNm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00123		
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00107 FlexRayNetworkManagement: SWS_FrNm_00116 FlexRayNetworkManagement: SWS_FrNm_00121 FlexRayNetworkManagement: SWS_FrNm_00123		
Requirements / Reference to Test Environment	Configuration use case : UC02.01		
Configuration Parameters	FrNmGlobalConfig(NmConfig.FlexrayNmCluster): - FrNmPassiveModeEnabled(NmEcu.nmPassiveModeEnabled) = FALSE - FrNmComUserDataSupport(NmEcu.nmUserDataEnabled and SenderReceiverToSignalMapping should exist on associated NmPdu) = TRUE - FrNmUserDataEnabled(NmEcu.nmUserDataEnabled) = TRUE FrNmChannel(FlexrayNmCluster): Bus FR1 - FrNmPduScheduleVariant(FlexrayNmClusterCoupling.nmScheduleVariant) = FRNM_PDU_SCHEDULE_VARIANT_1(FlexrayNmScheduleVariant.scheduleVariant1)		
Summary	The aim of this test is to test the sending of "positive" Nm vote on FlexRay bus when it is requested by the application. This test verifies also the Nm User Data modification. The Nm Data and the Nm Vote are located in the same PDU in a static segment.		
Needed Adaptation to other Releases	None		
Pre-conditions	The SUT is started.		
Main Test Execution			
Test Steps			**Pass Criteria**

Step 1	[LT] FrNmChannel (BUS FR1) start	
Step 2	[SWC] Requests the bus FR1 by calling ComM_RequestComMode(<user>, COMM_FULL_COMMUNICATION)	[LT] The Nm frame is sent with "positive" NM-Votes
Step 3	[SWC] Change the Nm User Data to 0x5A 0x97 0x19 0x33 0x28 by calling Rte_Write() API on Port NM_USER_DATA mapped on NmUserData by configuration.	[LT] Nm frame contains the following user data: 0x5A 0x97 0x19 0x33 0x28
Step 4	[SWC] Change the Nm User Data to 0x04 0x9A 0x32 0xB3 0x1A by calling Rte_Write() API on Port NM_USER_DATA mapped on NmUserData by configuration.	[LT] Nm frame contains the following user data: 0x04 0x9A 0x32 0xB3 0x1A
Post-conditions	None	

4.3.2 [ATS_COMM_00148] Stopping Nm vote when the network is no more requested by the application

Test Objective	Stopping Nm vote when the network is no more requested by the application		
ID	ATS_COMM_00148	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	FrNm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00123		
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00125 FlexRayNetworkManagement: SWS_FrNm_00126		
Requirements / Reference to Test Environment	Configuration use case : UC02.01		
Configuration Parameters	FrNmGlobalConfig(NmConfig.FlexrayNmCluster): - FrNmPassiveModeEnabled(NmEcu.nmPassiveModeEnabled) = FALSE FrNmChannel(FlexrayNmCluster): Bus FR1 - FrNmPduScheduleVariant(FlexrayNmClusterCoupling.nmScheduleVariant) = FRNM_PDU_SCHEDULE_VARIANT_2(FlexrayNmScheduleVariant.scheduleVariant2)		
Summary	The aim of this test is to verify that the Nm vote is stopped when the network is no more requested by the application. During this test, the Nm vote and the Nm data are located in the same PDU in a dynamic segment.		
Needed Adaptation to other Releases	None		

Pre-conditions	The SUT is started.	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] FrNmChannel (Bus FR1) start	
Step 2	[SWC] Request the network FR1 by calling Service ComM_RequestComMode(<user>, COMM_FULL_COMMUNICATION)	[LT] The Nm Vote is "positive"
Step 3	[SWC] Stop to request the network FR1 by calling Service ComM_RequestComMode(<user>, COMM_NO_COMMUNICATION)	[LT] The Nm Vote is "negative"
Post-conditions	None	

4.3.3 [ATS_COMM_00152] Keeping the FlexRay bus awake when "positive" Nm vote are received periodically

Test Objective	Keeping the FlexRay bus awake when "positive" Nm vote are received periodically		
ID	ATS_COMM_00152	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	FrNm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00123		
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00100 FlexRayNetworkManagement: SWS_FrNm_00101 FlexRayNetworkManagement: SWS_FrNm_00126 FlexRayNetworkManagement: SWS_FrNm_00129		
Requirements / Reference to Test Environment	Configuration use case : UC02.01		
Configuration Parameters	FrNmGlobalConfig(NmConfig.FlexrayNmCluster): - FrNmPassiveModeEnabled(NmEcu.nmPassiveModeEnabled) = FALSE FrNmChannel(FlexrayNmCluster): Bus FR2 - FrNmPduScheduleVariant(FlexrayNmClusterCoupling.nmScheduleVariant) = FRNM_PDU_SCHEDULE_VARIANT_3(FlexrayNmScheduleVariant.scheduleVariant3)		
Summary	The aim of this test is to verify that the SUT is able to keep the FlexRay bus awake when "positive" Nm votes are received. During this test, the Nm vote and the Nm data are located in 2 different PDUs, located in static segments.		
Needed Adaptation to	None		

other Releases		
Pre-conditions	The SUT is started.	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] FrNmChannel (Bus FR2) start	
Step 2	[LT] Send periodically "positive" Nm-Votes on the FR2 bus	[SWC] ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_FULL_COMMUNICATION
Post-conditions	None	

4.3.4 [ATS_COMM_00153] Setting the FlexRay bus in sleep when only "negative" votes are received

Test Objective	Setting the FlexRay bus in sleep when only "negative" votes are received		
ID	ATS_COMM_00153	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	FrNm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00123		
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00100 FlexRayNetworkManagement: SWS_FrNm_00101 FlexRayNetworkManagement: SWS_FrNm_00126 FlexRayNetworkManagement: SWS_FrNm_00129		
Requirements / Reference to Test Environment	Configuration use case : UC02.01		
Configuration Parameters	FrNmGlobalConfig(NmConfig.FlexrayNmCluster): - FrNmPassiveModeEnabled(NmEcu.nmPassiveModeEnabled) = FALSE FrNmChannel(FlexrayNmCluster): Bus FR2 - FrNmPduScheduleVariant(FlexrayNmClusterCoupling.nmScheduleVariant) = FRNM_PDU_SCHEDULE_VARIANT_4(FlexrayNmScheduleVariant.scheduleVariant4)		
Summary	The aim of this test is to verify that the SUT is able to set the FlexRay bus in sleep when it only receives "negative" votes. During this test, the Nm vote is located in a static segment and the Nm Data is located in a dynamic segment.		
Needed Adaptation to other Releases	None		

Pre-conditions	The SUT is started.	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] FrNmChannel (Bus FR2) start	
Step 2	[LT] Send "positive" votes to the SUT on the FR2 bus	
Step 3	[CP] Wait <NM MSG CYCLE TIME> + <safety margin>	[SWC] ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_FULL_COMMUNICATION
Step 4	[LT] Send "negative" votes to the SUT on the FR2 bus	
Step 5	[CP] Wait "(FrNmReadySleepCnt +1) * FrNmRepetitionCycle" cycles + <safety margin>	[SWC] ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_NO_COMMUNICATION
Post-conditions	None	

4.3.5 [ATS_COMM_00155] Keeping the FlexRay bus awake when "positive" Nm votes are received periodically in Passive Mode

Test Objective	Keeping the FlexRay bus awake when "positive" Nm votes are received periodically in Passive Mode		
ID	ATS_COMM_00155	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	FrNm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00123		
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00100 FlexRayNetworkManagement: SWS_FrNm_00101 FlexRayNetworkManagement: SWS_FrNm_00126 FlexRayNetworkManagement: SWS_FrNm_00129		
Requirements / Reference to Test Environment	Configuration use case : UC02.02		
Configuration Parameters	FrNmGlobalConfig(NmConfig.FlexrayNmCluster): - FrNmPassiveModeEnabled(NmEcu.nmPassiveModeEnabled) = FALSE FrNmChannel(FlexrayNmCluster): Bus FR2 - FrNmChannelTiming - FrNmRepetitionCycle(nmRepetitionCycle) is configured		

	- FrNmReadySleepCnt(nmReadySleepCount) = ((Float2Int(nmReadySleepTime/cycle))/nmRepetitionCycle)-1
Summary	The aim of this test is to verify that the SUT which is configured in Passive Mode is able to keep the FlexRay bus awake when "positive" vote are received periodically.
Needed Adaptation to other Releases	None
Pre-conditions	The SUT is started.
Main Test Execution	
Test Steps	**Pass Criteria**
Step 1 [LT] FrNmChannel (Bus FR2) start	
Step 2 [LT] Send <positive> NM-Votes periodically on the FR2 bus	
Step 3 [CP] Wait <NM MSG CYCLE TIME> + <safety margin>	[SWC] ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_FULL_COMMUNICATION
Post-conditions	None

4.3.6 [ATS_COMM_00138] Setting the FlexRay bus in sleep when only "negative" votes are received in Passive Mode

Test Objective	Setting the FlexRay bus in sleep when only "negative" votes are received in Passive Mode		
ID	ATS_COMM_00138	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	FrNm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00123		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00109 FlexRayNetworkManagement: SWS_FrNm_00100 FlexRayNetworkManagement: SWS_FrNm_00101 FlexRayNetworkManagement: SWS_FrNm_00126 FlexRayNetworkManagement: SWS_FrNm_00129		
Requirements / Reference to Test Environment	Configuration use case : UC02.02		
Configuration Parameters	FrNmGlobalConfig(NmConfig.FlexrayNmCluster): - FrNmPassiveModeEnabled(NmEcu.nmPassiveModeEnabled) = TRUE FrNmChannel(FlexrayNmCluster): Bus FR2 - FrNmChannelTiming - FrNmRepetitionCycle(nmRepetitionCycle) is configured - FrNmReadySleepCnt(nmReadySleepCount) = ((Float2Int(nmReadySleepTime/cycle))/nmRepetitionCycle)-1		

Summary	The aim of this test is to verify that the SUT is able to set the bus in sleep when it only receives "negative" votes.	
Needed Adaptation to other Releases	None	
Pre-conditions	The SUT is started.	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] FrNmChannel (Bus FR2) start	
Step 2	[LT] Send <positive> NM-Votes to the SUT on the FR2 bus	
Step 3	[CP] Wait <NM MSG CYCLE TIME> + <safety margin>	-
Step 4	[SWC] Get Current Communication Mode	
Step 5	[LT] Send <negative> NM-Votes to the SUT on the FR2 bus	
Step 6	[CP] Wait "(FrNmReadySleepCnt +1) * FrNmRepetitionCycle" cycles + <safety margin>	-
Step 7	[SWC] Get Current Communication Mode	[SWC] ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_NO_COMMUNICATION
Post-conditions	None	

4.3.7 [ATS_COMM_00157] Starting the FlexRay bus when the SUT receives a Nm frame

Test Objective	Starting the FlexRay bus when the SUT receives a Nm frame		
ID	ATS_COMM_00157	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	FrNm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00123		
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00317 FlexRayNetworkManagement: SWS_FrNm_00143 FlexRayNetworkManagement: SWS_FrNm_00116		
Requirements / Reference	Configuration use case : UC02.01		

to Test Environment		
Configuration Parameters	FrNmGlobalConfig(NmConfig.FlexrayNmCluster): - FrNmPassiveModeEnabled(NmEcu.nmPassiveModeEnabled) = FALSE FrNmChannel(FlexrayNmCluster): Bus FR2 - FrNmPduScheduleVariant(FlexrayNmClusterCoupling.nmScheduleVariant) = FRNM_PDU_SCHEDULE_VARIANT_5(FlexrayNmScheduleVariant.scheduleVariant5)	
Summary	The aim of this test is to verify that the SUT is able to start the bus (No Nm frame sent by the SUT) when the SUT receives a "positive" Nm vote. During this test, the Nm Vote is located in dynamic segment and the Nm Data is located in a static segment.	
Needed Adaptation to other Releases	None	
Pre-conditions	The SUT is started (Communication Allowed for Bus FR2 is FALSE).	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] FrNmChannel (Bus FR2)	
Step 2	[LT] Send <positive> NM-Votes periodically on FR2 bus	[SWC] ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_NO_COMMUNICATION ComM_GetState returns COMM_NO_COM_REQUEST_PENDING (Use of CDD is needed to access to ComM_GetState)
Step 3	[SWC] Call BswM Action to Allow Communication (custom action configured to call ComM_CommunicationAllowed)	[LT] The SUT must send FrNm on the FR2 bus
Post-conditions	None	

4.3.8 [ATS_COMM_00159] Repeating Nm messages on FlexRay bus when requested by the application, or by the network

Test Objective	Repeating Nm messages on FlexRay bus when requested by the application, or by the network		
ID	ATS_COMM_00159	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	FrNm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00123		

Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00116 FlexRayNetworkManagement: SWS_FrNm_00117 FlexRayNetworkManagement: SWS_FrNm_00120 FlexRayNetworkManagement: SWS_FrNm_00122 FlexRayNetworkManagement: SWS_FrNm_00126 FlexRayNetworkManagement: SWS_FrNm_00130	
Requirements / Reference to Test Environment	Configuration use case : UC02.01	
Configuration Parameters	FrNmGlobalConfig(NmConfig.FlexrayNmCluster): - FrNmPassiveModeEnabled(NmEcu.nmPassiveModeEnabled) = FALSE FrNmChannel(FlexrayNmCluster): Bus FR1 - FrNmChannelTiming - FrNmRepeatMessageTime(nmRepeatMessageTime) is configured - FrNmPduScheduleVariant(FlexrayNmClusterCoupling.nmScheduleVariant) = FRNM_PDU_SCHEDULE_VARIANT_6(FlexrayNmScheduleVariant.scheduleVariant6)	
Summary	The aim of this test is to verify that the SUT is able to repeat the Nm messages when it is requested by the application or when it is requested by the network. During this test, Nm vote and Nm data are located in separated PDU in dynamic segment.	
Needed Adaptation to other Releases	None	
Pre-conditions	The SUT is started. Test Bench send <positive> vote on FlexRay network User has requested the network ECU is in FULL COMMUNICATION state and Network is active	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] FrNmChannel (Bus FR1) start	
Step 2	[SWC] Request ComM_RequestComMode(<user>, COMM_NO_COMMUNICATION)	
Step 3	[CP] Wait 1s	
Step 4	[SWC] Get Current Communication Mode	[SWC] ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_FULL_COMMUNICATION
Step 5	[LT] Send on FR1 network : <RepeatMessageResquest> = 1	[LT] The SUT must send "positive" Nm-Vote
Step 6	[CP] Wait for <FrNmRepeastMessageTime> + <safety margin>	[LT] The SUT must not send any Nm frame
Step 7	[SWC] Request ModeSwitch (call Rte_Switch associated to BswMMode port) to REPEAT_MESSAGE_REQUEST (BswM	[LT] The SUT must send "positive" Nm-Vote <RepeatMessageResquest> bit = 1

	Action configured to request a switch in <RepeatMessageState> for the FR1 network)	
Post-conditions	None	

4.3.9 [ATS_COMM_00171] Synchronizing the FlexRay bus shutdown in a Master/Slave gateway

Test Objective	Synchronizing the FlexRay bus shutdown in a Master/Slave gateway		
ID	ATS_COMM_00171	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	FrNm CanNm Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00123		
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00181 FlexRayNetworkManagement: SWS_FrNm_00322 FlexRayNetworkManagement: SWS_FrNm_00396		
Requirements / Reference to Test Environment	Configuration use case : UC02.01		
Configuration Parameters	FrNmGlobalConfig(NmConfig.FlexrayNmCluster): - FrNmCoordinatorSyncSupport = TRUE (NmCoordinator is present on this cluster) - FrNmRemoteSleepIndicationEnabled(NmEcu.nmRemoteSleepIndEnabled) = TRUE FrNmChannel(FlexrayNmCluster): Bus FR1 CanNmGlobalConfig(NmConfig.CanNmCluster): - CanNmCoordinatorSyncSupport = TRUE (same NmCoordinator.index as FlexrayNmCluster) - CanNmRemoteSleepIndEnabled(NmEcu.nmRemoteSleepIndEnabled) = TRUE CanNmChannel(CanNmCluster): Bus CAN1 NmGlobalFeatures(NmConfig.NmEcu): - NmBusSynchronizationEnabled(nmBusSynchronizationEnabled) = TRUE - NmCoordinatorSupportEnabled = TRUE (NmCoordinator is present) NmChannelConfig(NmCluster) linked to Bus FR1: - NmActiveCoordinator(NmCoordinator.nmActiveCoordinator) = FALSE - NmSynchronizingNetwork(nmSynchronizingNetwork) = TRUE NmChannelConfig(NmCluster) linked to Bus CAN1: - NmActiveCoordinator(NmCoordinator.nmActiveCoordinator) = TRUE - NmSynchronizingNetwork(nmSynchronizingNetwork) = FALSE		
Summary	The aim of this test is to verify that the SUT is able to detect a remote sleep indication from the network, and to synchronize the shutdown of one FlexRay bus and one CAN bus. The FlexRay bus is configured in Passive coordinator, and the CAN bus is configured in Active coordinator (Backbone bus is the FlexRay bus).		
Needed Adaptation to other Releases	None		

Pre-conditions	The SUT is started. Nm frames with "positive" vote are sent on FR1 bus No Nm frames are sent on CAN1 bus	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] FrNmChannel (Bus FR1) start	
Step 2	[LT] CanNmChannel (Bus CAN1) start	
Step 3	[LT] Send on FR1 bus with "positive" vote	[LT] SUT must send Nm frames on CAN1 bus SUT must send "negative" vote on FR1 bus
Step 4	[SWC] Request the FR1 bus by calling ComM_RequestComMode(<user>, COMM_FULL_COMMUNICATION)	[LT] SUT must send Nm frames on CAN1 bus SUT must send "positive" vote on FR1 bus
Step 5	[LT] Send on FR1 bus with "negative" vote	[LT] SUT must send Nm frames on CAN1 bus SUT must send "positive" vote on FR1 bus
Step 6	[LT] Send on FR1 bus with "positive" vote	[LT] SUT must send Nm frames on CAN1 bus SUT must send "positive" vote on FR1 bus
Step 7	[SWC] Release the FR1 bus by calling ComM_RequestComMode(<user>, COMM_NO_COMMUNICATION)	[LT] SUT must send Nm frames on CAN1 bus SUT must send "negative" vote on FR1 bus
Step 8	[LT] Send on FR1 bus with "negative" vote	[SWC] FR1 : ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_FULL_COMMUNICATION CAN1 : ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_FULL_COMMUNICATION
Step 9	[CP] Wait for the end of one FlexRay cycle	[SWC] FR1 : ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_NO_COMMUNICATION

		CAN1 : ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_NO_COMMUNICATION
Step 10	-	[LT] SUT must not send Nm frames on CAN1 bus
Post-conditions	None	

4.3.10 [ATS_COMM_00263] Synchronizing the FlexRay bus shutdown in a Master gateway

Test Objective	Synchronizing the FlexRay bus shutdown in a Master gateway		
ID	ATS_COMM_00263	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	FrNm CanNm Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00123		
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00181 FlexRayNetworkManagement: SWS_FrNm_00322 FlexRayNetworkManagement: SWS_FrNm_00396		
Requirements / Reference to Test Environment	Configuration use case : UC02.01		
Configuration Parameters	FrNmGlobalConfig(NmConfig.FlexrayNmCluster): - FrNmCoordinatorSyncSupport = TRUE (NmCoordinator is present on this cluster) - FrNmRemoteSleepIndicationEnabled(NmEcu.nmRemoteSleepIndEnabled) = TRUE FrNmChannel(FlexrayNmCluster): Bus FR1 - FlexrayNmClusterCoupling.nmScheduleVariant = scheduleVariant7 CanNmGlobalConfig(NmConfig.CanNmCluster): - CanNmCoordinatorSyncSupport = TRUE (same NmCoordinator.index as FlexrayNmCluster) - CanNmRemoteSleepIndEnabled(NmEcu.nmRemoteSleepIndEnabled) = TRUE CanNmChannel(CanNmCluster): Bus CAN1 NmGlobalFeatures(NmConfig.NmEcu): - nmBusSynchronizationEnabled = TRUE - NmCoordinatorSupportEnabled = TRUE (NmCoordinator is present) NmChannelConfig(NmCluster) => FR1: - NmActiveCoordinator(NmCoordinator.nmActiveCoordinator) = TRUE - NmSynchronizingNetwork(nmSynchronizingNetwork) = TRUE NmChannelConfig(NmCluster) => CAN1: - NmActiveCoordinator(NmCoordinator.nmActiveCoordinator) = TRUE - NmSynchronizingNetwork(nmSynchronizingNetwork) = FALSE		
Summary	The aim of this test is to verify that the SUT is able to detect a remote sleep indication from the network, and to synchronize the shutdown of one FlexRay bus		

	and one CAN bus. The FlexRay bus is configured in Active coordinator, and the CAN bus is configured in Active coordinator During this test, the Nm-Vote and the CBV are located in static segment, and the Nm-Data is located in dynamic segment.	
Needed Adaptation to other Releases	None	
Pre-conditions	The SUT is started. Nm frames with "positive" vote are sent on FR1 bus Nm frames are sent on CAN1 bus	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] FrNmChannel (Bus FR1) start	
Step 2	[LT] CanNmChannel (Bus CAN1) start	
Step 3	[LT] Send on FR1 bus with "positive" vote Send Nm frames on CAN1 bus	[LT] SUT must send Nm frames on CAN1 bus SUT must send "positive" vote on FR1 bus (NM FR1 and NM CAN1 maintain coordinated cluster active)
Step 4	[LT] Stop sending Nm frames on CAN1 bus	[LT] SUT must send Nm frames on CAN1 bus SUT must send "positive" vote on FR1 bus (NM FR1 maintains coordinated cluster active)
Step 5	[SWC] Request the FR1 bus by calling ComM_RequestComMode(<user>, COMM_FULL_COMMUNICATION)	[LT] SUT must send Nm frames on CAN1 bus SUT must send "positive" vote on FR1 bus (NM FR1 maintains coordinated cluster active)
Step 6	[LT] Send on FR1 bus with "negative" vote	[LT] SUT must send Nm frames on CAN1 bus SUT must send "positive" vote on FR1 bus (NM FR1 maintains coordinated cluster active)
Step 7	[LT] Send Nm frames on CAN1 bus	[LT] SUT must send Nm frames on CAN1 bus SUT must send "positive" vote on FR1 bus (NM FR1 and NM CAN1 maintain coordinated cluster active)
Step 8	[SWC] Release the FR1 bus by calling ComM_RequestComMode(<user>, COMM_NO_COMMUNICATION)	[LT] SUT must send Nm frames on CAN1 bus SUT must send "positive" vote on FR1 bus (NM FR1 and NM CAN1 maintain coordinated cluster active)
Step 9	[LT] Before end of FlexRay Cycle, Stop sending Nm frames on CAN1 bus	[LT] SUT must send Nm frames on CAN1 bus SUT must send "positive" vote on

		FR1 bus (NM FR1 and NM CAN1 maintain coordinated cluster active)
Step 10	-	[SWC] FR1 : ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_FULL_COMMUNICATION CAN1 : ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_FULL_COMMUNICATION
Step 11	[CP] Wait for the end of one FlexRay cycle Wait for "NmGlobalCoordinatorTime" seconds	[LT] SUT must NOT send Nm frames on CAN1 bus SUT must send "negative" vote on FR1 bus (NM FR1 and NM CAN1 shutdown is coordinated)
Step 12	-	[SWC] FR1 : ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_NO_COMMUNICATION CAN1 : ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_NO_COMMUNICATION
Post-conditions	None	

4.3.11 [ATS_COMM_00791] Transition From Bus Sleep To Normal Operation Through Application Request

Test Objective	Transition From Bus Sleep To Normal Operation Through Application Request		
ID	ATS_COMM_00791	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	FRNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00121 FlexRayNetworkManagement: SWS_FrNm_00131		
Requirements / Reference to Test Environment	none		

Configuration Parameters	NetworkManagementFlexRayNmCluster.nmRepetitionCycle = FRNM_CYCLE_VALUE_8 FrNm_RepeatMessage = FALSE NetworkManagement::FlexRayNmCluster.nmReadySleepCount = Eg:2 FrNm_NetworkRequest = TRUE NetworkManagement::nmEcu.nmStateChangeIndEnabled = TRUE NetworkManagement::FlexrayNmCluster.nmRepeatMessageTime = FRNM_REPEAT_MESSAGE_TIME = Eg:6 NetworkManagementFlexRayNmCluster.nmVotingCycle = FRNM_CYCLE_VALUE_8	
Summary	By making the DUT in to full communication mode The FrNm module shall leave the Bus Sleep and enter the RepeatMessageState.Once Repeat Message timer expires and with the availability of Network request FrNm module will enter in to Normal operation state. State change notification service is enabled and FrNm module shall notify all its state changes to the NmIf module by calling Nm_StateChangeNotification.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Invoke Rte_Call to request ComM in COMM_FULL_COMMUNICATION.	[SWC] Rte_Call shall return RTE_E_OK.
Step 2	[LT] Monitor and validate the frame.	[LT] Observe Nm frames on bus.
Step 3	[SWC] Check if NM state and mode are desired one after FRNM_REPEAT_MESSAGE_TIME expires.-	[SWC] NM mode shall be in “network mode” and NM state shall be in “normal operation”
Step 4	-	[SWC] Nm_StateChangeNotification shall be invoked with nmPreviousState as “repeat message state” and nmCurrentState as “Normal Operation State”.
Step 5	-	[LT] Continue to observe Nm frames on the bus.
Post-conditions	None	

4.3.12 [ATS_COMM_00792] Transition From Normal Operation To Bus Sleep Through Application Request

Test Objective	Transition From Normal Operation To Bus Sleep Through Application Request		
ID	ATS_COMM_00792	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	FRNM	**State**	reviewed

Trace to Requirement on Acceptance Test Document		
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00129 FlexRayNetworkManagement: SWS_FrNm_00138	
Requirements / Reference to Test Environment	none	
Configuration Parameters	NetworkManagementFlexRayNmCluster.nmRepetitionCycle = FRNM_CYCLE_VALUE_8 FrNm_RepeatMessage = FALSE NetworkManagement::FlexRayNmCluster.nmReadySleepCount = Eg:2 FrNm_NetworkRequest = TRUE NetworkManagement::nmEcu.nmStateChangeIndEnabled = TRUE NetworkManagement::FlexrayNmCluster.nmRepeatMessageTime = FRNM_REPEAT_MESSAGE_TIME = Eg:6 NetworkManagementFlexRayNmCluster.nmVotingCycle = FRNM_CYCLE_VALUE_8	
Summary	By making the DUT in to full communication mode the FrNm module shall leave the Bus Sleep and enter the Repeat Message State. Once Repeat Message timer expires and with the availability of Network request FrNm module will enter in to Normal operation state. By making the DUT in to No communication mode the FrNm module shall leave the normal operation state and enter the ready sleep state and after ready sleep counter value expires FrNm will enter in to bus sleep mode. State change notification service is enabled and FrNm module shall notify all its state changes to the NmIf module by calling Nm_StateChangeNotification.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Invoke Rte_Call to request ComM in COMM_FULL_COMMUNICATION.	[SWC] Rte_Call shall return RTE_E_OK.
Step 2	[LT] Monitor and validate the frame.	[LT] Observe Nm frames on bus.
Step 3	[SWC] Check if NM state and mode are desired one after FRNM_REPEAT_MESSAGE_TIME expires.	[SWC] NM mode shall be in “network mode” and NM state shall be in “normal operation”
Step 4	[SWC] Invoke Rte_Call to request ComM in COMM_NO_COMMUNICATION.	[SWC] Rte_Call shall return RTE_E_OK.
Step 5	-	[SWC] Nm_StateChangeNotification shall be invoked with nmPreviousState as “normal operation” and nmCurrentState as “ready sleep”.
Step 6	-	[LT] Tester shall not observe Nm frames on the bus.

Step 7	[SWC] Check if NM state and mode are desired one after FRNM_READY_SLEEP_CNT expires.	[SWC] Nm_StateChangeNotification shall be invoked with nmPreviousState as “ready sleep” and nmCurrentState as “bus sleep”.
Post-conditions	NONE	

4.3.13 [ATS_COMM_00793] Transition From BUS Sleep To Ready Sleep State Through Passive Startup

Test Objective	Transition From BUS Sleep To Ready Sleep State Through Passive Startup		
ID	ATS_COMM_00793	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	FRNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00138		
Requirements / Reference to Test Environment	none		
Configuration Parameters	NetworkManagementFlexRayNmCluster.nmRepetitionCycle = FRNM_CYCLE_VALUE_8 FrNm_RepeatMessage = FALSE NetworkManagement::FlexRayNmCluster.nmReadySleepCount < Eg:1 FrNm_NetworkRequest = FALSE NetworkManagement::nmEcu.nmStateChangeIndEnabled = TRUE NetworkManagement::FlexrayNmCluster.nmRepeatMessageTime = FRNM_REPEAT_MESSAGE_TIME = Eg:6 NetworkManagementFlexRayNmCluster.nmVotingCycle = FRNM_CYCLE_VALUE_8 NetworkManagement::nmEcu.nmPassiveModeEnabled = TRUE		
Summary	Send a wakeup pattern from the other node DUT will accept and will go to repeat message state and when the repeat message timer expires and there is no application request FrNm module will transfer to ready sleep state. Hint: NM frame is configured in static slots (Vote bit will be checked in the receiving node)		
Needed Adaptation to other Releases			
Pre-conditions	DUT shall be initialized		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[LT] Send a valid Wakeup pattern to the DUT.		[SWC] Passive wakeup shall occur.
Step 2	[SWC] Check if NM state and mode are desired one		[SWC] NM mode shall be in “network mode”

	after FRNM_REPEAT_MESSAGE_TIME expires.	and NM state shall be in “ready sleep”.
Step 3	[LT] Send the NM frame with vote bit set from analyzer.	Hint: FrNm will stay in the ready sleep state by restarting the ready sleep counter
Step 4	[SWC] Check if NM state and mode are desired one.	[SWC] NM mode shall be in “network mode” and NM state shall be in “ready sleep”.
Step 5	[LT] Send the NM frame with vote bit not set from analyzer.	-
Step 6	[SWC] Check if NM state and mode are desired one after FRNM_READY_SLEEP_CNT+1 no of repetition cycles.	[SWC] NM mode shall be in “bus sleep mode” and NM state shall be in “ready sleep”.
Step 7	-	[LT] Tester shall observe no Frames on bus.
Post-conditions	NONE	

4.3.14 [ATS_COMM_00794] Transition From Sleep To Ready Sleep State Through Passive Startup And With Application Request Transition To Normal Operation

Test Objective	Transition From Sleep To Ready Sleep State Through Passive Startup And With Application Request Transition To Normal Operation		
ID	ATS_COMM_00794	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	FRNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00120 FlexRayNetworkManagement: SWS_FrNm_00104		
Requirements / Reference to Test Environment	none		
Configuration Parameters	NetworkManagementFlexRayNmCluster.nmRepetitionCycle = FRNM_CYCLE_VALUE_8 FrNm_RepeatMessage = FALSE NetworkManagement::FlexRayNmCluster.nmReadySleepCount = Eg:2 FrNm_NetworkRequest = TRUE NetworkManagement::nmEcu.nmStateChangeIndEnabled = TRUE NetworkManagement::FlexrayNmCluster.nmRepeatMessageTime = FRNM_REPEAT_MESSAGE_TIME = Eg:6 NetworkManagementFlexRayNmCluster.nmVotingCycle = FRNM_CYCLE_VALUE_8		

Summary	Send a wakeup pattern (passive startup) from the other node DUT will accept and will go to repeat message state and when the repeat message timer expires and there is application request FrNm module will transit to normal operation state. Hint: NM frame is configured in static slots (Vote bit will be checked in the receiving node)	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] Send a valid frame to the DUT.	[LT] Observe Nm frames on the bus as long as FRNM_REPEAT_MESSAGE_TIME is active.
Step 2	[SWC] Check if NM state and mode are desired one after FRNM_REPEAT_MESSAGE_TIME expires.	[SWC] NM mode shall be in “network mode” and NMstate shall be in “ready sleep”.
Step 3	[LT] Send the NM frame with Vote bit set from analyzer.	Hint :FrNm will stay in the “ready sleep” state by restarting the “ready sleep” counter.
Step 4	[SWC] Check if NM state and mode are the desired one.	[SWC] NM mode shall be in “network mode” and NM state shall be in “ready sleep”
Step 5	[SWC] Invoke Rte_Call to request ComM in COMM_FULL_COMMUNICATION.	[SWC] Rte_Call shall return RTE_E_OK.
Step 6	-	[SWC] Nm_StateChangeNotification shall be invoked with nmPreviousState as “ready sleep” and nmCurrentState as “normal operation” state.
Post-conditions	NONE	

4.3.15 [ATS_COMM_00795] Transition Bus Sleep To Synchronize Mode, Deactivation Of Nm Messages In Synchronize Mode And Activation Of Nm Messages In Repeat Message Mode

Test Objective	Transition Bus Sleep To Synchronize Mode, Deactivation Of Nm Messages In Synchronize Mode And Activation Of Nm Messages In Repeat Message Mode		
ID	ATS_COMM_00795	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	FRNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			

Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00316 FlexRayNetworkManagement: SWS_FrNm_00308 FlexRayNetworkManagement: SWS_FrNm_00116
Requirements / Reference to Test Environment	none
Configuration Parameters	NetworkManagementFlexRayNmCluster.nmRepetitionCycle = FRNM_CYCLE_VALUE_8 FrNm_RepeatMessage = FALSE NetworkManagement::FlexRayNmCluster.nmReadySleepCount = Eg:2 FrNm_NetworkRequest = TRUE NetworkManagement::nmEcu.nmStateChangeIndEnabled = TRUE NetworkManagement::FlexrayNmCluster.nmRepeatMessageTime = FRNM_REPEAT_MESSAGE_TIME = Eg:6 NetworkManagementFlexRayNmCluster.nmVotingCycle = FRNM_CYCLE_VALUE_8
Summary	Request for network which pushes the FrNm from bus sleep mode to synchronize mode and the de-activation of NM Message happens in synchronize mode and again the NM messages are activated in repeat message mode.
Needed Adaptation to other Releases	
Pre-conditions	DUT shall be initialized

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Invoke Rte_Call to request ComM in COMM_FULL_COMMUNICATION.	[SWC] Rte_Call shall return RTE_E_OK.
Step 2	[SWC] Check if NM state and mode is the desired one.	[SWC] NM mode shall be “synchronize Mode” and NM state shall be in “synchronize” state.
Step 3	[LT] Tester shall monitor and validate the frame.	[LT] Tester shall observe Nm frames on bus.
Step 4	[SWC] Wait till the repetition cycle is completed.	-
Step 5	[SWC] Check for NM state and mode.	[SWC] NM mode shall be in “network mode” and NM state shall be in “repeat Message”
Step 6	[LT] Tester shall monitor and validate the frame.	[LT] Tester shall observe Nm frames on bus.
Post-conditions	NONE	

4.3.16 [ATS_COMM_00796] Setting The Source Node Identifier In Nm-Pdu

Test Objective	Setting The Source Node Identifier In Nm-Pdu		
ID	ATS_COMM_00796	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2

Affected Modules	FRNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00037		
Requirements / Reference to Test Environment	none		
Configuration Parameters	NetworkManagement::FlexrayNmNode.nmNodeId = Eg:0x01 NetworkManagement::nmEcu.nmPassiveModeEnabled = FALSE NetworkManagement::NmEcu.nmNodeDetectionEnabled = TRUE NetworkManagement::NmEcu.nmUserDataEnabled = TRUE NetworkManagement::NmEcu.nmNodeIdEnabled = TRUE		
Summary	Configure the source node identifier and check the NM frames for the configured source node identifier on TESTER.		
Needed Adaptation to other Releases			
Pre-conditions	DUT shall be initialized		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[SWC] Invoke Rte_Call to request ComM in COMM_FULL_COMMUNICATION.		[SWC] Rte_Call shall return RTE_E_OK.
Step 2	[LT] Monitor and validate the frame.		[LT] Observe Nm frames on bus with the configured source node identifier i.e. “0x01”.
Post-conditions	NONE		

4.3.17 [ATS_COMM_00797] Get The Nm Pdu Data Out Of The Most Recently Received Nm Message

Test Objective	Get The Nm Pdu Data Out Of The Most Recently Received Nm Message		
ID	ATS_COMM_00797	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	FRNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00265		
Requirements / Reference to Test Environment	none		

Configuration Parameters	NetworkManagement::FlexrayNmNode.nmNodeId = Eg:0x01 NetworkManagement::nmEcu.nmPassiveModeEnabled = FALSE NetworkManagement::NmEcu.nmNodeDetectionEnabled = TRUE NetworkManagement::NmEcu.nmUserDataEnabled = TRUE NetworkManagement::NmEcu.nmNodeIdEnabled = TRUE	
Summary	Nm will get the whole Pdu Data out of the most recently received NM message by calling FrNm. This is from OEM specific extensions of the NM stack which shall be used to analyse PDU Data.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be in full communication	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] Monitor and validate the frame for NM-Pdus.	[LT] Observe Nm frames on bus.
Step 2	[LT] Send a NM frame periodically to DUT with Node ID CBV and User Data.	-
Step 3	[SWC] Trigger Nm_GetPduDatato get the most recently received network management PDU.	[SWC] Nm_GetPduData shall return E_OK. nmPduDatapointer shall be updated with the NM PDU data out of the most recently received NM message.
Step 4	-	[SWC] Observe the same Node ID Control Bit Vector and User Data.
Post-conditions	NONE	

4.3.18 [ATS_COMM_00798] Transition From Normal Operation State To Repeat Message State On Reception Of Repeat Message Request Bit From The Other Node

Test Objective	Transition From Normal Operation State To Repeat Message State On Reception Of Repeat Message Request Bit From The Other Node		
ID	ATS_COMM_00798	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	FRNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00124		
Requirements / Reference to Test Environment	none		
Configuration Parameters	NetworkManagement::NmEcu.nmRepeatMsgIndEnabled = TRUE NetworkManagement::NmEcu.nmNodeDetectionEnabled = TRUE NetworkManagement::NmEcu.nmUserDataEnabled = TRUE		

Summary	When the repeat message request bit is received in the normal operation state an indication to Nm is checked.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Invoke Rte_Call to request ComM in COMM_FULL_COMMUNICATION.	[SWC] Rte_Call shall return RTE_E_OK.
Step 2	[LT] Monitor and validate the frame for NM-Pdus.	[LT] Observe Nm frames on bus.
Step 3	[LT] Send a NM frame periodically to DUT with repeat message request bit (Bit 0 in CBV) as set.	[SWC] Nm_StateChangeNotification shall be invoked with nmPreviousState as “normal operation” and nmCurrentState as “repeat message”.
Post-conditions	NONE	

4.3.19 [ATS_COMM_00799] Get The Node Id Of The Recently Received Nm Message And To Get The Local Node Identifier For The Local Host Node

Test Objective	Get The Node Id Of The Recently Received Nm Message And To Get The Local Node Identifier For The Local Host Node		
ID	ATS_COMM_00799	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	FRNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00190 FlexRayNetworkManagement: SWS_FrNm_00047 FlexRayNetworkManagement: SWS_FrNm_00046		
Requirements / Reference to Test Environment	none		
Configuration Parameters	NetworkManagement::FlexrayNmNode.nmNodeId = Eg:0x01 NetworkManagement::nmEcu.nmPassiveModeEnabled = FALSE NetworkManagement::NmEcu.nmNodeDetectionEnabled = TRUE NetworkManagement::NmEcu.nmUserDataEnabled = TRUE NetworkManagement::NmEcu.nmNodeIdEnabled = TRUE NetworkManagement::nmEcu.nmPduRxIndicationEnabled = TRUE		
Summary	Nm will get the node identifier out of the last successfully received NM message by calling FrNm to get the node Id and Nm shall get the node identifier configured for the local node by calling FrNm for that purpose.		
Needed Adaptation to other Releases			

Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Invoke Rte_Call to request ComM in COMM_FULL_COMMUNICATION.	[SWC] Rte_Call shall return RTE_E_OK.
Step 2	[LT] Tester shall send a NM frame periodically to DUT with node identifier and User Data.	[CP] NM_PduRxIndication callback function shall be invoked indicating receiption of NM frame.
Step 3	[SWC] Trigger Nm_GetNodeIdentifier to get node identifier out of most recently received NmPdu. Monitor and validate the frame for node Id to match.	[SWC] Nm_GetNodeIdentifier shall return E_OK. nmNodeIdPtr shall be loaded with node identifier out of last successfully received NM message.
Step 4	[SWC] Trigger Nm_GetLocalNodeIdentifier request to get the node identifier of local node.	[SWC] nmNodeIdPtr shall be loaded with the node id configured for the local node i.e. “FRNM_NODE_ID”
Post-conditions	The DUT shall be reset	

4.3.20 [ATS_COMM_00800] Setting The Nm User Data For The Nm Pdus Transmitted Next On The Bus And To Get User Data Out Of The Most Recently Received Nm Message

Test Objective	Setting The Nm User Data For The Nm Pdus Transmitted Next On The Bus And To Get User Data Out Of The Most Recently Received Nm Message		
ID	ATS_COMM_00800	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	FRNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00043 FlexRayNetworkManagement: SWS_FrNm_00044		
Requirements / Reference to Test Environment	none		
Configuration Parameters	NetworkManagement::nmEcu.nmPassiveModeEnabled = FALSE NetworkManagement::NmEcu.nmUserDataEnabled = TRUE FibexCore::CoreCommunication::NmPdu.nmDataInformation = FALSE FrNmPduLength = Eg:0x06		
Summary	To set the NM user data for NM pdus to be transmitted and to check that Nm User data is obtained out of the most recently received NM message when FrNm is requested by Nm to get the user data.		
Needed Adaptation to other Releases			

Pre-conditions	DUT shall be in full communication	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Trigger Nm_SetUserData to send data from DUT.	[SWC] Nm_SetUserData shall return E_OK.
Step 2	[LT] Monitor and validate frame on bus	[LT] Observe NM frames on bus without any errors.
Step 3	[LT] Tester shall send valid NM frame periodically to DUT with User Data different from the one written by SWC in Step 1.	[CP] NM_PduRxIndication callback function shall be invoked indicating reception of NM frame.
Step 4	[SWC] Trigger Nm_GetUserData to read data out of the last successfully received NM message in DUT.	[SWC] Nm_GetUserData shall return E_OK. nmUserDataPtr pointer shall be updated with user data of the most recently received Nm-PDU.
Step 5	[LT] Again send valid NM frames periodically to DUT with User Data other than the previously received user data from SWC. Hint: This is to check whether we get the user data out of the last successfully received NM message or not.	[CP] NM_PduRxIndication callback function shall be invoked indicating reception of NM frame.
Step 6	[SWC] Trigger Nm_GetUserData to read data out of the last successfully received NM message in DUT.	[SWC] Nm_GetUserData shall return E_OK. nmUserDataPtr pointer shall be updated with user data of the most recently received Nm-PDU (Transmitted by LT).
Post-conditions	NONE	

4.3.21 [ATS_COMM_00801] Identify The Ecu That Awakes The Network

Test Objective	Identify The Ecu That Awakes The Network		
ID	ATS_COMM_00801	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	FRNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item			
Requirements / Reference to Test Environment	none		
Configuration Parameters	NetworkManagement::nmEcu.nmPassiveModeEnabled = FALSE NetworkManagement::NmEcu.nmUserDataEnabled = TRUE		

	FibexCore::CoreCommunication::NmPdu.nmDataInformation = FALSE FrNmPduLength = Eg:0x06	
Summary	FrNm performs a state change from state Synchronize to state Network Mode and the previous state change from state Bus Sleep Mode to Synchronize was caused by an application call (due to an active wakeup) the FrNm shall set the ActiveWakeupBit in the CBV. If the FrNm leaves the Network Mode the FrNm shall reset the ActiveWakeupBit in the CBV.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Invoke Rte_Call to request ComM in COMM_FULL_COMMUNICATION.	[SWC] Rte_Call shall return RTE_E_OK.
Step 2	[SWC] Check if the NM state and mode is the desired one after FRNM_REPETITION_CYCLE completes.	[SWC] NM mode shall be in “network mode” and NM state shall be in “repeat message” state.
Step 3	[LT] Monitor and validate frames.	[LT] Observe frames on the bus without any error and with the fourth bit of Byte0 in the NM frame set.
Step 4	[SWC] Invoke Rte_Call to request ComM in COMM_NO_COMMUNICATION.	[SWC] Rte_Call shall return RTE_E_OK.
Step 5	-	[SWC] Nm_StateChangeNotification shall be invoked with nmPreviousState as “ready sleep” and nmCurrentState as “bus sleep”.
Step 6	[LT] Send a valid Wakeup pattern to the DUT.	[SWC] Passive wakeup shall occur.
Step 7	[LT] Monitor and validate the frames.	[LT] Observe for a FRNM_REPEAT_MESSAGE_TIME amount of time frames on bus without any error and with the fourth bit of Byte0 in the NM frame not set.
Post-conditions	NONE	

4.3.22 [ATS_COMM_00802] NM Messages Periodicity in Network Mode and PNI Bit not set in the CBV

Test Objective	NM Messages Periodicity in Network Mode and PNI Bit not set in the CBV		
ID	ATS_COMM_00802	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	FRNM	**State**	reviewed
Trace to Requirement			

on Acceptance Test Document			
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00404		
Requirements / Reference to Test Environment	none		
Configuration Parameters	NetworkManagement::nmEcu.nmPassiveModeEnabled = FALSE NetworkManagement::NmEcu.nmUserDataEnabled = TRUE FrNmPduLength = Eg:0x06 FrNmPnEnabled = FALSE NetworkManagement::FlexrayNmCluster.nmRepeatMessageTime = Eg:6 NetworkManagement::FlexrayNmCluster.nmMessageTimeoutTime = Eg:0x04 NetworkManagement::nmEcu.nmPduRxIndicationEnabled = TRUE		
Summary	To check the bus for transmission of NM-PDUs periodically. Configuration of FrNmPnEnabled is FALSE so FrNm shall clear the PNI Bit in the CBV.		
Needed Adaptation to other Releases			
Pre-conditions	DUT shall be in full communication		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[LT] Monitor and validate frames.		[LT] Observe NM frames for the configured message cycle time and with the PNI bit cleared on the bus.
Post-conditions	NONE		

4.3.23 [ATS_COMM_00803] Received NM PDU evaluation, PNC bit evaluation and Enable-Disable configured PNCs

Test Objective	Received NM PDU evaluation, PNC bit evaluation and Enable-Disable configured PNCs		
ID	ATS_COMM_00803	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	FRNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00411 FlexRayNetworkManagement: SWS_FrNm_00418 FlexRayNetworkManagement: SWS_FrNm_00420 FlexRayNetworkManagement: SWS_FrNm_00258		
Requirements / Reference to Test Environment	none		

Configuration Parameters	NetworkManagement::FlexRayNmClusterCoupling.nmControlBitVectorEnabled = TRUE NetworkManagement::nmEcu.nmPassiveModeEnabled = TRUE FrNmPnEnabled = TRUE FrNmPnInfoOffset = Eg:0x02 FrNmPnInfoLength = Eg:0x01 FrNmPnFilterMaskByteIndex = Eg:0x02 FrNmPnFilterMaskByteValue = Eg:0x01	
Summary	From the CBV byte of the NM-pdu partial networking information availability can be confirmed and Relevancy of the Nm-frames to a particular ECU can be determined by parameters such as FrNmPnInfoOffset FrNmPnInfoLength and FrNmPnFilterMaskByteIndex FrNmPnFilterMaskByteValue.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] Send a valid frame to the DUT.	[SWC] Passive wakeup shall occur on the DUT.
Step 2	[LT] Send NM frames with the sixth bit from byte 0 and second bit from byte 2 set periodically to DUT.	-
Step 3	[LT] Monitor and validate the frames.	[LT] Observe frames on bus related to PNC1.
Step 4	[LT] Send NM frames with the sixth bit from byte 0 set and the second bit from byte 2 not set periodically to DUT.	-
Step 5	[LT] Monitor and validate the frames.	[LT] Observe no frames on the bus.
Post-conditions	The DUT shall be reset	

4.3.24 [ATS_COMM_00804] Behavioural Check For Reception Path When CarWakeUp Is Enabled And To Get The Pdu Data For The Nm Pdu Data That Causes The CarWakeUp Indication

Test Objective	Behavioural Check For Reception Path When CarWakeUp Is Enabled And To Get The Pdu Data For The Nm Pdu Data That Causes The CarWakeUp Indication		
ID	ATS_COMM_00804	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	FRNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00190 FlexRayNetworkManagement: SWS_FrNm_00410		

	FlexRayNetworkManagement: SWS_FrNm_00411 FlexRayNetworkManagement: SWS_FrNm_00412
Requirements / Reference to Test Environment	none
Configuration Parameters	FrNmCwuRxEnabled = TRUE NetworkManagement::NmEcu.nmUserDataEnabled = TRUE NmPduRxIndicationEnabled = TRUE NetworkManagement::NmEcu.nmPduRxIndicationEnabled = Eg:0x02 NetworkManagement::FlexrayNmCluster.nmCarWakeUpBitPosition = Eg:1 Hint: FrNmCarWakeUpBytePosition = 2, FrNmCarWakeUpBitPosition = 0
Summary	Every ECUs of the car shall be able to wake up all other ECUs of the car and keep them awake. This includes all ECUs independent of the used bus system and the position in the topology of the car. To provide the request for a “CarWakeUp” (CWU) in an efficient and standardized way the request shall be contained in the NM message of the ECU. The requesting ECU sends its own NM message with a CWU request to check the nmPduData when FrNm will call Nm to get the Pdu data which causes the call of CarWakeUp indication in Nm.
Needed Adaptation to other Releases	
Pre-conditions	DUT shall be in full communication
Main Test Execution	
Test Steps	**Pass Criteria**
Step 1 [LT] Send a NM frame with UserData to the DUT.	[SWC] Nm_CarWakeUpIndication shall be invoked.
Step 2 [SWC] Trigger FrNm_GetPduData to get the most recently received network management PDU.	[SWC] FrNm_GetPduData shall return E_OK. nmPduDatapointer shall be updated with the NM PDU data out of the most recently received NM message.
Post-conditions	NONE

4.3.25 [ATS_COMM_00805] Behaviour Of Stack For Passive Startup, When Nm Frame Is Configured In Dynamic Segment

Test Objective	Behaviour Of Stack For Passive Startup, When Nm Frame Is Configured In Dynamic Segment		
ID	ATS_COMM_00805	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	FRNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	FlexRayNetworkManagement: SWS_FrNm_00148 FlexRayNetworkManagement: SWS_FrNm_00258		

Requirements / Reference to Test Environment	none	
Configuration Parameters	FrNmPduScheduleVariant = FRNM_PDU_SCHEDULE_VARIANT_2 FrNmPassiveModeEnabled = TRUE	
Summary	DUT which is started using passive wake up and NM frames configured in dynamic segment would be in bus communication for certain amount of time and go to bus no communication for absence of NM frames on the bus.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] Send a wakeup pattern frame to the DUT.	[SWC] Passive wakeup shall occur.
Step 2	[LT] Send NM frames continuously.	[SWC] DUT will stay awake.
Step 3	[LT] Monitor and validate frames.	[LT] Observe for a FRNM_REPEAT_MESSAGE_TIME amount of time NM frames on bus
Step 4	[LT] Stop sending NM frames.	Hint: Without the reception of NM frames the DUT will go to sleep.
Step 5	[LT] Monitor and validate frames.	[LT] Observe no NM frames on bus.
Post-conditions	NONE	

4.3.26 [ATS_COMM_00172] Waking up the LIN bus when the network is requested by the application

Test Objective	Waking up the LIN bus when the network is requested by the application		
ID	ATS_COMM_00172	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	LinNm	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	LINNetworkManagement: SWS_LinNm_00162 LINNetworkManagement: SWS_LinNm_00015		
Requirements / Reference to Test Environment	Configuration use case : UC02.03		

Configuration Parameters	LinNmGlobalConfig(NmConfig.NmCluster.communicationCluster is associated to LinCluster): - LinNmPassiveModeEnabled(NmEcu.nmPassiveModeEnabled) = FALSE LinNmChannelConfig(NmCluster): Bus LIN1
Summary	The aim of this test is to verify that the SUT is able to wake up the LIN bus when the network is requested by the application.
Needed Adaptation to other Releases	None
Pre-conditions	The SUT is started. no LIN frame sent on the bus

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] LinNmChannelConfig (Bus LIN1) start	
Step 2	[SWC] Requests the network by calling ComM_RequestComMode(<user>,COMM_FULL_COMMUNICATION)	[SWC] ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_FULL_COMMUNICATION
Step 3	-	[LT] The SUT starts to send frames on the LIN1 network
Post-conditions	None	

4.3.27 [ATS_COMM_00173] Setting the LIN bus in sleep when the network is no more requested by the application

Test Objective	Setting the LIN bus in sleep when the network is no more requested by the application		
ID	ATS_COMM_00173	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	LinNm, LinSM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	LINNetworkManagement: SWS_LinNm_00004 LINNetworkManagement: SWS_LinNm_00015		
Requirements / Reference to Test Environment	Configuration use case : UC02.03		

Configuration Parameters	LinNmGlobalConfig(NmConfig.NmCluster.communicationCluster is associated to LinCluster): - LinNmPassiveModeEnabled(NmEcu.nmPassiveModeEnabled) = FALSE LinNmChannelConfig(NmCluster): Bus LIN1 LinSMChannel(LinCluster): - LinSMSleepSupport(EcuInstance.sleepModeSupported) = TRUE	
Summary	The aim of this test is to verify that the SUT is able to shutdown the LIN bus when the network is no more requested by the application.	
Needed Adaptation to other Releases	None	
Pre-conditions	The SUT is started. no LIN frame sent on the bus	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] LinNmChannelConfig (Bus LIN1) start	
Step 2	[SWC] SWC requests the network by calling ComM_RequestComMode(<user>,COMM_FULL_COMMUNICATION)	
Step 3	[SWC] Send frame on LIN1 Network by calling Rte_Write() on associated signal	[LT] Check that frames have been received on the Test Bench side
Step 4	[SWC] SWC releases the network by calling ComM_RequestComMode(<user>,COMM_NO_COMMUNICATION)	[LT] The SUT stops to send frames on LIN1 network
Step 5	-	[SWC] ComM service GetCurrentComMode(<user>) shall return E_OK Current ComMode shall be COMM_NO_COMMUNICATION
Post-conditions	None	

4.3.28 [ATS_COMM_00174] Switching the schedule table of the LIN bus when requested by the application

Test Objective	Switching the schedule table of the LIN bus when requested by the application		
ID	ATS_COMM_00174	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	LinIf, LinSM, LinNm	**State**	reviewed
Trace to Requirement			

on Acceptance Test Document	
Trace to SWS Item	LINInterface: SWS_LinIf_00444 LINInterface: SWS_LinIf_00028 LINInterface: SWS_LinIf_00393 LINInterface: SWS_LinIf_00495 LINInterface: SWS_LinIf_00029 LINInterface: SWS_LinIf_00397 LINInterface: SWS_LinIf_00485
Requirements / Reference to Test Environment	Configuration use case : UC02.03
Configuration Parameters	LinNmGlobalConfig(NmConfig.NmCluster.communicationCluster is associated to LinCluster): - LinNmPassiveModeEnabled(NmEcu.nmPassiveModeEnabled) = FALSE LinNmChannelConfig(NmCluster): Bus LIN1
Summary	The aim of this test is to verify that the SUT is able to switch the schedule table of the LIN bus when requested by the application.
Needed Adaptation to other Releases	None
Pre-conditions	The SUT is started.
Main Test Execution	
Test Steps	**Pass Criteria**
Step 1	[LT] LinNmChannelConfig (Bus LIN1) start
Step 2	[SWC] Request LIN network to go in FULL_COMMUNICATION by calling the API ComM_RequestComMode(<user>, COMM_FULL_COMMUNICATION)	[LT] Check that LIN Master (SUT) sends FRAME1, then FRAME2 as defined in the LIN_ScheduleTable_Base
Step 3	[SWC] Request ModeSwitch (call Rte_Switch associated to BswMMode port) to LIN_SCHEDULE_TABLE2_MODE (BswM Action configured to request Switch of schedule table to LinSMSchedule2)	[LT] Check that LIN MASTER (SUT) sends FRAME1, FRAME2 and FRAME3 as defined in the LIN_ScheduleTable2
Post-conditions	None

4.3.29 [ATS_COMM_00177] Starting the LIN bus when the network is requested by a LIN slave node

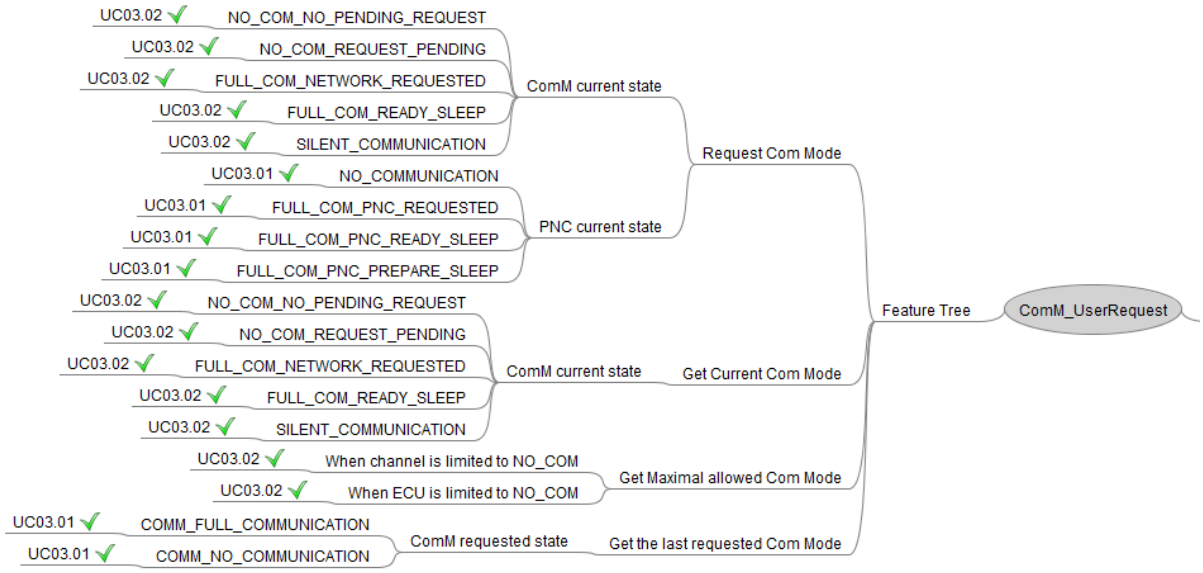
Test Objective	Starting the LIN bus when the network is requested by a LIN slave node		
ID	ATS_COMM_00177	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	LinNm, LinSM	**State**	reviewed
Trace to Requirement			

on Acceptance Test Document		
Trace to SWS Item	LINNetworkManagement: SWS_LinNm_00161 LINNetworkManagement: SWS_LinNm_00014 LINNetworkManagement: SWS_LinNm_00160	
Requirements / Reference to Test Environment	Configuration use case : UC02.03	
Configuration Parameters	LinNmGlobalConfig(NmConfig.NmCluster.communicationCluster is associated to LinCluster): - LinNmPassiveModeEnabled(NmEcu.nmPassiveModeEnabled) = FALSE LinNmChannelConfig(NmCluster): Bus LIN1 LinSMChannel(LinCluster): - LinSMSleepSupport(EcuInstance.sleepModeSupported) = TRUE	
Summary	The aim of this test is to verify that the SUT is able to wake up the bus when the network is requested by a LIN slave node.	
Needed Adaptation to other Releases	None	
Pre-conditions	The SUT is started (Communication Allowed for Bus LIN1 is FALSE).	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] LinNmChannelConfig (Bus LIN1) start	
Step 2	[LT] LIN_Slave_Node requests the Bus	[SWC] ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_NO_COMMUNICATION ComM_GetState returns COMM_NO_COM_REQUEST_PENDING (Use of CDD is needed to access to ComM_GetState)
Step 3	[SWC] Call BswM Action to Allow Communication (custom action configured to call ComM_CommunicationAllowed)	[LT] The SUT must start the LIN1 network
Post-conditions	None	

5 RS_BRF_01688 – ComM User Request

5.1 General Test Objective and Approach

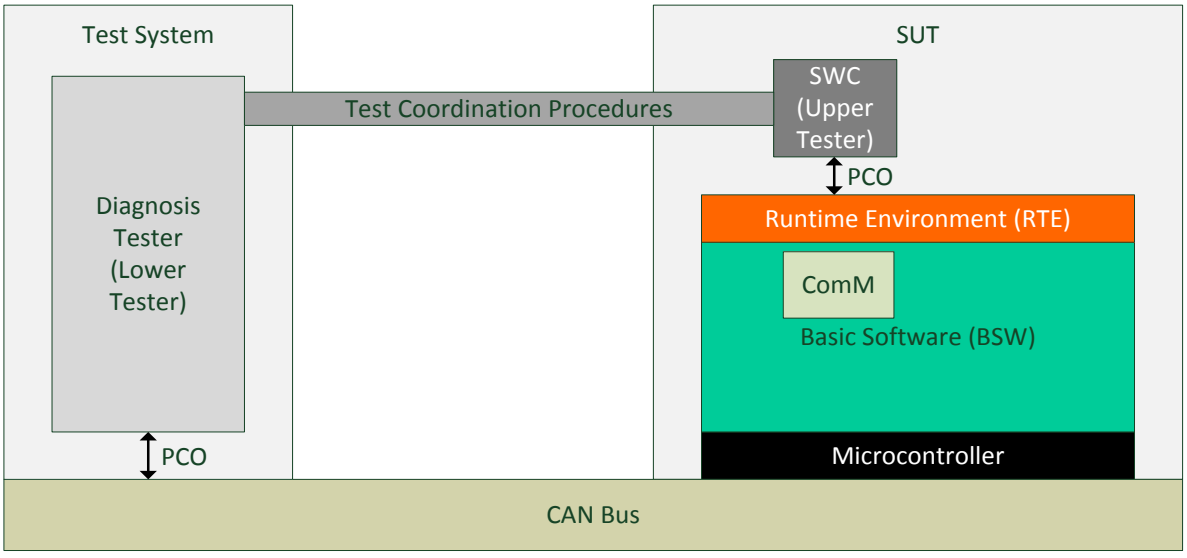
This test specification intends to cover the ComM services features.
This test case document has been established to cover the following features:



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

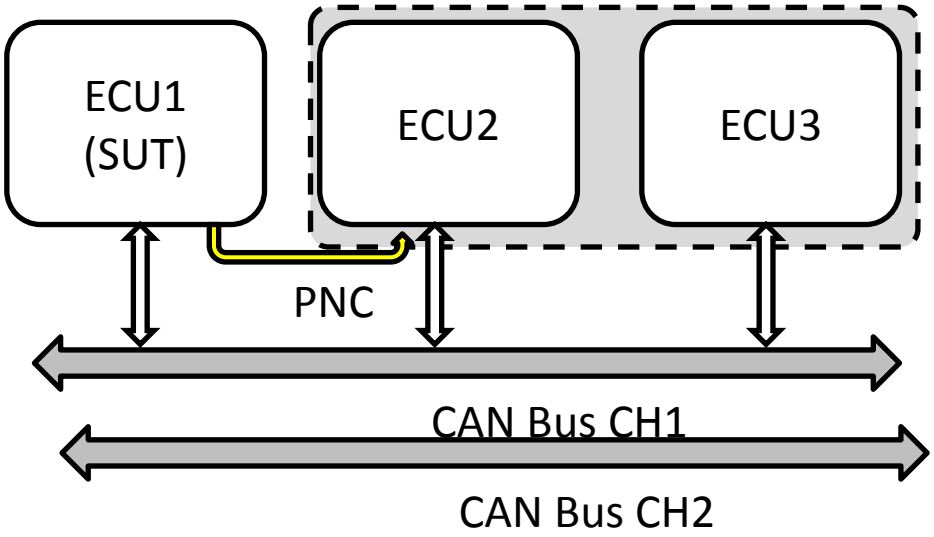
5.1.1.1 Overview on Architecture



The test system architecture consists of testing the two state machine of the ComM: Channel state machine and Pnc state machine.

To perform this, the Test Bench will simulate 2 ECUs and SUT produces either signals to Partial Network Clusters and Full network.

The infrastructure is described below:



5.1.1.2 Specific Requirements

Not Applicable

5.1.1.3 Test Coordination Requirements

Not Applicable

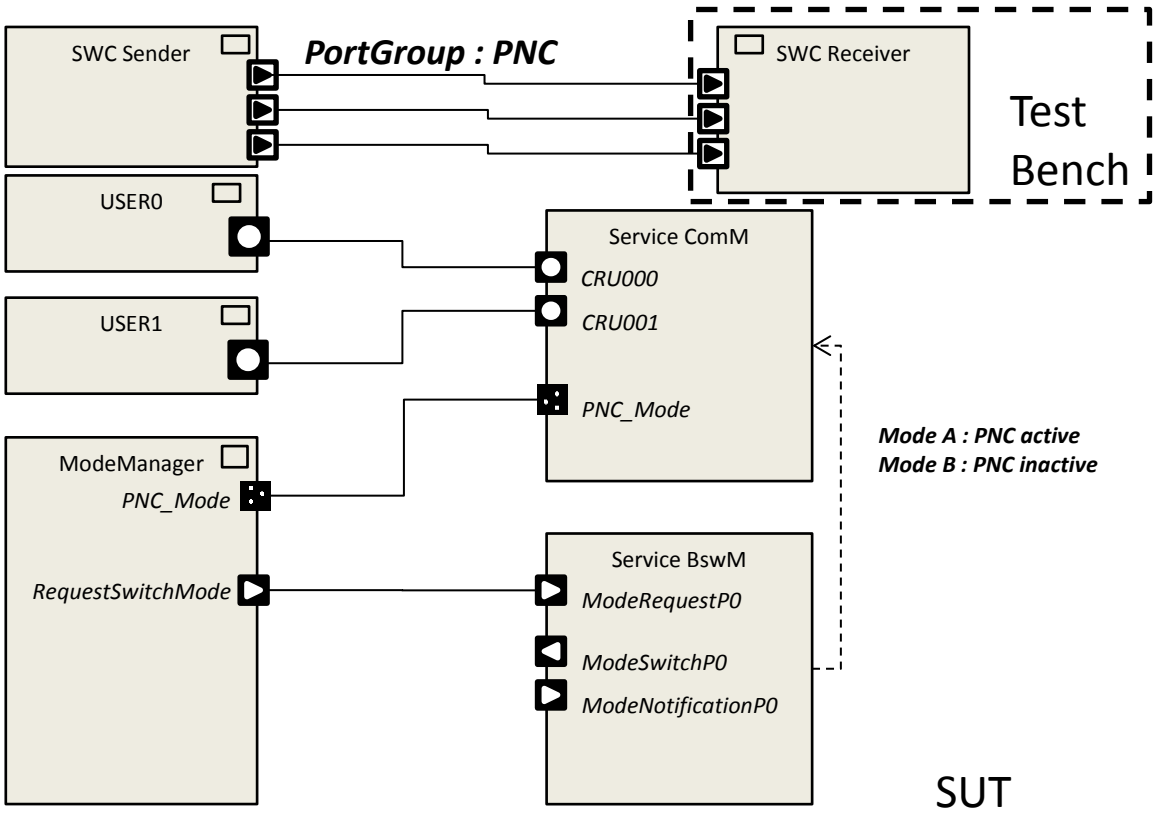
5.1.2 Test Configuration

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

Execution of the test cases requires one configuration set to be created. Two user is needed, one linked to a PN and then a channel, one other directly linked to a channel.

The SWC description is defined below:



Regarding the needed configuration on the SystemTemplate point of view:

- 1 CAN Cluster CAN_CH1_PNC 125kbps allocated to can controller 0
- 1 CAN cluster CAN_CH1 125kbps allocated to can controller 0
- 1 CAN cluster CAN_CH2 125kbps allocated to can controller 1

Even if the Test suite focuses on ComM Service state machine (NO_COMM, FULL_COMM, …etc…), the SWC description creates :

- 1 Sender SWC on SUT side that sends data on SR interface to Test Bench through a PortGroup mapped on the PNC on CAN CH1
- 1 Sender SWC on SUT side that sends data on SR interface to Test Bench through a Signal mapped on CAN CH1
- 1 Sender SWC on SUT side that sends data on SR interface to Test Bench through a Signal mapped on CAN CH2

User Requests done on CRU000 corresponds to CH1 and user requests done on CRU001 corresponds to CH2.

5.1.2.2 Required ECU Configuration Description Files

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

5.1.2.3 Required Software Component Description Files

No specific configuration requirements for Software Components.

5.1.2.4 Mandatory vs. Customizable Parts

In the configuration set, the only mandatory definition is to have at least 2 User, one with PNC, one other without PNC.

5.1.3 Test Case Design

Not Applicable

5.2 Re-usable Test Steps

Not Applicable

5.3 Test Cases

5.3.1 [ATS_COMM_00189] Getting the current communication mode

Test Objective	Getting the current communication mode		
ID	ATS_COMM_00189	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00111 ATR: ATR_ATR_00038		
Trace to SWS Item	COMManager: SWS_ComM_00083 COMManager: SWS_ComM_00176 COMManager: SWS_ComM_00798		
Requirements / Reference to Test Environment	Configuration use case : UC03.02		
Configuration Parameters	No Partial Network activated one Network User 2 network channels		
Summary	The aim of this test is to verify that the SUT is able to provide the current communication mode using the Client Server API GetCurrentComMode().		
Needed Adaptation to other Releases	None		
Pre-conditions	- Communication on the channel must be enabled (Call ComM_CommunicationAllowed(TRUE)) - The ComM parameter ComMEcuGroupClassification must be set to 0xFF		
Main Test Execution			
Test Steps		**Pass Criteria**	
Step 1	[CP] Power on SUT		
Step 2	[SWC] Call ComM_GetCurrentComMode()	[SWC] ComM_GetCurrentComMode should return E_OK	

		Current Com Mode should be COMM_NO_COMMUNICATION
Step 3	[SWC] Send request for changing ComM mode to FULL COMMUNICATION for User.	[SWC] RTE shall return E_OK.
Step 4	[SWC] Send request for current mode of ComM User.	[SWC] ComM mode shall be in FULL COMMUNICATION.
Step 5	[SWC] Call ComM Service ComM_ECUModeLimitation() to COMM_NO_COMMUNICATION	
Step 6	[SWC] Call ComM Service ComM_RequestComMode() with COMM_FULL_COMMUNICATION	
Step 7	[SWC] Call ComM Service ComM_GetCurrentComMode()	[SWC] ComM_GetCurrentComMode should return E_OK Current Com Mode should be COMM_NO_COMMUNICATION
Step 8	[SWC] Call ComM_ECUModeLimitation() to COMM_FULL_COMMUNICATION	
Step 9	[SWC] Call ComM Service ComM_RequestComMode() with COMM_FULL_COMMUNICATION	
Step 10	[CP] Wait n seconds (wait for the network to be started - Bus dependent time)	
Step 11	[SWC] Call ComM_GetCurrentComMode()	[SWC] ComM_GetCurrentComMode should return E_OK Current Com Mode should be COMM_FULL_COMMUNICATION
Step 12	[SWC] Switch Mode to disable ComAllowed=FALSE	
Step 13	[SWC] Call ComM Service ComM_RequestComMode() with COMM_NO_COMMUNICATION	
Step 14	[CP] Wait n seconds (wait for the network to be stopped - Bus dependent time)	
Step 15	[SWC] Call ComM Service ComM_RequestComMode() with COMM_FULL_COMMUNICATION	
Step 16	[CP] Wait n seconds (wait for the network to be started - Bus dependent time)	
Step 17	[SWC] Call ComM_GetCurrentComMode()	[SWC] ComM_GetCurrentComMode should

		return E_OK Current Com Mode should be COMM_NO_COMMUNICATION
Post-conditions	None	

5.3.2 [ATS_COMM_00190] Getting the maximal allowed communication mode

Test Objective	Getting the maximal allowed communication mode		
ID	ATS_COMM_00190	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00111 ATR: ATR_ATR_00038		
Trace to SWS Item	COMManager: SWS_ComM_00303 COMManager: SWS_ComM_00841 COMManager: SWS_ComM_00085 COMManager: SWS_ComM_00163 COMManager: SWS_ComM_00124		
Requirements / Reference to Test Environment	Configuration use case : UC03.02		
Configuration Parameters	No Partial Network activated one Network User 2 network channels		
Summary	The aim of this test is to verify that the SUT is able to provide the maximal allowed communication mode. There must be 2 main parts in this test : * First part : Test that the SUT can provide the maximal allowed communication mode when one channel is limited to NO_COM * Second part : Test that the SUT can provide the maximal allowed communication mode when the whole ECU is limited to NO_COM		
Needed Adaptation to other Releases			
Pre-conditions	SUT is started		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[CP] Power on SUT		
Step 2	[SWC] Call ComM_LimitChannelToNoComMode to enable limit channel to COMM_NO_COMMUNICATION		[SWC] ComM_LimitChannelToNoComMode should return E_OK
Step 3	[SWC] Call ComM_GetMaxComMode()		[SWC] ComM_GetMaxComMode should return E_OK Max Communication Mode should be COMM_NO_COMMUNICATION

Step 4	[SWC] Send request for changing ComM mode to FULL COMMUNICATION for User.	[SWC] RTE shall return E_OK.
Step 5	[SWC] Send request for current mode of ComM User.	[SWC] ComM mode shall be in FULL COMMUNICATION.
Step 6	[SWC] Call ComM_LimitChannelToNoComMode() to disable limit channel to COMM_NO_COMMUNICATION	[SWC] ComM_LimitChannelToNoComMode should return E_OK
Step 7	[SWC] Call ComM_GetMaxComMode()	[SWC] ComM_GetMaxComMode should return E_OK Max Communication Mode should be COMM_FULL_COMMUNICATION
Step 8	[SWC] Call ComM_LimitECUToNoComMode() to enable limit ECU to COMM_NO_COMMUNICATION	[SWC] ComM_LimitECUToNoComMode should return E_OK
Step 9	[SWC] Call ComM_GetMaxComMode()	[SWC] ComM_GetMaxComMode should return E_OK Max Communication Mode should be COMM_NO_COMMUNICATION
Step 10	[SWC] Call ComM_LimitECUToNoComMode() to disable limit ECU to COMM_NO_COMMUNICATION	[SWC] ComM_LimitECUToNoComMode should return E_OK
Step 11	[SWC] Call ComM_GetMaxComMode()	[SWC] ComM_GetMaxComMode should return E_OK Max Communication Mode should be COMM_FULL_COMMUNICATION
Post-conditions	None	

5.3.3 [ATS_COMM_00191] Getting the last requested communication mode

Test Objective	Getting the last requested communication mode		
ID	ATS_COMM_00191	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00111 ATR: ATR_ATR_00038		
Trace to SWS Item	COMManager: SWS_ComM_00924 COMManager: SWS_ComM_00079		
Requirements / Reference	Configuration use case : UC03.01		

to Test Environment		
Configuration Parameters	A Partial Network cluster activated allocated to 2 network channels one Network User 2 network channels	
Summary	The aim of this test is to verify that the SUT is able to provide the last requested communication mode. This test must verify when the last ComM requested state was COMM_FULL_COMMUNICATION and COMM_NO_COMMUNICATION.	
Needed Adaptation to other Releases	None	
Pre-conditions	SUT is started	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[CP] Power on SUT	
Step 2	[SWC] Call ComM_GetRequestedComMode()	[SWC] Requested Communication Mode should COMM_NO_COMMUNICATION (default value as no request before call to ComM_GetRequestedComMode())
Step 3	[SWC] Send request for changing COM mode to FULL COMMUNICATION for the User.	[SWC] RTE shall return E_OK.
Step 4	[SWC] Send request for the current mode of ComM User.	[SWC] ComM mode shall be in FULL COMMUNICATION.
Step 5	[SWC] Call ComM_ChannelLimitation() with Status = TRUE (to limit channel to COMM_NO_COMMUNICATION state).	
Step 6	[SWC] Call ComM_RequestComMode() COMM_NO_COMMUNICATION	
Step 7	[SWC] Execute ComM_GetRequestedComMode()	[SWC] ComM_GetRequestedComMode should return E_OK Requested Com Mode should be COMM_NO_COMMUNICATION
Step 8	[SWC] Call ComM_RequestComMode() to COMM_FULL_COMMUNICATION	
Step 9	[SWC] Call ComM_GetCurrentComMode()	[SWC] Current Com Mode should be COMM_NO_COMMUNICATION (as channel limitation is active)

Step 10	[SWC] Call ComM_GetRequestedComMode()	[SWC] ComM_GetRequestedComMode should return E_OK Requested Com Mode should be COMM_FULL_COMMUNICATION
Step 11	[SWC] Call ComM_ChannelLimitation() with Status=FALSE	
Step 12	[SWC] Call ComM_RequestComMode() to COMM_FULL_COMMUNICATION	
Step 13	[SWC] Call ComM_GetCurrentComMode()	[SWC] Current Com Mode should be COMM_FULL_COMMUNICATION (as channel limitation is inactive and network requested)
Step 14	[SWC] Call ComM_GetRequestedComMode()	[SWC] ComM_GetRequestedComMode should return E_OK Requested Com Mode should be COMM_FULL_COMMUNICATION
Post-conditions	None	

5.3.4 [ATS_COMM_00266] Requesting/Releasing Network when Diag session is active

Test Objective	Requesting/Releasing Network when Diag session is active		
ID	ATS_COMM_00266	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	Rte, Dcm, ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00111 ATR: ATR_ATR_00038		
Trace to SWS Item	COMManager: SWS_ComM_00876 COMManager: SWS_ComM_00870 COMManager: SWS_ComM_00889 COMManager: SWS_ComM_00888 COMManager: SWS_ComM_00890 COMManager: SWS_ComM_00883 COMManager: SWS_ComM_00182		
Requirements / Reference to Test Environment	Use case 03.02		
Configuration Parameters	1 Swc user that is connected to ComM service to request/release the network.		
Summary	This test verifies that Network keep the FULL_COMMUNICATION state when Diag session is active on channel even if all users have released the network.		

Needed Adaptation to other Releases	None	
Pre-conditions	SUT is started. Channel is in COMM_NO_COMMUNICATION state.	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Call to ComM_GetCurrentComMode()	[SWC] CurrentComMode should be equal to COMM_NO_COMMUNICATION
Step 2	[LT] Active Diagnostic session	
Step 3	[SWC] Call to ComM_GetCurrentComMode()	[SWC] CurrentComMode should be equal to COMM_FULL_COMMUNICATION
Step 4	[LT] Close Diagnostic session	
Step 5	[SWC] Call to ComM_GetCurrentComMode()	[SWC] CurrentComMode should be equal to COMM_NO_COMMUNICATION
Step 6	[SWC] Call to ComM_RequestComMode(<user>, COMM_FULL_COMMUNICATION)	
Step 7	[SWC] Call to ComM_GetCurrentComMode()	[SWC] CurrentComMode should be equal to COMM_FULL_COMMUNICATION
Step 8	[LT] Active Diagnostic Session	
Step 9	[SWC] Call to ComM_GetCurrentComMode()	[SWC] CurrentComMode should be equal to COMM_FULL_COMMUNICATION
Step 10	[SWC] Call to ComM_RequestComMode(<user>, COMM_NO_COMMUNICATION)	
Step 11	[SWC] Call to ComM_GetCurrentComMode()	[SWC] CurrentComMode should be equal to COMM_FULL_COMMUNICATION (as Diagnostic session still active)
Step 12	[LT] Close Diagnostic session	
Step 13	[SWC] Call to ComM_GetCurrentComMode()	[SWC] CurrentComMode should be equal to COMM_NO_COMMUNICATION (as no pending user requests)
Post-conditions	none.	

5.3.5 [ATS_COMM_00770] FULL/NO Communication Request By SW-C Before Processing Old Request

Test Objective	FULL/NO Communication Request By SW-C Before Processing Old Request

ID	ATS_COMM_00770	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	BswM,ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	BSWModeManager: SWS_BswM_00038 COMManager: SWS_ComM_00500 COMManager: SWS_ComM_00897 COMManager: SWS_ComM_00847		
Requirements / Reference to Test Environment	none		
Configuration Parameters	ComM user = SWC Channel 0		
Summary	If user requests "Full or No communication" mode change more than once then the latest user request shall overwrite the old request even if the old request is not processed.		
Needed Adaptation to other Releases			
Pre-conditions	DUT shall be initialized		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[SWC] Request ComM to be in Full Communication		[SWC] RTE_E_OK shall be returned
Step 2	[SWC] Request ComM to be in Full Communication before ComM main function period elapses		[SWC] RTE_E_OK shall be returned
Step 3	[SWC] Request ComM for the current mode for the associated channel		[SWC] COMM_NO_COMMUNICATION shall be returned
Step 4	[SWC] Request ComM for the current mode for the associated channel		[SWC] COMM_NO_COMMUNICATION shall be returned
Post-conditions	NONE		

5.3.6 [ATS_COMM_00774] Communication Inhibition Functionality Of ComM

Test Objective	Communication Inhibition Functionality Of ComM		
ID	ATS_COMM_00774	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			

Trace to SWS Item	COMManager: SWS_ComM_00215 COMManager: SWS_ComM_00740 COMManager: SWS_ComM_00741	
Requirements / Reference to Test Environment	none	
Configuration Parameters	User defined configuration reference: ComM User = SWC<1>	
Summary	The purpose of mode inhibition is to limit the communication capabilities. When mode inhibition is enabled the ComM module shall not process the user requests instead the status of the user requests will be stored.	
Needed Adaptation to other Releases		
Pre-conditions	ComM channel shall be in Full Communication mode	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Request the mode inhibition count value	[SWC] CounterValue shall be updated to “0”
Step 2	[SWC] Request ComM for Full Communication Mode	[SWC] COMM_E_MODE_LIMITATION shall be returned
Step 3	[SWC<1>] Request ComM to read the mode inhibition count value	[SWC<1>] CounterValue should be updated with 1 indicating that there is a FULL communication request by a user
Step 4	[SWC] Request ComM for Full Communication Mode	[SWC] COMM_E_MODE_LIMITATION shall be returned
Step 5	[SWC<1>] Request ComM to read the mode inhibition count value	[SWC<1>] CounterValue should be updated with 2 indicating that there are 2 FULL communication requests by a user
Step 6	[SWC] Request ComM to reset the mode inhibition count	[SWC] RTE_E_OK shall be returned
Step 7	[SWC<1>] Request ComM to read the mode inhibition count value	[SWC<1>] CounterValue shall be reset to “0”
Post-conditions	NONE	

5.3.7 [ATS_COMM_00775] Communication Inhibition Functionality Of ComM During Active Diagnostic Session

Test Objective	Communication Inhibition Functionality Of ComM During Active Diagnostic Session		
ID	ATS_COMM_00775	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed

Trace to Requirement on Acceptance Test Document		
Trace to SWS Item	COMManager: SWS_ComM_00182 COMManager: SWS_ComM_00215	
Requirements / Reference to Test Environment	none	
Configuration Parameters	User defined configuration reference: CANChannel= Ex:1 Transmission frame: data= 0x02 0x10 0x01 0x00 0x00 0x00 0x00 and 0x00 CANId = Ex: 0x200 and 0x201 Response frame: Data= 0x02 0x50 and 0x01 CANId= Ex: 0x200 and 0x201	
Summary	The purpose of mode inhibition is to limit the communication capabilities. When mode inhibition is enabled the ComM module shall not process the user requests instead the status of the user requests will be stored. The communication inhibition shall get temporarily inactive during an active diagnostic session and the corresponding channel shall be put into FULL communication mode.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Request ComM for Full Communication Mode	[SWC] COMM_E_MODE_LIMITATION shall be returned
Step 2	[LT] Transmit the diagnostic session request to the DUT	[LT] Positive response shall be observed
Step 3	[SWC] Request ComM for Full Communication Mode	[SWC] COMM_E_MODE_LIMITATION shall be returned No application messages shall be observed on the bus
Post-conditions	NONE	

5.3.8 [ATS_COMM_00776] Limiting A Channel To No Communication Mode During Active Diagnostic Session

Test Objective	Limiting A Channel To No Communication Mode During Active Diagnostic Session		
ID	ATS_COMM_00776	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed

Trace to Requirement on Acceptance Test Document		
Trace to SWS Item	COMManager: SWS_ComM_00182 COMManager: SWS_ComM_00303 COMManager: SWS_ComM_00752 COMManager: SWS_ComM_00743	
Requirements / Reference to Test Environment	none	
Configuration Parameters	ComMMode(Mode Inhibition) = Enabled User defined configuration reference: Counter value = 1 Transmit frame: data = 0x02 0x10 0x01 0x00 0x00 0x00 0x00 and 0x00 on CAN channel 1 with DLC value as 8 bytes Response frame: Data = 0x02 0x50 and 0x01 with CAN ID 0x200 0x201	
Summary	When a channel is in FULL communication mode and if there is a request to limit the channel to NO communication mode then the ComM module will clear the user requests and the corresponding channel enters NO communication mode. The communication inhibition shall get temporarily inactive during an active diagnostic session and the corresponding channel shall be put into FULL communication mode.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Request ComM for Full Communication Mode	[SWC] RTE_E_OK shall be returned
Step 2	[SWC] Request ComM for the current mode for the associated channel	[SWC] COMM_FULL_COMMUNICATION shall be updated
Step 3	[SWC<2>] Request ComM to Limit ChannelToNoCommunication.	[SWC<2>] RTE_E_OK shall be returned
Step 4	[SWC<2>] Request ComM for the current mode for the associated channel	[SWC<2>] COMM_NO_COMMUNICATION shall be updated
Step 5	[LT] Transmit a valid diagnostic session request frame.	[LT] The DUT shall send the positive response
Step 6	[SWC] Request ComM for Full Communication Mode	[SWC] COMM_E_MODE_LIMITATION shall be returned
Post-conditions	NONE	

5.3.9 [ATS_COMM_00777] Limiting ECU To No Communication Mode During Active Diagnostic Session

Test Objective	Limiting ECU To No Communication Mode During Active Diagnostic Session		
ID	ATS_COMM_00777	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00182 COMManager: SWS_ComM_00303 COMManager: SWS_ComM_00740 COMManager: SWS_ComM_00741		
Requirements / Reference to Test Environment	none		
Configuration Parameters	ComMMode(Mode Inhibition) = Enabled ComM users : User 1 <Channel1>, User 2 <Channel2> Transmit frame: data = 0x02 0x10 0x01 0x00 0x00 0x00 0x00 and 0x00 on CAN channel 1 with DLC value as 8 bytes Response frame: Data = 0x02 0x50 and 0x01 with CAN ID 0x200 0x20		
Summary	Configure SWC1 and SWC2 (User 1 and User 2) and map User 1 to Channel 1, User 2 to Channel 2. When the channels are in FULL communication mode and if there is a request to limit the ECU to NO communication mode then the ComM module will clear the user requests and all the configured channels enters NO communication mode. The communication inhibition shall get temporarily inactive during an active diagnostic session and the corresponding channel shall be put into FULL communication mode.		
Needed Adaptation to other Releases			
Pre-conditions	ComM channel shall be in Full Communication mode		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[SWC<1>] Request ComM for Full Communication Mode for User 1.		[SWC<1>] RTE_E_OK shall be returned
Step 2	[SWC<1>] Request ComM for the current mode for the associated Channel 1.		[SWC<1>] COMM_FULL_COMMUNICATION shall be updated.
Step 3	[SWC<2>] Request ComM for Full Communication Mode for User 2.		[SWC<2>] RTE_E_OK shall be returned.
Step 4	[SWC<2>] Request ComM for the current mode for the associated Channel 2.		[SWC<2>] COMM_FULL_COMMUNICATION shall be updated

Step 5	[SWC] Request ComM to Limit ECUToNoCommunication.	[SWC] RTE_E_OK shall be returned.
Step 6	[SWC<1>] Request ComM current mode for the associated Channel 1.	[SWC<1>] COMM_NO_COMMUNICATION shall be updated.
Step 7	[SWC<2>] Request ComM current mode for the associated Channel 2.	[SWC<2>] COMM_NO_COMMUNICATION shall be updated.
Step 8	[LT] Transmit a valid diagnostic session request frame on Channel 1.	[LT] The DUT shall send the positive response.
Step 9	[SWC<1>] Request ComM for Full Communication for User 1.	[SWC<1>] COMM_E_MODE_LIMITATION shall be returned
Step 10	[LT] Transmit a valid diagnostics session request frame on Channel 2.	[LT] The DUT shall send positive response.
Step 11	[SWC<2>] Request ComM for Full Communication mode for User 2.	[SWC<2>] COMM_E_MODE_LIMITATION shall be returned.
Step 12	-	[LT] Application messages shall not be observed on Channel 1, 2.
Post-conditions	NONE	

5.3.10 [ATS_COMM_00816] Testing Whether Limiting ComM Channel To No Communication Is Possible Only When The Current State Is Comm_Full_Com_Network_Requested

Test Objective	Testing Whether Limiting ComM Channel To No Communication Is Possible Only When The Current State Is Comm_Full_Com_Network_Requested		
ID	ATS_COMM_00816	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00841 COMManager: SWS_ComM_00842		
Requirements / Reference to Test Environment	none		

Configuration Parameters	ComMNmVariant = FULL COMM_MODE_LIMITATION_ENABLED = TRUE	
Summary	Bring the Integrated Stack to Full communication. A callout is created to indicate the occurrence of COMM_FULL_COM_NETWORK_REQUESTED. Once it is confirmed limit the channel 0 to NO COMMUNICATION.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Send the request for changing COM mode to Full communication for user0	[SWC] RTE shall return E_OK
Step 2	[SWC] Send the request for the current mode of ComM for user0	[SWC] ComM mode shall be in Full communication
Step 3	[SWC] Send the request for limiting channel 0 to No communication	[SWC] RTE shall return E_OK
Step 4	[SWC] Send the request for changing COM mode to Full communication for user0	[SWC] RTE Returns COMM_E_MODE_LIMITATION for the request
Post-conditions	NONE	

5.3.11 [ATS_COMM_00818] Behaviour Of ComM Channel When It Is In Mode Limitation And The Two Users Mapped To This Channel Requests For Different Communication Modes

Test Objective	Behaviour Of ComM Channel When It Is In Mode Limitation And The Two Users Mapped To This Channel Requests For Different Communication Modes		
ID	ATS_COMM_00818	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00488		
Requirements / Reference to Test Environment	none		
Configuration Parameters	COMM_CHANNEL_ID = 0 COMM_MODE_LIMITATION_E-BLED = TRUE		
Summary	Configure User 0, User 1 and map both the Users to channel 0. Request Full Communication from User 0. In the next step request channel limitation to No Communication for channel 0. Request Full Communication from User 0 which shall		

	return E_MODE_LIMITATION. Request No Communication from User 1. After the confirmation of No Communication, request for disable mode limitation. Check current communication mode for User 0 and User 1, then both the Users shall be in No Communication mode.
Needed Adaptation to other Releases	
Pre-conditions	DUT shall be initialized

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Send the request for changing COM mode to Full communication for user0	[SWC] RTE shall return E_OK
Step 2	[SWC] Send the request for the current mode of ComM for user0	[SWC] ComM mode shall be in Full communication
Step 3	[SWC] Send the request for limiting channel 0 to No communication	[SWC] RTE shall return E_OK
Step 4	[SWC] Send the request for changing COM mode to Full communication for user0	[SWC] RTE Returns COMM_E_MODE_LIMITATION for the request
Step 5	[SWC] Send the request for changing COM mode to No communication for user1	[SWC] RTE shall return E_OK
Step 6	[SWC] Disable the mode limitation	-
Step 7	[SWC] Send the request for the current mode of ComM for user0	[SWC] ComM mode shall be in No Communication
Step 8	[SWC] Send the request for the current mode of ComM for user1	[SWC] ComM mode shall be in No Communication
Post-conditions	NONE	

5.3.12 [ATS_COMM_00819] Enabling And Disabling Mode Inhibition To The Channels Mapped To The Single User

Test Objective	Enabling And Disabling Mode Inhibition To The Channels Mapped To The Single User		
ID	ATS_COMM_00819	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00488 COMManager: SWS_ComM_00840		

Requirements / Reference to Test Environment	none
Configuration Parameters	COMM_USER = 1 COMM_CHANNEL_ID = 1 COMM_MODE_LIMITATION_ENABLED = TRUE COMM_NM_VARIANT = FULL
Summary	Configure user 0 and map channel 0 to it. Request full communication. In the next step request for no communication. Configure a callout for indicating the entry of silent communication. After the confirmation of silent communication request for full communication. Check for the current communication mode then the channel 0 will be in full communication mode.
Needed Adaptation to other Releases	
Pre-conditions	DUT shall be initialized

Main Test Execution

Test Steps		**Pass Criteria**
Step 1	[SWC] Send the request to change the value of the parameter ComMEcuGroupClassification	[SWC] RTE shall return E_OK
Step 2	[SWC] Send the request for changing COM mode to Full communication for user0	[SWC] RTE shall return E_OK
Step 3	[SWC] Send the request for the current mode of ComM for user0	[SWC] ComM mode shall be in Full communication
Step 4	[SWC] Request to limit channel 0 to No communication	[SWC] RTE shall return E_OK
Step 5	[SWC] Request to ComM to keep the DUT in Full Communication for user 0	[SWC] COMM_E_MODE_LIMITATION is returned for the request
Step 6	[SWC] Send the request for changing COM mode to Full communication for user0	[SWC] ComM shall be in No communication mode
Step 7	[SWC] Send the request for changing COM mode to Full communication for user1	[SWC] ComM mode shall be in Full communication
Step 8	[SWC] Send the request to release the mode inhibition for the channel 0	[SWC] E_OK shall be returned for the request
Step 9	[SWC] Send the request for changing COM mode to Full communication for user0	[SWC] RTE shall return E_OK
Step 10	[SWC] Send the request for the current mode of ComM for user0	[SWC] ComM mode shall be in Full communication
Step 11	[SWC] Request to limit channel 0 to No communication	-
Step 12	[SWC] Send request to ComM to keep the DUT in Full Communication for user 0	[SWC] Return with Error (COMM_E_MODE_LIMITATION)

Step 13	[SWC] Send request to Release the mode inhibition for the channel 1	-
Step 14	[SWC] Request ComM for the current mode for user 1	[SWC] ComM mode shall be in Full communication
Post-conditions	NONE	

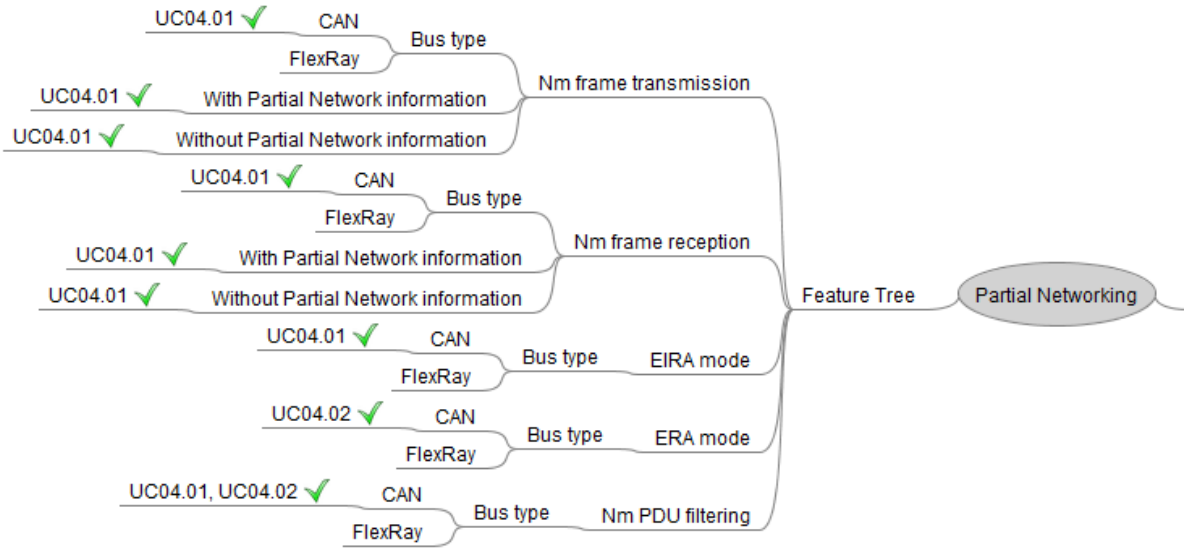
6 RS_BRF_01696 - Partial Networking

6.1 General Test Objective and Approach

This Test Specification intends to cover the Partial Networking feature as described in the AUTOSAR Feature [RS_BRF_01696].

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:



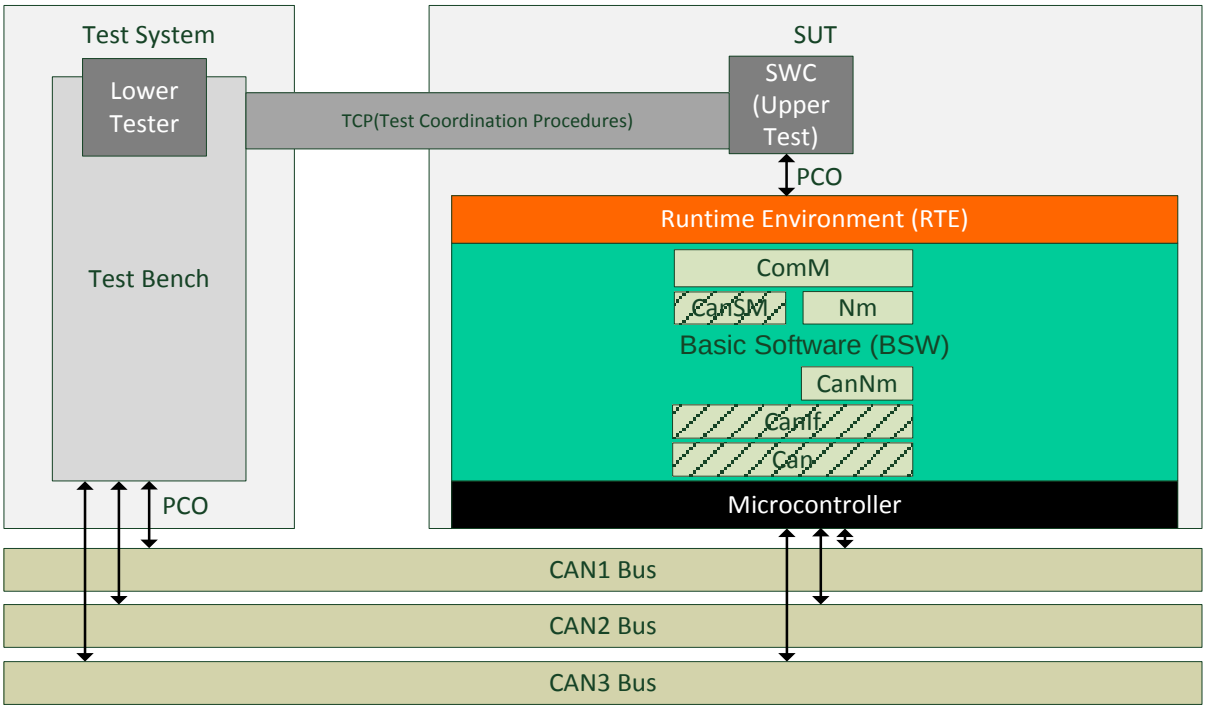
This specification gives the description of required tests environments (test bench, uses case, arxml files) and detailed tests cases for executing tests.

6.1.1 Test System

6.1.1.1 Overview on Architecture

6.1.1.1.1 Use case 04.01: Partial networking on CAN bus with SUT configured in EIRA mode

For this use case, the aim is to test the partial networking feature on three CAN busses when the SUT is configured in EIRA mode.



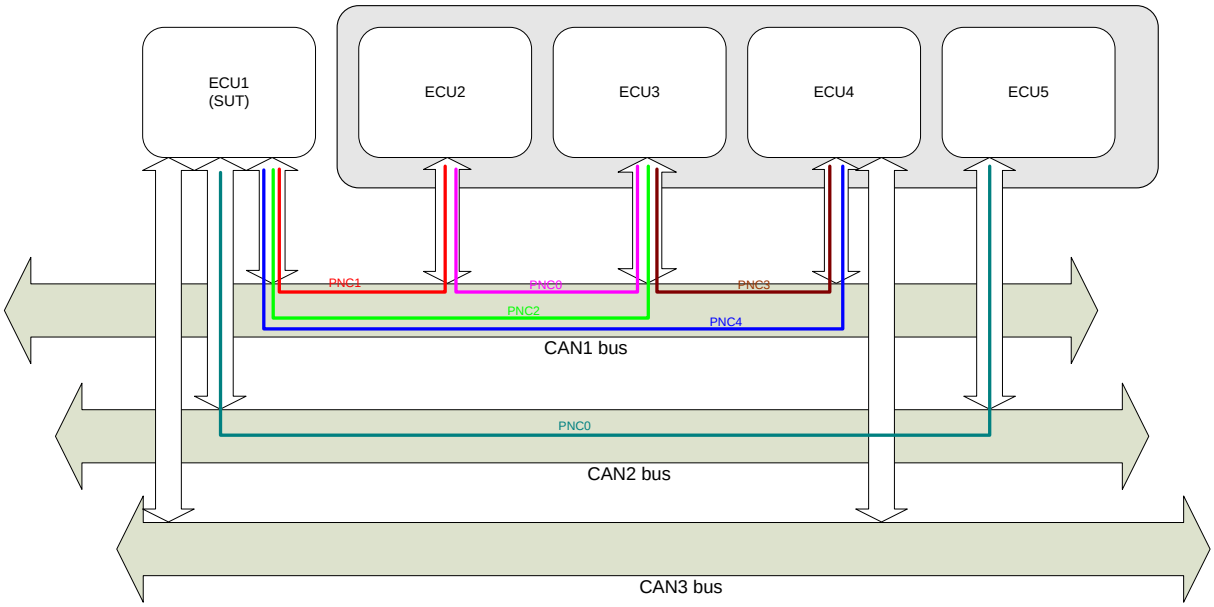
The test system architecture consists of Test Bench that executes only test sequencer and gives actions request through Test coordination Procedures to embedded SWC.

Three modules are only used for test but they are not in the scope of the tests:

- CanSM: Used to start and stop the CAN busses
- CanIf: Used to transmit the PDUs over the CAN busses
- Can: Used to drive the CAN controllers

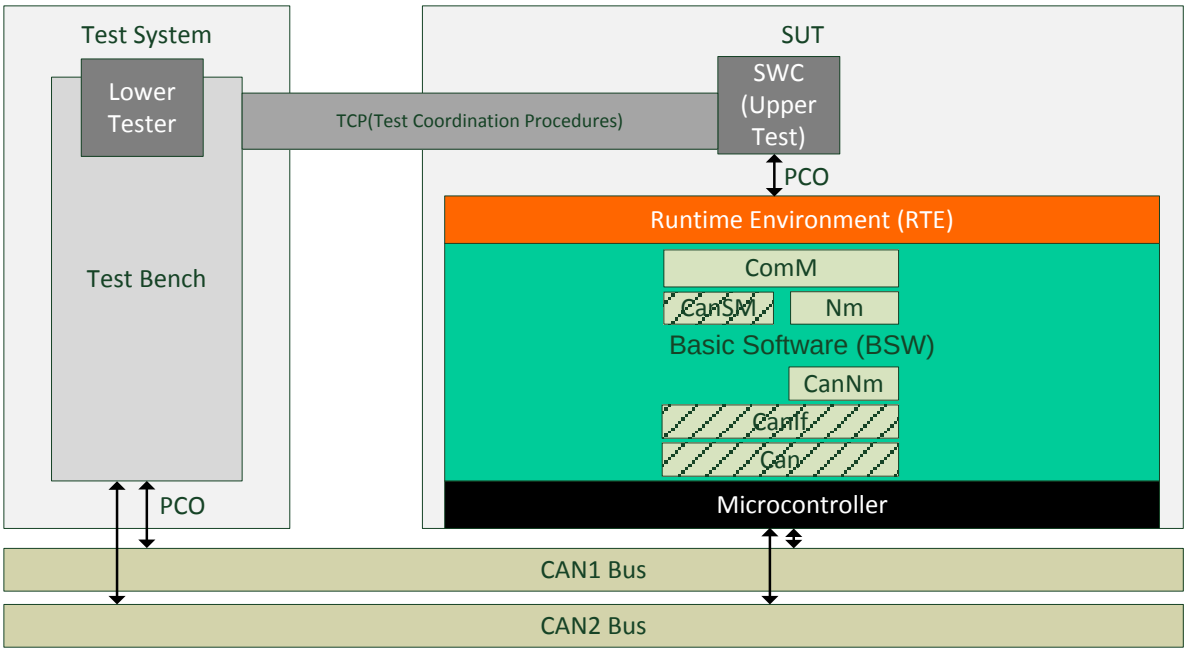
In this use case, the test bench will simulate 4 ECUs which will be connected to the different CAN busses.

The infrastructure is described below:



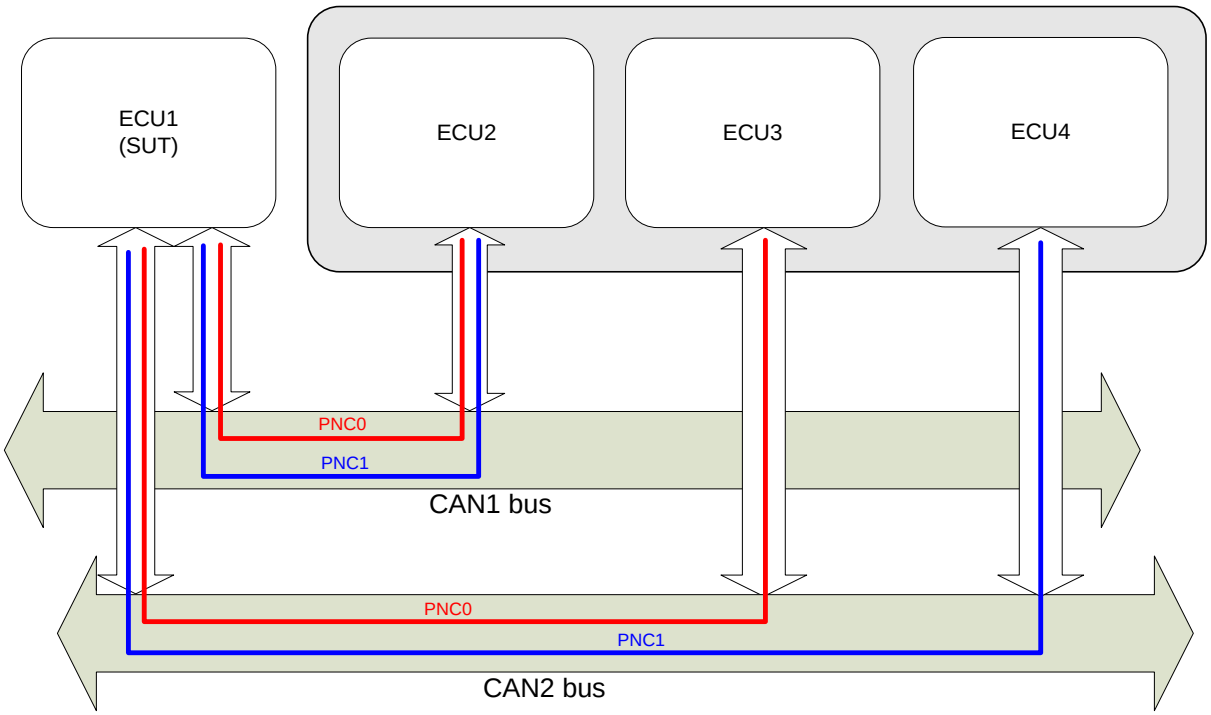
6.1.1.1.2 Use case 04.02: Partial networking on CAN bus with SUT configured in ERA mode

For this use case, the aim is to test the partial networking feature on two CAN busses when the SUT is configured in ERA mode (Gateway ECU).



In this use case, the test bench will simulate 3 ECUs which will be connected to the different CAN busses. The SUT will be only the gateway from the CAN1 bus to the CAN2 bus.

The infrastructure is described below:



6.1.1.2 Specific Requirements

Not Applicable

6.1.1.3 Test Coordination Requirements

Not Applicable

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.

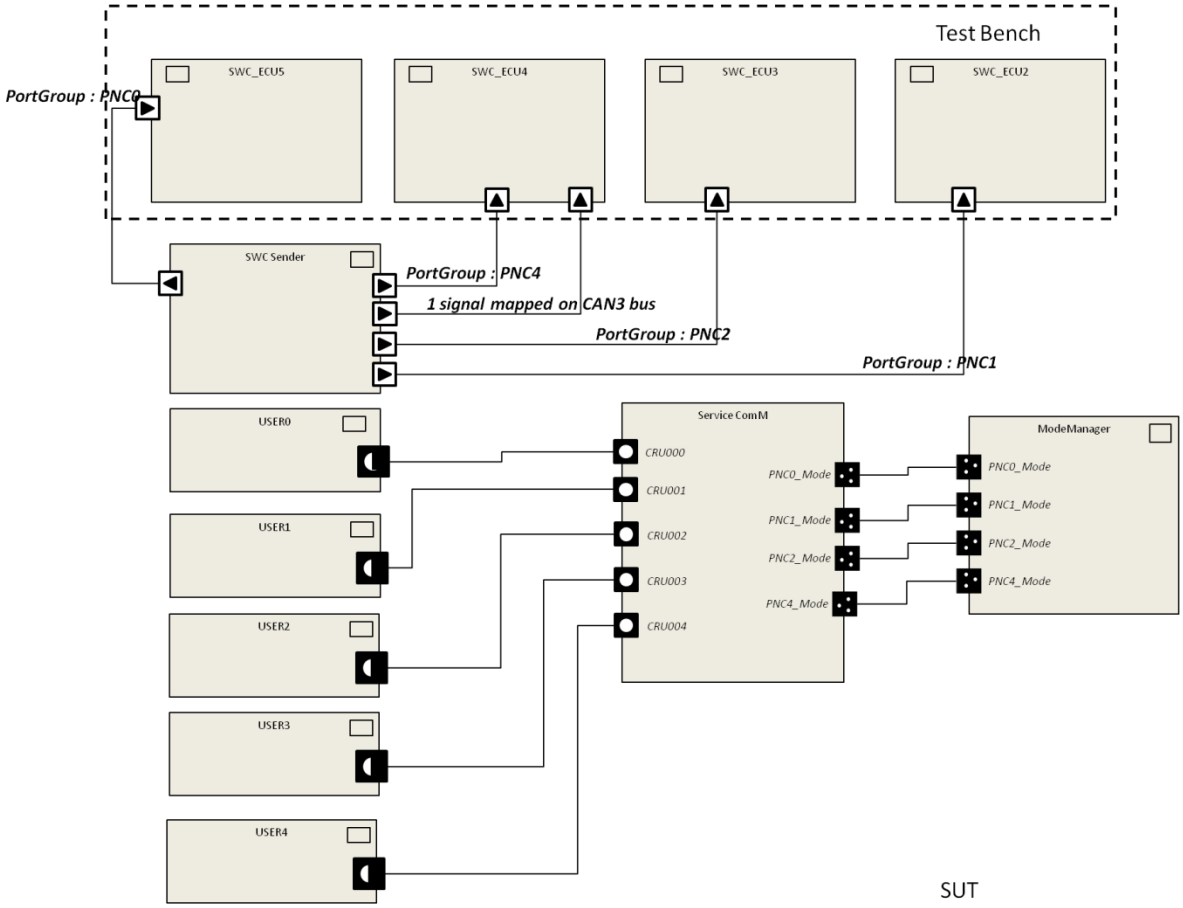
6.1.2.1 Required ECU Extract of System Description Files

There is one configuration set for each use case defined. Here are the description of these configurations sets :

6.1.2.1.1 Use case 04.01

In this use case, there are multiple partial networks mapped on different CAN busses. 5 users are needed, one linked to each Partial Network, and one mapped on the CAN3 bus which have no Partial Network configured.

The SWC description is defined below:

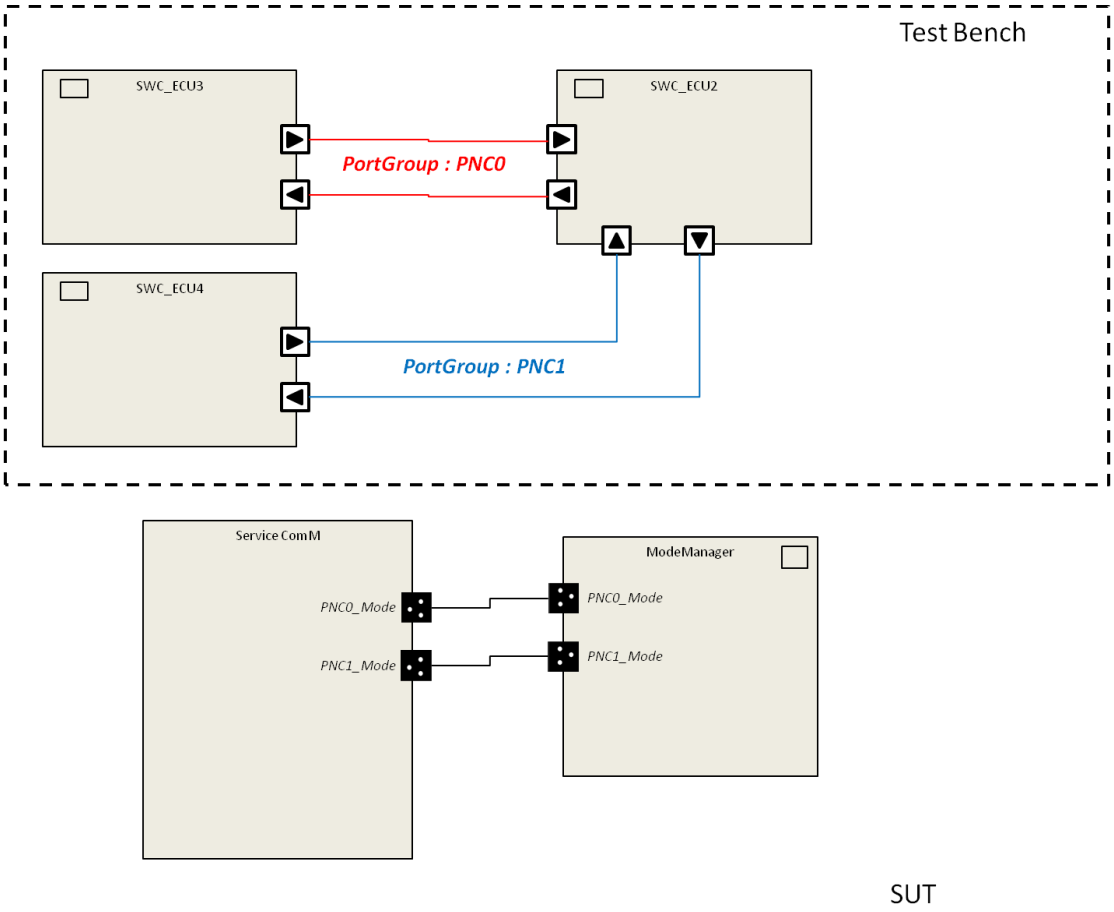


Here are the mapping of the users on Partial Networks :

- User requests done on CRU000 will request the Partial Network 0.
- User requests done on CRU001 will request the Partial Network 1.
- User requests done on CRU002 will request the Partial Network 2.
- User requests done on CRU003 will request the Partial Network 4.
- User requests done on CRU004 will request the CAN3 bus

6.1.2.1.2 Use case 04.02

In this use case, there is only one partial network which is gatewayed through the SUT. There is no user as the SUT is only used for gatewaying.



The PortGroup PNC0 must be configured to be gatewayed from the CAN1 bus to the CAN2 bus through the SUT in order to link the SWC_ECU2 with the SWC_ECU3.

The PortGroup PNC1 must be configured to be gatewayed from the CAN1 bus to the CAN2 bus through the SUT in order to link the SWC_ECU2 with the SWC_ECU4.

The communication database is depicted below:

IPduGroup	IPdu	Signal	Tx ECU	Rx ECU
PNC0_IpduGroup	PNC0_ECU3_To_ECU2_Ipdu	PNC0_Sg1	ECU3	ECU2
	PNC0_ECU2_To_ECU3_Ipdu	PNC0_Sg2	ECU2	ECU3
PNC1_IpduGroup	PNC0_ECU4_To_ECU2_Ipdu	PNC1_Sg1	ECU4	ECU2
	PNC0_ECU2_To_ECU4_Ipdu	PNC1_Sg2	ECU2	ECU4

6.1.2.2 Required ECU Configuration Description Files

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

6.1.2.3 Required Software Component Description Files

No specific configuration requirements for Software Components.

6.1.2.4 Mandatory vs. Customizable Parts

Mandatory parameters are listed in Tests Cases (see chapter 6.3 Test Cases).

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

- The different timeouts
- The activation of the Source node identifier
- The activation of the Car Wakeup feature
- CAN frames identifiers

6.1.3 Test Case Design

Not Applicable

6.2 Re-usable Test Steps

Not Applicable

6.3 Test Cases

6.3.1 [ATS_COMM_00178] Sending Partial Network Vector in EIRA mode when one Partial Network is requested by the application

Test Objective	Sending Partial Network Vector in EIRA mode when one Partial Network is requested by the application		
ID	ATS_COMM_00178	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CanNm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00114		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00413 CANNetworkManagement: SWS_CanNm_00427 CANNetworkManagement: SWS_CanNm_00430		
Requirements / Reference to Test Environment	Configuration use case : UC04.01		
Configuration Parameters	CanNmGlobalConfig(NmConfig.CanNmCluster): - CanNmPnEiraCalcEnabled(local) = TRUE - CanNmPnResetTime(local) is configured - CanNmPnEiraRxNSduRef(local) is configured		

	CanNmChannelConfig(CanNmCluster): Bus CAN1 - CanNmPnEnabled(PncMapping present) = TRUE
Summary	The aim of this test is to verify that the SUT (configured in EIRA mode) is able to send Nm frames periodically, with the PNV correctly set, when the network is requested by the application.
Needed Adaptation to other Releases	None
Pre-conditions	The SUT is started.
Main Test Execution	
Test Steps	**Pass Criteria**
Step 1	[LT] CanNmChannelConfig (Bus CAN1) start
Step 2	[SWC] Requests the Partial Network 1 by calling ComM_RequestComMode(<user_pnc1>, COMM_FULL_COMMUNICATION)	[LT] The Nm frame is sent periodically on the CAN1 bus - PNI bit is set to 1 - Partial Network 1 bit is set to 1
Post-conditions	None

6.3.2 [ATS_COMM_00179] Managing Partial Network Vector in EIRA mode when relevant PN bit received is equal to 1

Test Objective	Managing Partial Network Vector in EIRA mode when relevant PN bit received is equal to 1		
ID	ATS_COMM_00179	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CanNm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00114		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00411 CANNetworkManagement: SWS_CanNm_00412 CANNetworkManagement: SWS_CanNm_00415 CANNetworkManagement: SWS_CanNm_00416 CANNetworkManagement: SWS_CanNm_00417 CANNetworkManagement: SWS_CanNm_00418 CANNetworkManagement: SWS_CanNm_00419 CANNetworkManagement: SWS_CanNm_00420 CANNetworkManagement: SWS_CanNm_00423 CANNetworkManagement: SWS_CanNm_00424 CANNetworkManagement: SWS_CanNm_00425 CANNetworkManagement: SWS_CanNm_00426 CANNetworkManagement: SWS_CanNm_00428 CANNetworkManagement: SWS_CanNm_00429 CANNetworkManagement: SWS_CanNm_00431 CANNetworkManagement: SWS_CanNm_00432		
Requirements / Reference to Test Environment	Configuration use case : UC04.01		

Configuration Parameters	CanNmGlobalConfig(NmConfig.CanNmCluster): - CanNmPnEiraCalcEnabled(local) = TRUE - CanNmPnResetTime(local) is configured - CanNmPnEiraRxNSduRef(local) is configured CanNmChannelConfig(CanNmCluster): Bus CAN1 - CanNmPnEnabled(PncMapping present) = TRUE - CanNmAllNmMessagesKeepAwake(local) = FALSE	
Summary	The aim of this test is to verify that the SUT (configured in EIRA mode) is able to keep the bus awake when it receives Nm frames periodically with the PNV which is relevant to the SUT.	
Needed Adaptation to other Releases	None	
Pre-conditions	The SUT is started. <user_pnc1> ComM State equals to COMM_NO_COMMUNICATION	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] Bus CAN1 start	
Step 2	[LT] Send periodically the Nm frame with the Partial Network 1 bit to 1. NM frames Not relevant to SUT are also sent.	
Step 3	[CP] Wait <NM CYCLE TIME> + <safety margin>	[SWC] ComM service for <user_pnc1> GetCurrentComMode shall return E_OK Current ComMode shall be COMM_FULL_COMMUNICATION
Step 4	[LT] Send periodically the Nm frame with the Partial Network 1 bit to 0. NM frames Not relevant to SUT are also sent.	
Step 5	[CP] Wait <NM CYCLE TIME> + <safety margin>	[SWC] ComM service for <user_pnc1> GetCurrentComMode shall return E_OK Current ComMode shall be COMM_NO_COMMUNICATION
Post-conditions	None	

6.3.3 [ATS_COMM_00180] Managing Partial Network Vector in EIRA mode when all received relevant PN bits are equal to 0

Test Objective	Managing Partial Network Vector in EIRA mode when all received relevant PN bits are equal to 0		
ID	ATS_COMM_00180	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CanNm	**State**	reviewed

Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00114
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00412 CANNetworkManagement: SWS_CanNm_00415 CANNetworkManagement: SWS_CanNm_00416 CANNetworkManagement: SWS_CanNm_00417 CANNetworkManagement: SWS_CanNm_00418 CANNetworkManagement: SWS_CanNm_00420 CANNetworkManagement: SWS_CanNm_00423 CANNetworkManagement: SWS_CanNm_00424 CANNetworkManagement: SWS_CanNm_00425 CANNetworkManagement: SWS_CanNm_00426 CANNetworkManagement: SWS_CanNm_00428 CANNetworkManagement: SWS_CanNm_00429 CANNetworkManagement: SWS_CanNm_00431 CANNetworkManagement: SWS_CanNm_00432
Requirements / Reference to Test Environment	Configuration use case : UC04.01
Configuration Parameters	CanNmGlobalConfig(NmConfig.CanNmCluster): - CanNmPnEiraCalcEnabled(local) = TRUE - CanNmPnEiraRxNSduRef(local) is configured CanNmChannelConfig(CanNmCluster): Bus CAN1 - CanNmPnEnabled(PncMapping present) = TRUE - CanNmAllNmMessagesKeepAwake(local) = FALSE - CanNmTimeoutTime(nmNetworkTimeout) is configured
Summary	The aim of this test is to verify that the SUT (configured in EIRA mode) is able to make the bus sleep when it receives Nm frames periodically with the PNV which is not relevant to the SUT.
Needed Adaptation to other Releases	None
Pre-conditions	The SUT is started. BUS request usage for PNC1, PNC2 and PNC4 (send Nm Frames periodically with PNV equals to PNC1, PNC2, PNC4)

Main Test Execution

Test Steps		**Pass Criteria**
Step 1	[LT] Send periodically a request to PNC3 only (not used by the SUT) by setting PNV to PNC3 value.	
Step 2	[CP] Wait <nmNetworkTimeout> + <safety margin>	[SWC] ComM service for <user_pnc1> GetCurrentComMode shall return E_OK Current ComMode shall be COMM_NO_COMMUNICATION ComM service for <user_pnc2> GetCurrentComMode shall return E_OK Current ComMode shall be COMM_NO_COMMUNICATION ComM service for <user_pnc4>

		GetCurrentComMode shall return E_OK Current ComMode shall be COMM_NO_COMMUNICATION
Post-conditions	None	

6.3.4 [ATS_COMM_00181] Managing Partial Network Vector in ERA mode when relevant PN bit received is equal to 1

Test Objective	Managing Partial Network Vector in ERA mode when relevant PN bit received is equal to 1		
ID	ATS_COMM_00181	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	<Bus>Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00114		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00411 CANNetworkManagement: SWS_CanNm_00412 CANNetworkManagement: SWS_CanNm_00415 CANNetworkManagement: SWS_CanNm_00416 CANNetworkManagement: SWS_CanNm_00417 CANNetworkManagement: SWS_CanNm_00418 CANNetworkManagement: SWS_CanNm_00419 CANNetworkManagement: SWS_CanNm_00420 CANNetworkManagement: SWS_CanNm_00434 CANNetworkManagement: SWS_CanNm_00435 CANNetworkManagement: SWS_CanNm_00436 CANNetworkManagement: SWS_CanNm_00437 CANNetworkManagement: SWS_CanNm_00438 CANNetworkManagement: SWS_CanNm_00439 CANNetworkManagement: SWS_CanNm_00442 CANNetworkManagement: SWS_CanNm_00443		
Requirements / Reference to Test Environment	Configuration use case : UC04.02		
Configuration Parameters	<Bus>NmGlobalConfig(NmConfig.<Bus>NmCluster): - <Bus>NmPnResetTime(local) is configured <Bus>NmChannelConfig(<Bus>NmCluster): BUS1 - <Bus>NmPnEnabled(PncMapping present) = TRUE - <Bus>NmPnEraRxNSduRef(local) is configured - <Bus>NmPnEraCalcEnabled(local) = TRUE <Bus>NmChannelConfig(<Bus>NmCluster): BUS2 - <Bus>NmPnEnabled(PncMapping present) = TRUE - <Bus>NmPnEraRxNSduRef(local) is configured - <Bus>NmPnEraCalcEnabled(local) = TRUE		

	BswM: - A mode declaration group containing the following modes : PNC_NO_COM and PNC_FULL_COM must be managed by the BswM - BswM rules : -- When PNC 0 is in FULL_COM, the mode declaration group must switch in PNC_FULL_COM -- When PNC 0 is in NO_COM, the mode declaration group must switch in PNC_NO_COM
Summary	The aim of this test is to verify that the SUT (configured in ERA mode) is able to keep the bus awake when it receives Nm frames periodically with the PNV which is relevant to the SUT.
Needed Adaptation to other Releases	None
Pre-conditions	The SUT is started. ComM State equals to COMM_NO_COMMUNICATION for both BUS1 and BUS2 channels

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT<BUS1>] <Bus>NmChannelConfig (BUS1) start	
Step 2	[LT<BUS2>] <Bus>NmChannelConfig (BUS2) start	
Step 3	[LT<BUS2>] Check the Nm frames on the bus	[LT<BUS2>] No Nm frame on the BUS2
Step 4	[LT<BUS1>] Send periodically on BUS1 the Nm frame with the Partial Network 0 bit to 1	
Step 5	[CP] Wait MAX(<NM MSG CYCLE TIME>, CH1/CH2) + <safety margin>	[SWC] Check BswM PNC Mode declaration group (Rte_Mode()). Mode declaration group must be in state PNC_FULL_COM.
Step 6	-	[LT<BUS2>] Nm frames must be sent by SUT with Partial Network 0 bit to 1
Post-conditions	None	

6.3.5 [ATS_COMM_00182] Managing Partial Network Vector in ERA mode when all received relevant PN bits are equal to 0

Test Objective	Managing Partial Network Vector in ERA mode when all received relevant PN bits are equal to 0		
ID	ATS_COMM_00182	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CanNm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00114		

Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00412 CANNetworkManagement: SWS_CanNm_00415 CANNetworkManagement: SWS_CanNm_00416 CANNetworkManagement: SWS_CanNm_00417 CANNetworkManagement: SWS_CanNm_00418 CANNetworkManagement: SWS_CanNm_00420 CANNetworkManagement: SWS_CanNm_00434 CANNetworkManagement: SWS_CanNm_00435 CANNetworkManagement: SWS_CanNm_00436 CANNetworkManagement: SWS_CanNm_00437 CANNetworkManagement: SWS_CanNm_00438 CANNetworkManagement: SWS_CanNm_00439 CANNetworkManagement: SWS_CanNm_00442 CANNetworkManagement: SWS_CanNm_00443
Requirements / Reference to Test Environment	Configuration use case : UC04.02
Configuration Parameters	CanNmGlobalConfig(NmConfig.CanNmCluster): - CanNmPnResetTime(local) is configured CanNmChannelConfig(CanNmCluster): Bus CAN1 - CanNmPnEnabled(PncMapping present) = TRUE - CanNmPnEraRxNSduRef(local) is configured - CanNmPnEraCalcEnabled(local) = TRUE - CanNmTimeoutTime(nmNetworkTimeout) is configured CanNmChannelConfig(CanNmCluster): Bus CAN2 - CanNmPnEnabled(PncMapping present) = TRUE - CanNmPnEraRxNSduRef(local) is configured - CanNmPnEraCalcEnabled(local) = TRUE - CanNmTimeoutTime(nmNetworkTimeout) is configured BswM: - A mode declaration group containing the following modes : PNC_GATEWAY_ACTIVATED and PNC_GATEWAY_DEACTIVATED must be managed by the BswM - BswM rules : -- When PNC 0 is in FULL_COM, the mode declaration group must switch in PNC_GATEWAY_ACTIVATED -- When PNC 0 is in NO_COM, the mode declaration group must switch in PNC_GATEWAY_DEACTIVATED
Summary	The aim of this test is to verify that the SUT (configured in ERA mode) is able to stop the bus when it receives Nm frames periodically with the PNV which is not relevant to the SUT.
Needed Adaptation to other Releases	None
Pre-conditions	The SUT is started. ComM State equals to COMM_NO_COMMUNICATION for both BUS1 and BUS2 channels

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] Bus CAN1 start and send Nm request to PNC0.	[LT] SUT start communication on CAN2 BUS

Step 2	-	[SWC] Check BswM PNC Mode declaration group (Rte_Mode()). Mode declaration group must be in state PNC_GATEWAY_ACTIVATED.
Step 3	[LT] Send periodically a request to PNC3 only (not relevant PNC for SUT) by setting PNV to PNC3 value.	
Step 4	[CP] Wait MAX(<nmNetworkTimeout>, CH1/CH2) + <safety margin>	[SWC] Check BswM PNC Mode declaration group (Rte_Mode()). Mode declaration group must be in state PNC_GATEWAY_DEACTIVATED.
Step 5	-	[LT] SUT must not send frames on CAN Buses
Post-conditions	None	

6.3.6 [ATS_COMM_00264] Managing Partial Network gatewaying when multiple Partial Networks are configured

Test Objective	Managing Partial Network gatewaying when multiple Partial Networks are configured		
ID	ATS_COMM_00264	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CanNm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00114		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00412 CANNetworkManagement: SWS_CanNm_00415 CANNetworkManagement: SWS_CanNm_00416 CANNetworkManagement: SWS_CanNm_00417 CANNetworkManagement: SWS_CanNm_00418 CANNetworkManagement: SWS_CanNm_00420 CANNetworkManagement: SWS_CanNm_00434 CANNetworkManagement: SWS_CanNm_00435 CANNetworkManagement: SWS_CanNm_00436 CANNetworkManagement: SWS_CanNm_00437 CANNetworkManagement: SWS_CanNm_00438 CANNetworkManagement: SWS_CanNm_00439 CANNetworkManagement: SWS_CanNm_00442 CANNetworkManagement: SWS_CanNm_00443		
Requirements / Reference to Test Environment	Configuration use case : UC04.02		
Configuration Parameters	CanNmGlobalConfig(NmConfig.CanNmCluster): - CanNmPnResetTime(local) is configured		

	CanNmChannelConfig(CanNmCluster): Bus CAN1 - CanNmPnEnabled(PncMapping present) = TRUE - CanNmPnEraRxNSduRef(local) is configured - CanNmPnEraCalcEnabled(local) = TRUE CanNmChannelConfig(CanNmCluster): Bus CAN2 - CanNmPnEnabled(PncMapping present) = TRUE - CanNmPnEraRxNSduRef(local) is configured - CanNmPnEraCalcEnabled(local) = TRUE
Summary	The aim of this test is to verify that the SUT (configured in ERA mode) is able to gateway the Partial networks when there are multiple Partial Networks configured. Relevant Partial Networks for the SUT are the following : - Partial Network 0 : PNV = 0x01 - Partial Network 1 : PNV = 0x02
Needed Adaptation to other Releases	None
Pre-conditions	The SUT is started.
Main Test Execution	
Test Steps	**Pass Criteria**
Step 1	[LT] CanNmChannelConfig (Bus CAN1) start
Step 2	[LT] CanNmChannelConfig (Bus CAN2) start
Step 3	[LT] Send periodically on CAN1 bus with PNV = 0x01	[LT] The SUT must send Nm frames on CAN2 bus with PNV = 0x01
Step 4	[LT] Send the I-Pdu PNC0_ECU3_To_ECU2_Ipdu on CAN2 bus	[LT] The SUT must gateway the I-Pdu on CAN1 bus
Step 5	[LT] Send the I-Pdu PNC0_ECU2_To_ECU3_Ipdu on CAN1 bus	[LT] The SUT must gateway the I-Pdu on the CAN2 bus
Step 6	[LT] Send periodically on CAN2 bus with PNV = 0x02	[LT] The SUT must send Nm frames on CAN1 bus with PNV = 0x02
Step 7	[LT] Send the I-Pdu PNC0_ECU4_To_ECU2_Ipdu on CAN2 bus	[LT] The SUT must gateway the I-Pdu on the CAN1 bus
Step 8	[LT] Send periodically on CAN1 bus with PNV = 0x03	[LT] The SUT must send Nm frames on CAN2 bus with PNV = 0x03
Step 9	[LT] Send the I-Pdu PNC0_ECU2_To_ECU4_Ipdu on CAN1 bus	[LT] The SUT must gateway the I-Pdu on the CAN2 bus
Step 10	[LT] Send periodically on CAN1 bus with PNV = 0x02	[LT] The SUT must send Nm frames on CAN2 bus with PNV = 0x02
Step 11	[LT] Send the I-Pdu PNC0_ECU3_To_ECU2_Ipdu on CAN2 bus	[LT] The SUT must NOT gateway the I-Pdu on the CAN1 bus
Post-conditions	None

6.3.7 [ATS_COMM_00183] Keeping the bus awake whatever the value of the PN vector when all the Nm frames keep awake the SUT

Test Objective	Keeping the bus awake whatever the value of the PN vector when all the Nm frames keep awake the SUT		
ID	ATS_COMM_00183	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CanNm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00114		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00410 CANNetworkManagement: SWS_CanNm_00421		
Requirements / Reference to Test Environment	Configuration use case : UC04.01		
Configuration Parameters	CanNmGlobalConfig(NmConfig.CanNmCluster): - CanNmPnEiraCalcEnabled(local) = TRUE - CanNmPnResetTime(local) is configured - CanNmPnEiraRxNSduRef(local) is configured CanNmChannelConfig(CanNmCluster): Bus CAN2 - CanNmPnEnabled(PncMapping present) = TRUE - CanNmAllNmMessagesKeepAwake(local) = TRUE		
Summary	The aim of this test is to verify that the SUT (configured with CanNmAllNmMessagesKeepAwake = true) is able to keep the bus awake when it only receives Nm frames with the PNI bit set to 0.		
Needed Adaptation to other Releases	None		
Pre-conditions	The SUT is started.		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[LT] CanNmChannelConfig (Bus CAN2) start		
Step 2	[LT] Send on CAN2 Nm Frame with PNI bit = 0		[LT] The SUT keep the bus awake
Step 3	-		[SWC] ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_FULL_COMMUNICATION
Post-conditions	None		

6.3.8 [ATS_COMM_00185] Sending Nm frames periodically, with PN vector equals to 0 when Partial Network is deactivated

Test Objective	Sending Nm frames periodically, with PN vector equals to 0 when Partial Network is deactivated		
ID	ATS_COMM_00185	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CanNm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00114		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00414		
Requirements / Reference to Test Environment	Configuration use case : UC04.01		
Configuration Parameters	CanNmGlobalConfig(NmConfig.CanNmCluster): - CanNmPnEiraCalcEnabled(local) = TRUE - CanNmPnEiraRxNSduRef(local) is configured CanNmChannelConfig(CanNmCluster): Bus CAN3 - CanNmPnEnabled(PncMapping present) = FALSE		
Summary	The aim of this test is to verify that the SUT (configured with CanNmPnEnabled = false) is able to send Nm frames periodically, with the PNI bit set to 0, when the network is requested by the application.		
Needed Adaptation to other Releases	None		
Pre-conditions	The SUT is started.		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[LT] CanNmChannelConfig (Bus CAN3) start		
Step 2	[LT] Send on CAN3 Nm Frame periodically with PNI bit = 0		
Step 3	[SWC] Request access to the BUS by calling ComM_RequestComMode(<user_pnc1>, COMM_FULL_COMMUNICATION)		[LT] The SUT send Nm frame periodically with PNI bit should be equal to 0
Post-conditions	None		

6.3.9 [ATS_COMM_00186] Keeping the bus awake whatever the value of the PN vector when partial network is deactivated

Test Objective	Keeping the bus awake whatever the value of the PN vector when partial network is deactivated		
ID	ATS_COMM_00186	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2

Affected Modules	CanNm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00114		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00409		
Requirements / Reference to Test Environment	Configuration use case : UC04.01		
Configuration Parameters	CanNmGlobalConfig(NmConfig.CanNmCluster): - CanNmPnEiraCalcEnabled(local) = TRUE - CanNmPnEiraRxNSduRef(local) is configured CanNmChannelConfig(CanNmCluster): Bus CAN3 - CanNmPnEnabled(PncMapping present) = FALSE		
Summary	The aim of this test is to verify that the SUT (configured with CanNmPnEnabled = false) is able to keep the bus awake when it only receives Nm frames with the PNI bit set to 1.		
Needed Adaptation to other Releases	None		
Pre-conditions	The SUT is started.		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[LT] CanNmChannelConfig (Bus CAN3) start		
Step 2	[LT] Send on CAN3 Nm Frame with PNI bit = 1		[LT] The SUT keep the bus awake
Step 3	-		[SWC] ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_FULL_COMMUNICATION
Post-conditions	None		

6.3.10 [ATS_COMM_00262] Managing Partial Network Vector in EIRA mode when multiple PN are relevant to the SUT

Test Objective	Managing Partial Network Vector in EIRA mode when multiple PN are relevant to the SUT		
ID	ATS_COMM_00262	**AUTOSAR Releases**	3.2.1|3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CanNm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00114		

Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00411 CANNetworkManagement: SWS_CanNm_00412 CANNetworkManagement: SWS_CanNm_00415 CANNetworkManagement: SWS_CanNm_00416 CANNetworkManagement: SWS_CanNm_00417 CANNetworkManagement: SWS_CanNm_00418 CANNetworkManagement: SWS_CanNm_00419 CANNetworkManagement: SWS_CanNm_00420 CANNetworkManagement: SWS_CanNm_00423 CANNetworkManagement: SWS_CanNm_00424 CANNetworkManagement: SWS_CanNm_00425 CANNetworkManagement: SWS_CanNm_00426 CANNetworkManagement: SWS_CanNm_00428 CANNetworkManagement: SWS_CanNm_00429 CANNetworkManagement: SWS_CanNm_00431 CANNetworkManagement: SWS_CanNm_00432
Requirements / Reference to Test Environment	Configuration use case : UC04.01
Configuration Parameters	CanNmGlobalConfig(NmConfig.CanNmCluster): - CanNmPnEiraCalcEnabled(local) = TRUE - CanNmPnResetTime(local) is configured - CanNmPnEiraRxNSduRef(local) is configured - CanNmPnInfo : - CanNmPnInfoLength(System.pncVectorLength) = 1 - CanNmPnFilterMaskByte(all contributing CanCommunicationConnector.pncWakeupDataMask will be bitwise ORed to obtain the value of CanNmPnFilterMaskByte) = 0x16 CanNmChannelConfig(CanNmCluster): Bus CAN1 - CanNmPnEnabled(PncMapping present) = TRUE - CanNmTimeoutTime(nmNetworkTimeout) is configured
Summary	The aim of this test is to verify that the SUT (configured in EIRA mode) is able to keep the bus awake and to set it to sleep when multiple Partial Networks are relevant to the SUT. Relevant Partial Networks for the SUT : - Partial Network 1 : PNV = 0x02 - Partial Network 2 : PNV = 0x04 - Partial Network 4 : PNV = 0x10
Needed Adaptation to other Releases	None
Pre-conditions	The SUT is started.

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] CanNmChannelConfig (Bus CAN1) start	
Step 2	[LT] Send periodically the Nm frame with PNV = 0x04	[LT] The SUT to keep the bus awake
Step 3	[SWC] Request the Partial Network 4 by calling ComM_RequestComMode(<user_pnc4>, COMM_FULL_COMMUNICATION)	[LT] The SUT must send its Nm frame with PNV = 0x10

Step 4	[SWC] Request the Partial Network 1 by calling ComM_RequestComMode(<user_pnc1>, COMM_FULL_COMMUNICATION)	[LT] The SUT must send its Nm frame with PNV = 0x12
Step 5	[SWC] Release the Partial Network 4 by calling ComM_RequestComMode(<user_pnc4>, COMM_NO_COMMUNICATION)	[LT] The SUT must send its Nm frame with PNV = 0x02
Step 6	[SWC] Request the Partial Network 2 by calling ComM_RequestComMode(<user_pnc2>, COMM_FULL_COMMUNICATION)	[LT] The SUT must send its Nm frame with PNV = 0x06
Step 7	[SWC] Release the Partial Network 1 by calling ComM_RequestComMode(<user_pnc1>, COMM_NO_COMMUNICATION)	[LT] The SUT must send its Nm frame with PNV = 0x04
Step 8	[SWC] Release the Partial Network 2 by calling ComM_RequestComMode(<user_pnc2>, COMM_NO_COMMUNICATION)	[SWC] ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_FULL_COMMUNICATION
Step 9	-	[LT] The SUT must stop sending its Nm frame The SUT must keep the bus awake
Step 10	[LT] Send periodically the Nm frame with PNV = 0x19	[LT] The SUT must keep the bus awake
Step 11	[SWC] Get current Communication mode	[SWC] ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_FULL_COMMUNICATION
Step 12	[LT] Send periodically the Nm frame with PNV = 0x11	[LT] The SUT must keep the bus awake
Step 13	[SWC] Get current Communication mode	[SWC] ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_FULL_COMMUNICATION
Step 14	[LT] Send periodically the Nm frame with PNV = 0x01	[LT] The SUT must keep the bus awake during <CanNmPnResetTime> seconds.
Step 15	[CP] Wait <nmNetworkTimeout> + <safety margin>	[LT] No Nm Messages transmitted by SUT. (After this timeout, the SUT must set the bus in sleep)
Step 16	[SWC] Get current Communication mode	[SWC] ComM service GetCurrentComMode shall return E_OK Current ComMode shall be COMM_NO_COMMUNICATION
Post-conditions	None	

6.3.11 [ATS_COMM_00806] Keep the PNC active as long as at least one ComM User is assigned to this PNC requests “Full Communication”

Test Objective	Keep the PNC active as long as at least one ComM User is assigned to this PNC requests “Full Communication”		
ID	ATS_COMM_00806	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00936		
Requirements / Reference to Test Environment	none		
Configuration Parameters	COMM_USER = 1 COMM_NM_VARIANT = FULL COMM_PNC_ENABLED = TRUE ComMUserPerChannel = Two Instance		
Summary	Configure SWC0 with one PNC which is mapped to ComM Channel 0 and one RX and TX Ipdu-Group is mapped to this particular PNC. Configure SWC1 with reference to same PNC. In NM Frame 2nd bytes first bit is mapped to this particular PNC. User 0 and User 1 are mapped to PNC 1.		
Needed Adaptation to other Releases			
Pre-conditions	DUT shall be initialized		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[SWC] Send the request for changing COM mode to Full communication for user1		[SWC] RTE shall return E_OK
Step 2	[SWC] Request for the current communication mode of ComM user1		[SWC] ComM mode shall be in Full communication
Step 3	[SWC] Send the request for changing COM mode to No communication for user1		[SWC] RTE shall return E_OK
Step 4	[SWC] Send the request for changing COM mode to Full communication for user0		[SWC] RTE shall return E_OK BswM user callout shall be invoked for transition to PNC_REQUESTED State
Step 5	-		[LT] DUT sends NM Frames with the respective PNC bit as set

Step 6	[SWC] Send the request for the current mode of ComM	[SWC] ComM mode shall be in Full communication
Step 7	[SWC] Check the PNC state	[SWC] Partial Network shall be in PNC_REQUESTED State
Post-conditions	NONE	

6.3.12 [ATS_COMM_00807] All ComMUsers assigned to this PNC request “No Communication” and the PNC bit in all ERA is equal to ‘0’

Test Objective	All ComMUsers assigned to this PNC request “No Communication” and the PNC bit in all ERA is equal to ‘0’		
ID	ATS_COMM_00807	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00991		
Requirements / Reference to Test Environment	none		
Configuration Parameters	COMM_NM_VARIANT = FULL COMM_PNC_ENABLED = TRUE COMM_PNC_GW_E-BLED = TRUE ComMPncComSignalKind = ERA		
Summary	Configure SWC0 and SWC1 with one PNC which is mapped to ComM Channel 0 and one RX and TX Ipdu-Group is mapped to this particular PNC In NM Frame 2 bytes first bit is mapped to this particular PNC.		
Needed Adaptation to other Releases			
Pre-conditions	DUT shall be initialized		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[SWC] Send the request for changing COM mode to Full communication for user1		[SWC] RTE shall return E_OK
Step 2	[SWC] Send the request for the current mode of ComM		[SWC] ComM shall be in Full communication
Step 3	[SWC] Send the request for changing COM mode to No communication for user1		[SWC] RTE shall return E_OK BswM user callout shall be invoked for the transition to PNC Ready sleep State

Step 4	[SWC] Send the request for the current mode of ComM	[SWC] ComM shall be in No communication mode
Step 5	[SWC] Check the PNC state	[SWC] PNC shall be in PNC READY SLEEP State
Step 6	-	[LT] Nm Frames transmitted by the DUT shall be observed on the bus
Post-conditions	NONE	

6.3.13 [ATS_COMM_00808] Test DUT For Wakeup On The Bus In A PNC When The Parameter 'ComMSynchronousWakeUp' Is Set To FALSE Test Case Description

Test Objective	Test DUT For Wakeup On The Bus In A PNC When The Parameter 'ComMSynchronousWakeUp' Is Set To FALSE Test Case Description		
ID	ATS_COMM_00808	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00960		
Requirements / Reference to Test Environment	none		
Configuration Parameters	COMM_NM_VARIANT = FULL COMM_PNC_ENABLED = TRUE		
Summary	Configure SWC with one PNC which is mapped to ComM Channel 0 and one RX and TX Ipdu-Group is mapped to this particular PNC In NM Frame 2 bytes first bit is mapped to this particular PNC.		
Needed Adaptation to other Releases			
Pre-conditions	DUT shall be initialized		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[LT] Send a valid frame from Tester to DUT (The DUT shall wake up passively)		-
Step 2	[LT] Tester to Send Nm message with 2nd byte's first bit is set periodically to DUT		[SWC] BswM user callout shall be invoked for transition to PNC NO COMMUNICATION State

Step 3	[SWC] Send the request for the current mode of ComM	[SWC] ComM mode shall be in Full communication
Step 4	[SWC] Check the PNC state	[SWC] PNC shall be in PNC NO COMMUNICATION State
Post-conditions	NONE	

6.3.14 [ATS_COMM_00809] Behaviour Of The PNC State Machine When It Is In PNC Prepare Sleep And ComMUser Requests For Full Communication Through Application Request

Test Objective	Behaviour Of The PNC State Machine When It Is In PNC Prepare Sleep And ComMUser Requests For Full Communication Through Application Request		
ID	ATS_COMM_00809	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00948		
Requirements / Reference to Test Environment	none		
Configuration Parameters	COMM_NM_VARIANT = FULL COMM_PNC_ENABLED = TRUE		
Summary	Configure SWC0 with one PNC which is mapped to ComM Channel 0 and one RX and TX Ipdu-Group is mapped to this particular PNC In NM Frame 2 bytes first bit is mapped to this particular PNC.		
Needed Adaptation to other Releases			
Pre-conditions	DUT shall be initialized		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[SWC] Send the request for changing COM mode to Full communication for User0.		[SWC] RTE shall return E_OK
Step 2	[SWC] Send the request for the current mode of ComM		[SWC] ComM mode shall be in Full communication
Step 3	[SWC] Send the request for changing COM mode to No communication for User0.		[SWC] RTE shall return E_OK BswM user callout shall be invoked for transition to PNC_PREPARE_SLEEP State

Step 4	[SWC] Check the PNC state	[SWC] PNC shall be in PNC_READY_SLEEP State
Step 5	[SWC] Send the request for changing COM mode to Full communication for User0.	[SWC] RTE shall return E_OK BswM user callout shall be invoked for transition to PNC_REQUESTED State
Step 6	-	[LT] NM Frames transmitted by the DUT shall be observed on bus
Post-conditions	NONE	

6.3.15 [ATS_COMM_00810] Behaviour Of The PNC State Machine When It Is In PNC Prepare Sleep And ComM User Requests For Full Communication Through Passive Wakeup

Test Objective	Behaviour Of The PNC State Machine When It Is In PNC Prepare Sleep And ComM User Requests For Full Communication Through Passive Wakeup		
ID	ATS_COMM_00810	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00950		
Requirements / Reference to Test Environment	none		
Configuration Parameters	COMM_NM_VARIANT = FULL COMM_PNC_ENABLED = TRUE		
Summary	Configure SWC0 with one PNC which is mapped to ComM Channel 0 and one RX and TX Ipdu-Group is mapped to this particular PNC In NM Frame 2 bytes first bit is mapped to this particular PNC When this ECU receives NM frame with Particular bit as set it is expected the ComM will intimate BswM to start the particular IPD-Group.		
Needed Adaptation to other Releases			
Pre-conditions	DUT shall be initialized		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[LT] Send a valid frame from Tester to DUT (The DUT shall wake up passively)		-

Step 2	[LT] Tester to Send Nm message with 2nd byte's first bit is set periodically to DUT	[SWC] BswM user callout shall be invoked for transition to PNC PREPARE SLEEP State
Step 3	[SWC] Check the PNC state	[SWC] PNC shall be in PNC PREPARE SLEEP State
Step 4	[SWC] Send the request for changing COM mode to Full communication for user3	[SWC] RTE shall return E_OK BswM user callout shall be invoked for transition to PNC_REQUESTED State
Step 5	[SWC] Check the PNC state	[SWC] PNC shall be in PNC_REQUESTED State
Step 6	-	[LT] DUT shall send NM frames with the respective values set
Post-conditions	NONE	

7 RS_BRF_01680 – Network Management on CAN

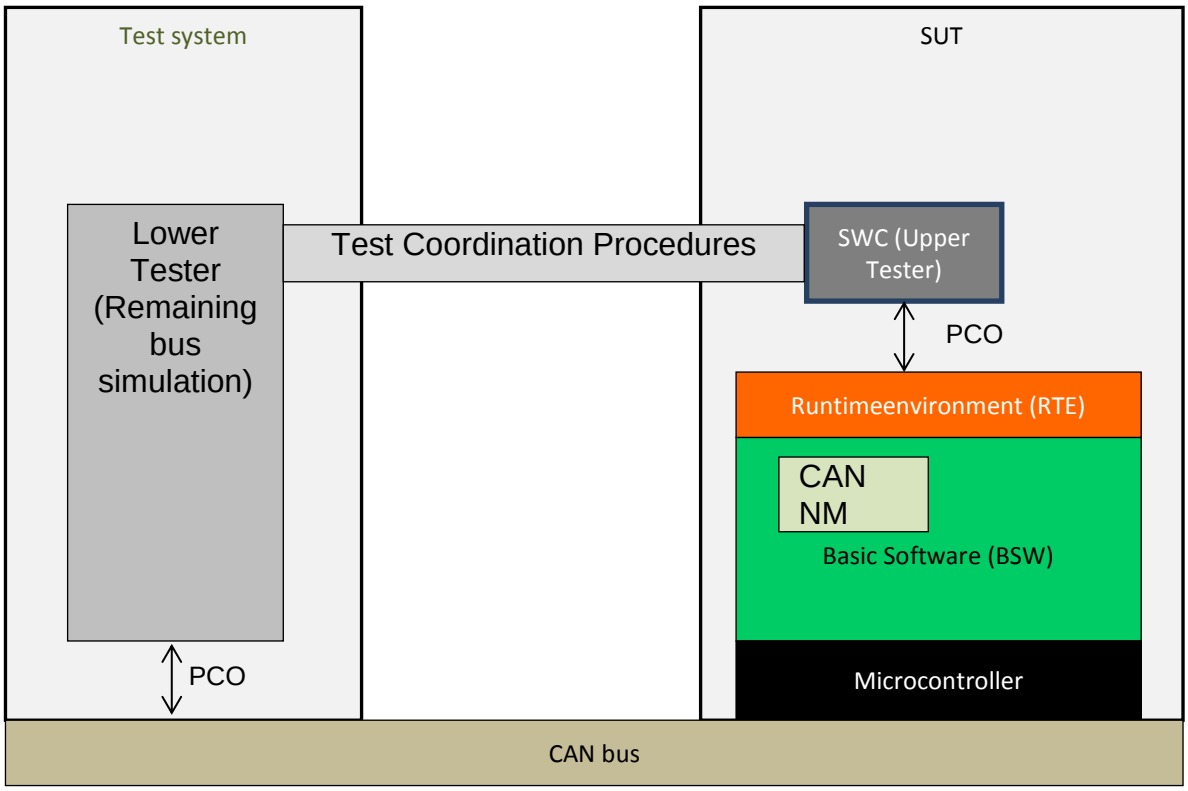
7.1 General Test Objective and Approach

The “CAN NM” features are tested by setting the conditions which should trigger transitions between Network management modes and states, and then checking whether the transitions are performed correctly, following the right timing constraints.

7.1.1 Test System

7.1.1.1 Overview on Architecture

The basic test setup is depicted in the following figure:



The test cases require a SWC as Upper Tester. This SWC indirectly stimulates the CAN NM functionality under test, and the behavior is observed through the CAN Bus.

7.1.1.2 Specific Requirements

Some test cases additionally require a “Remaining bus simulation” as Lower Tester. It is used for simulating the NM messages sent by other nodes (ECUs) of the cluster.

To meet certain test case behavior conditions (such as Bus Load Reduction), the Low Tester shall send NM frames on the bus using only one frame ID.

7.1.1.3 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.

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 Configuration Sets

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 3 proposals of configuration sets (PS001, PS002 and PS003). Each test case description includes a field mentioning the configuration sets which are applicable to that test case among these proposals.

System Template Parameter	EcuC parameter	PS001	PS002	PS003
NetworkManagement::CanNmCluster .nmWaitBusSleepTime	CANNM_WAIT_BUS_SLEEP_TIME	OEM >0	OEM >0	OEM >0
NetworkManagement::CanNmCluster .nmNetworkTimeout	CANNM_TIMEOUT_TIME	OEM >0	OEM >0	OEM >0
NetworkManagement::NmClusterCoupling.nmBusloadReductionEnabled	CANNM_BUS_LOAD_REDUCTION_ENABLED	FALSE	TRUE	TRUE
NetworkManagement::CanNmCluster .nmRepeatMessageTime	CANNM_REPEAT_MESSAGE_TIME	OEM >0	OEM >0	OEM >0
NetworkManagement::NmEcu.nmComControlEnabled	CANNM_COM_CONTROL_ENABLED	TRUE	TRUE	TRUE
NetworkManagement::CanNmCluster .nmNidPosition	CANNM_PDU_NID_POSITION	OEM >0	OEM >0	OEM >0
NetworkManagement::CanNmNode. nmNodeId	CANNM_NODE_ID	OEM >0	OEM >0	OEM >0
NetworkManagement::CanNmNode. nmMsgReducedTime	CANNM_MSG_REDUCED_TIME		OEM >0	OEM >0
NetworkManagement::CanNmCluster .nmMsgCycleTime	CANNM_MSG_CYCLE_TIME	OEM >0	OEM >0	OEM >0
NetworkManagement::CanNmNode. nmMsgCycleOffset	CANNM_MSG_CYCLE_OFFSET	OEM >0	OEM >0	OEM >0
NetworkManagement::CanNmCluster .nmImmediateNmTransmissions	CANNM_IMMEDIATE_NM_TRANSMISSIONS	0	0	OEM >0
NetworkManagement::NmEcu.nmNodeDetectionEnabled	CANNM_NODE_DETECTION_ENABLED	TRUE	TRUE	TRUE
NetworkManagement::CanNmCluster .nmImmediateNmCycleTime	CANNM_IMMEDIATE_NM_CYCLETIME			OEM >0
NetworkManagement::CanNmCluster .nmBusloadReductionActive	CANNM_BUS_LOAD_REDUCTION_ACTIVE	FALSE	TRUE	TRUE
PncMapping::PncMapping	CANNM_PN_ENABLED	=∅	=∅	=∅

7.1.2.2 Mandatory vs. Customizable Parts

Timing and counter values may be changed to the user's requirements or typical values.

7.1.3 Test Case Design

The test cases have been designed following the state machines specified by SW Autosar specification, as shown on the table below.

Transition to be tested		Number of test cases
NM modes transitions		
Network Mode -> Prepare Bus sleep Mode		
	timeout fulfilled	1
	timeout not fulfilled	1
Prepare Bus sleep Mode -> Bus sleep mode		
	On timer expiration	1
Bus sleep mode -> Network Mode		
	On passive start-up	1
	On network request	1
Prepare Bus sleep Mode -> Network Mode		
	On NM Rx Indication without bus load reduction	1
	On NM Rx Indication with bus load reduction	1
	On Network Request	1
NM states transitions		
Repeat Message State-> Normal operation State		
	with busload reduction inactive	1
	with busload reduction active	1
	with immediate nm transmissions	1
Repeat Message State -> Ready Sleep State		
	with no immediate nm transmissions	1
Normal operation State -> Repeat Message State		
	with repeat message bit received	1
Ready Sleep State -> Repeat Message State		
	with repeat message bit received	1

7.2 Re-usable Test Steps

No common test steps subsequence used in this test specification has been assessed as long enough for being considered as a test steps group.

7.3 Test Cases

7.3.1 [ATS_COMM_00036] Transition Network Mode to Prepare Bus-Sleep Mode - timeout fulfilled

Test Objective	Transition Network Mode to Prepare Bus-Sleep Mode - timeout fulfilled		
ID	ATS_COMM_00036	**AUTOSAR Releases**	4.0.3|4.1.1|4.2.1|4.2.2

Affected Modules	CANNm, Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00122		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00092 CANNetworkManagement: SWS_CanNm_00094 CANNetworkManagement: SWS_CanNm_00315 CANNetworkManagement: SWS_CanNm_00096 CANNetworkManagement: SWS_CanNm_00098 CANNetworkManagement: SWS_CanNm_00206 CANNetworkManagement: SWS_CanNm_00100 CANNetworkManagement: SWS_CanNm_00102 CANNetworkManagement: SWS_CanNm_00106 CANNetworkManagement: SWS_CanNm_00108 CANNetworkManagement: SWS_CanNm_00109 CANNetworkManagement: SWS_CanNm_00115 CANNetworkManagement: SWS_CanNm_00124 CANNetworkManagement: SWS_CanNm_00246 CANNetworkManagement: SWS_CanNm_00247 CANNetworkManagement: SWS_CanNm_00248 CANNetworkManagement: SWS_CanNm_00237 CANNetworkManagement: SWS_CanNm_00032 CANNetworkManagement: SWS_CanNm_00040 CANNetworkManagement: SWS_CanNm_00051		
Requirements / Reference to Test Environment	Test SWC shall be able to call ComM_RequestComMode to release or request the communication. Test environment shall be capable to measure CAN signal timings….Test logging and reporting, timing measurements, access capabilities, injection capabilities…		
Configuration Parameters	Parameter set: PS001, PS002		
Summary	It shall be tested that the SUT switches from “Network Mode” (sub-state "Ready Sleep State") to "Prepare Bus-Sleep Mode" when there aren’t any NM messages received from the CAN bus [SWS_CanNm_00109]. The transition to "Prepare Bus-Sleep Mode" can only be tested indirectly. When the transmission of NM messages on the bus is stopped by the test environment and the <CANNM_TIMEOUT_TIME> [SWS_CanNm_00246] expires the transition from “Network Mode” (sub-state “Ready Sleep State”) to "Prepare Bus-Sleep Mode" shall happen [SWS_CanNm_00109] (as the network is still released). In this mode the SUT is still not sending any NM messages [SWS_CanNm_00108]. If the test environment starts sending NM messages again right after the timeout and within <CANNM_WAIT_BUS_SLEEP_TIME> [SWS_CanNm_00115][SWS_CanNm_00248] the SUT switches back to “Network Mode” (sub-state "Repeat Message State") [SWS_CanNm_00124]. That transition can be detected on the bus since the SUT is now sending NM messages [SWS_CanNm_00100] for the duration of <CANNM_REPEAT_MESSAGE_TIME> [SWS_CanNm_00102][SWS_CanNm_00247] with a cycle time of <CANNM_MSG_CYCLE_TIME> [SWS_CanNm_00040]. Afterwards the SUT will switch back to “Ready Sleep State” [SWS_CanNm_00106] and stops the transmission of NM messages again [SWS_CanNm_00108]. If all pass criteria are met, the transition to be tested was correctly made by the SUT.		
Needed Adaptation to other Releases	Needed adaptation for release [3.2.2] Configuration: none		

	Test steps: skip step action 3 from the test steps sequence.	
Pre-conditions	The SUT is in Network mode (in sub-state "Ready Sleep State"). The network is released by the test-application SWC.	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] The test environment stops sending NM messages.	[LT] No NM messages on the bus [SWS_CanNm_00108].
Step 2	[CP] Wait for <CANNM_TIMEOUT_TIME> [SWS_CanNm_00246]	[LT] There are still no NM messages on the bus
Step 3	[SWC] Check NM mode via Get State API	[SWC] SUT switches to "Prepare Bus-Sleep Mode" [SWS_CanNm_00109]
Step 4	[LT] The test environment immediately restarts sending NM messages (has to be within <CANNM_WAIT_BUS_SLEEP_TIME> [SWS_CanNm_00115][SWS_CanNm_00248]) (using only one frame ID) on the bus with a cycle time of <CANNM_MSG_CYCLE_TIME>.	[LT] [CANNM124]: The SUT sends NM messages [SWS_CanNm_00100] with a cycle time of <CANNM_MSG_CYCLE_TIME> [SWS_CanNm_00040].
Step 5	[SWC] Check the state and mode of NM via Getstate API	[SWC] NM mode shall be in "Network Mode" and state be in "Repeat Message State".
Step 6	[CP] Wait for <CANNM_REPEAT_MESSAGE_TIME> [SWS_CanNm_00247]	[SWC] SUT switches to “Ready Sleep State” [SWS_CanNm_00102][[SWS_CanNm_00106]
Step 7	-	[LT] There are just the NM messages from the test environment on the bus and no messages from SUT anymore [SWS_CanNm_00108].
Post-conditions	The SUT is in "Network Mode" (sub-state "Ready Sleep State"). The test environment sends NM messages on the bus.	

7.3.2 [ATS_COMM_00037] Transition Network Mode to Prepare Bus-Sleep Mode - timeout not fulfilled

Test Objective	Transition Network Mode to Prepare Bus-Sleep Mode - timeout not fulfilled		
ID	ATS_COMM_00037	**AUTOSAR Releases**	4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CANNm, Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00122		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00092 CANNetworkManagement: SWS_CanNm_00094		

	CANNetworkManagement: SWS_CanNm_00315 CANNetworkManagement: SWS_CanNm_00096 CANNetworkManagement: SWS_CanNm_00098 CANNetworkManagement: SWS_CanNm_00206 CANNetworkManagement: SWS_CanNm_00100 CANNetworkManagement: SWS_CanNm_00102 CANNetworkManagement: SWS_CanNm_00106 CANNetworkManagement: SWS_CanNm_00108 CANNetworkManagement: SWS_CanNm_00109 CANNetworkManagement: SWS_CanNm_00115 CANNetworkManagement: SWS_CanNm_00124 CANNetworkManagement: SWS_CanNm_00246 CANNetworkManagement: SWS_CanNm_00247 CANNetworkManagement: SWS_CanNm_00248 CANNetworkManagement: SWS_CanNm_00237 CANNetworkManagement: SWS_CanNm_00032 CANNetworkManagement: SWS_CanNm_00040 CANNetworkManagement: SWS_CanNm_00051
Requirements / Reference to Test Environment	Test SWC shall be able to call ComM_RequestComMode to release or request the communication Test environment shall be capable to measure CAN signal timings….Test logging and reporting, timing measurements, access capabilities, injection capabilities…
Configuration Parameters	Parameter set: PS001, PS002
Summary	It shall be tested that the transition “Network Mode” (sub-state "Ready Sleep State") to "Prepare Bus-Sleep Mode" is linked to the right timeout value <CANNM_TIMEOUT_TIME>, i.e. the SUT does >not< switch from “Network Mode” (sub-state "Ready Sleep State") to "Prepare Bus-Sleep Mode" when there are any NM messages received from the CAN bus just when the timeout event is imminent [SWS_CanNm_00109]. The transition to "Prepare Bus-Sleep Mode" can only be tested indirectly. When the test environment sends the next NM message on the bus a little bit less than <CANNM_TIMEOUT_TIME> [SWS_CanNm_00246] after the last NM message the transition from “Network Mode” (sub-state “Ready Sleep State”) to "Prepare Bus-Sleep Mode" [SWS_CanNm_00109] shall >not< happen and the SUT shall stay in “Ready Sleep State” not sending any NM messages [SWS_CanNm_00108]. To check the current state the test environment sends another NM message within <CANNM_WAIT_BUS_SLEEP_TIME> [SWS_CanNm_00115][SWS_CanNm_00248]. The SUT shall still be not sending any NM messages. If the transition to "Prepare Bus-Sleep Mode" would have been wrongly made the SUT would have restart sending NM messages [SWS_CanNm_00100] for the duration of <CANNM_REPEAT_MESSAGE_TIME> [SWS_CanNm_00102][SWS_CanNm_00247] with a cycle time of <CANNM_MSG_CYCLE_TIME> [SWS_CanNm_00040].
Needed Adaptation to other Releases	Needed adaptation for release [3.2.2] *Configuration:* none *Test steps:* add of an offset time in step 6 • Main 6 : Wait for at least <CANNM_MSG_CYCLE_OFFSET> + <CANNM_MSG_CYCLE_TIME> [SWS_CanNm_00005][SWS_CanNm_00040].
Pre-conditions	The SUT is in Network mode (in sub-state "Ready Sleep State"). The network is released by the test-application SWC.

	The test environment sends NM messages on the bus (with a cycle time of <CANNM_MSG_CYCLE_TIME>).	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] The test environment sends the next NM message (keeping current cycle time).	[LT] There are only the NM messages from the test environment on the bus.
Step 2	[CP] Wait for a little bit less than <CANNM_TIMEOUT_TIME> [SWS_CanNm_00246].	[LT] During and after this time: The SUT is still not sending NM messages.
Step 3	[LT] The test environment sends the next NM message [SWS_CanNm_00109] to keep current SUT state “Ready Sleep State”.	[LT] During this time, the SUT is still not sending NM messages.
Step 4	[SWC] Get status of NM via Get status API	[SWC] NM shall be in "Ready Sleep" state.
Step 5	[CP] Wait for MIN(<CANNM_MSG_CYCLE_TIME>, less than <CANNM_WAIT_BUS_SLEEP_TIME>) [SWS_CanNm_00115][SWS_CanNm_00248].	[LT] During and after this time: The SUT is still not sending NM messages.
Step 6	[LT] The test environment sends the next NM message.	[LT] During and after this time: The SUT is still not sending NM messages.
Step 7	[CP] Wait for at least <CANNM_MSG_CYCLE_TIME> [SWS_CanNm_00040].	[LT] During and after this time: The SUT is still not sending NM messages.
Step 8	[LT] The test environment sends the next NM message [SWS_CanNm_00109] to keep current SUT state “Ready Sleep State”.	[LT] The SUT is still not sending NM messages.
Step 9	[SWC] Get status of NM via Get status API	[SWC] NM shall be in "Ready Sleep" state
Post-conditions	The SUT is in "Network Mode" (sub-state "Ready Sleep State"). The test environment sends NM messages on the bus.	

7.3.3 [ATS_COMM_00038] Transition Prepare Bus-Sleep Mode to Bus-Sleep Mode

Test Objective	Transition Prepare Bus-Sleep Mode to Bus-Sleep Mode		
ID	ATS_COMM_00038	**AUTOSAR Releases**	3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CANNm, Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00122		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00092 CANNetworkManagement: SWS_CanNm_00115		

	CANNetworkManagement: SWS_CanNm_00246 CANNetworkManagement: SWS_CanNm_00248
Requirements / Reference to Test Environment	Test SWC shall be able to call ComM_RequestComMode to release or request the communication Test environment shall be capable to measure CAN signal timings….Test logging and reporting, timing measurements, access capabilities, injection capabilities…
Configuration Parameters	Parameter set: PS001, PS002
Summary	It shall be tested that the SUT switches from “Prepare Bus-Sleep Mode” to " Bus-Sleep Mode" when there aren’t any NM messages received from the CAN bus [SWS_CanNm_00115]. The transition to "Bus-Sleep Mode" can only be tested indirectly. When the transmission of NM messages on the bus is stopped by the test environment and the<CANNM_WAIT_BUS_SLEEP_TIME> [SWS_CanNm_00115] expires, the transition from “Prepare Bus-Sleep Mode” to " Bus-Sleep Mode " shall happen [SWS_CanNm_00115] (as the network is still released). In this mode the SUT is still not sending any NM messages [SWS_CanNm_00087]. If all pass criteria are met, the transition to be tested was correctly made by the SUT.
Needed Adaptation to other Releases	
Pre-conditions	The SUT is in Network mode (in sub-state "Ready Sleep State"). The network is released by the test-application SWC.

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] The test environment stops sending NM messages.	[LT] No NM messages on the bus.
Step 2	[CP] Wait for <CANNM_TIMEOUT_TIME> [SWS_CanNm_00246]	[LT] There are still no NM messages on the bus [SWS_CanNm_00108].
Step 3	[SWC] Get status of NM Via Get status API	[SWC] SUT switches to "Prepare Bus-Sleep Mode" [SWS_CanNm_00109]
Step 4	[CP] Wait for <CANNM_WAIT_BUS_SLEEP_TIME> [SWS_CanNm_00115][SWS_CanNm_00248]	[LT] There are still no NM messages on the bus [SWS_CanNm_00108].
Step 5	[SWC] Get status of NM via Get Status API	[SWC] SUT: Can Nm shall be in “Bus-Sleep Mode"
Post-conditions	The SUT is in "Bus-Sleep Mode". The test environment does not send NM messages on the bus.	

7.3.4 [ATS_COMM_00039] Transition Bus-Sleep Mode to Network Mode on passive start-up

Test Objective	Transition Bus-Sleep Mode to Network Mode on passive start-up

ID	ATS_COMM_00039	**AUTOSAR Releases**	3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CANNm, Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00122		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00087 CANNetworkManagement: SWS_CanNm_00088 CANNetworkManagement: SWS_CanNm_00092 CANNetworkManagement: SWS_CanNm_00094 CANNetworkManagement: SWS_CanNm_00314 CANNetworkManagement: SWS_CanNm_00100 CANNetworkManagement: SWS_CanNm_00102 CANNetworkManagement: SWS_CanNm_00106 CANNetworkManagement: SWS_CanNm_00108 CANNetworkManagement: SWS_CanNm_00128 CANNetworkManagement: SWS_CanNm_00105 CANNetworkManagement: SWS_CanNm_00246 CANNetworkManagement: SWS_CanNm_00247 CANNetworkManagement: SWS_CanNm_00248 CANNetworkManagement: SWS_CanNm_00237 CANNetworkManagement: SWS_CanNm_00032 CANNetworkManagement: SWS_CanNm_00040 CANNetworkManagement: SWS_CanNm_00051 CANNetworkManagement: SWS_CanNm_00211 CANNetworkManagement: SWS_CanNm_00214		
Requirements / Reference to Test Environment	Test SWC shall be able to set network state and to request a passive start-up. Test environment shall be capable to measure CAN signal timings….Test logging and reporting, timing measurements, access capabilities, injection capabilities…		
Configuration Parameters	Parameter set: PS002		
Summary	It shall be tested that the SUT switches from "Bus-Sleep Mode" to "Network Mode" (sub-state "Repeat Message State")[SWS_CanNm_00128][SWS_CanNm_00314] when function CanNm_PassiveStartUp() is called (when a Nm Frame is received on the bus) (when a Nm Frame is received on the bus). After this function call, the SUT starts transmitting cyclically NM messages [SWS_CanNm_00100] itself with a cycle time of <CANNM_MSG_CYCLE_TIME> [SWS_CanNm_00032][SWS_CanNm_00040][SWS_CanNm_00237]. Hint: If NM messages are received in Bus Sleep, CanNm calls "Nm_NetworkStartIndication" which leads to a "Nm_PassiveStartUp" by ComM. [SWS_Nm_00181]		
Needed Adaptation to other Releases			
Pre-conditions	No NM messages are on the bus. The network is requested by the test-application SWC. The SUT is in "Normal Operation State".		
Main Test Execution			

Test Steps		Pass Criteria
Step 1	[SWC] Release the Network by calling ComM_RequestComMode with COMM_NO_COMMUNICATION.	-
Step 2	[LT] Stop sending NM messages on the bus	[LT] No NM messages on the bus
Step 3	[CP] Wait for <CANNM_TIMEOUT_TIME> [SWS_CanNm_00246]	[LT] There are still no NM messages on the bus [SWS_CanNm_00108].
Step 4	[SWC] Check status of NM using Get Status API.	[SWC] SUT switches to "Prepare Bus-Sleep Mode" [SWS_CanNm_00109]
Step 5	[CP] Wait for <CANNM_WAIT_BUS_SLEEP_TIME> [SWS_CanNm_00115][SWS_CanNm_00248]	[LT] There are still no NM messages on the bus [SWS_CanNm_00108].
Step 6	[SWC] Check status of NM using Get Status API.	[SWC] SUT switches to “Bus-Sleep Mode"
Step 7	[LT] Start sending NM messages on the bus	[LT] Periodic NM messages are sent again on the bus by the SUT [SWS_CanNm_00128][SWS_CanNm_00087]
Post-conditions	The network is released by the test-application SWC. The SUT is in "Network Mode" (sub-state " Repeat Message State").	

7.3.5 [ATS_COMM_00040] Transition Bus-Sleep Mode to Network Mode on network request

Test Objective	Transition Bus-Sleep Mode to Network Mode on network request		
ID	ATS_COMM_00040	**AUTOSAR Releases**	3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CANNm, Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00122		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00087 CANNetworkManagement: SWS_CanNm_00088 CANNetworkManagement: SWS_CanNm_00092 CANNetworkManagement: SWS_CanNm_00094 CANNetworkManagement: SWS_CanNm_00314 CANNetworkManagement: SWS_CanNm_00100 CANNetworkManagement: SWS_CanNm_00102 CANNetworkManagement: SWS_CanNm_00106 CANNetworkManagement: SWS_CanNm_00108		

	CANNetworkManagement: SWS_CanNm_00128 CANNetworkManagement: SWS_CanNm_00104 CANNetworkManagement: SWS_CanNm_00105 CANNetworkManagement: SWS_CanNm_00246 CANNetworkManagement: SWS_CanNm_00247 CANNetworkManagement: SWS_CanNm_00248 CANNetworkManagement: SWS_CanNm_00237 CANNetworkManagement: SWS_CanNm_00032 CANNetworkManagement: SWS_CanNm_00040 CANNetworkManagement: SWS_CanNm_00051 CANNetworkManagement: SWS_CanNm_00211 CANNetworkManagement: SWS_CanNm_00214
Requirements / Reference to Test Environment	Test SWC shall be able to set network state Test environment shall be capable to measure CAN signal timings….Test logging and reporting, timing measurements, access capabilities, injection capabilities…
Configuration Parameters	Parameter set: PS002
Summary	It shall be tested that the SUT switches from "Bus-Sleep Mode" to "Network Mode" (sub-state "Repeat Message State")[SWS_CanNm_00128][SWS_CanNm_00314][SWS_CanNm_00104] when calling ComM_RequestComMode with COMM_FULL_COMMUNICATION. After this function call, the SUT starts transmitting cyclically NM messages [SWS_CanNm_00100] itself with a cycle time of <CANNM_MSG_CYCLE_TIME> [SWS_CanNm_00032][SWS_CanNm_00040][SWS_CanNm_00237].
Needed Adaptation to other Releases	
Pre-conditions	No NM messages are on the bus The network is requested by the test-application SWC The SUT is in "Normal Operation State"

Main Test Execution

Test Steps		**Pass Criteria**
Step 1	[SWC] Release the Network by calling ComM_RequestComMode with COMM_NO_COMMUNICATION	
Step 2	[LT] Stop sending NM messages on the bus	[LT] No NM messages on the bus
Step 3	[CP] Wait for <CANNM_TIMEOUT_TIME> [SWS_CanNm_00246]	-
Step 4	[SWC] Check status of NM using Get Status API.	[SWC] After this time, SUT switches to "Prepare Bus-Sleep Mode" [SWS_CanNm_00109]
Step 5	-	[LT] There are still no NM messages on the bus [SWS_CanNm_00108]

Step 6	[CP] Wait for <CANNM_WAIT_BUS_SLEEP_TIME> [SWS_CanNm_00115][SWS_CanNm_00248]	-
Step 7	[SWC] Check status of NM using Get Status API.	[SWC] After this time, SUT switches to “Bus-Sleep Mode"
Step 8	-	[LT] There are still no NM messages on the bus [SWS_CanNm_00108]
Step 9	[SWC] Request the Network by calling ComM_RequestComMode with COMM_FULL_COMMUNICATION	[LT] Periodic NM messages are sent again on the bus by the SUT [SWS_CanNm_00129][SWS_CanNm_00104][SWS_CanNm_00087]
Post-conditions	The network is released by the test-application SWC The SUT is in "Network Mode" (sub-state " Repeat Message State ")	

7.3.6 [ATS_COMM_00041] Transition Prepare Bus-Sleep Mode to Network Mode on NM Rx Indication

Test Objective	Transition Prepare Bus-Sleep Mode to Network Mode on NM Rx Indication		
ID	ATS_COMM_00041	**AUTOSAR Releases**	3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CANNm, Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00122		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00092 CANNetworkManagement: SWS_CanNm_00094 CANNetworkManagement: SWS_CanNm_00315 CANNetworkManagement: SWS_CanNm_00100 CANNetworkManagement: SWS_CanNm_00102 CANNetworkManagement: SWS_CanNm_00106 CANNetworkManagement: SWS_CanNm_00108 CANNetworkManagement: SWS_CanNm_00115 CANNetworkManagement: SWS_CanNm_00124 CANNetworkManagement: SWS_CanNm_00247 CANNetworkManagement: SWS_CanNm_00248 CANNetworkManagement: SWS_CanNm_00237 CANNetworkManagement: SWS_CanNm_00032 CANNetworkManagement: SWS_CanNm_00040 CANNetworkManagement: SWS_CanNm_00051		

Requirements / Reference to Test Environment	Test SWC shall be able to call ComM_RequestComMode to release or request the communication Test environment shall be capable to measure CAN signal timings….Test logging and reporting, timing measurements, access capabilities, injection capabilities…
Configuration Parameters	Parameter set: PS001, PS002
Summary	It shall be tested that the SUT switches from "Prepare Bus-Sleep Mode" to "Network Mode" (sub-state "Repeat Message State")[SWS_CanNm_00124][SWS_CanNm_00315] when receiving NM messages from the CAN bus. After reception of NM messages from the bus the SUT starts transmitting cyclically NM messages [SWS_CanNm_00100] itself with a cycle time of <CANNM_MSG_CYCLE_TIME> [SWS_CanNm_00032][SWS_CanNm_00040][SWS_CanNm_00237]. After transmitting NM messages by the SUT for <CANNM_REPEAT_MESSAGE_TIME> [SWS_CanNm_00102] the SUT will switch into the sub-state "Ready Sleep State" of the mode "Network Mode" [SWS_CanNm_00106] and no NM messages are transmitted anymore by the SUT [SWS_CanNm_00108].
Needed Adaptation to other Releases	
Pre-conditions	The SUT is in "Prepare Bus-Sleep Mode" for less than "CANNM_WAIT_BUS_SLEEP_TIME". The network is released by the test-application SWC. No NM messages are on the bus.

Main Test Execution

Test Steps		**Pass Criteria**
Step 1	[LT] The test environment sends NM messages (with only one frame ID) on the bus with a cycle time of <CANNM_MSG_CYCLE_TIME>.	[LT] The SUT sends NM messages [SWS_CanNm_00100] on the bus with a cycle time of <CANNM_MSG_CYCLE_TIME>[SWS_CanNm_00032][SWS_CanNm_00040][SWS_CanNm_0037] with a delay of <CANNM_MSG_REDUCED_TIME> regarding Lower Tester NM frames.
Step 2	[CP] Wait for <CANNM_REPEAT_MESSAGE_TIME> [SWS_CanNm_00102] to trigger transition "Repeat Message State" to “Ready Sleep State” [SWS_CanNm_00106].	[LT] After this time, there are just the NM messages from the test environment on the bus and no messages from SUT anymore [SWS_CanNm_00108].
Post-conditions	The SUT is in "Network Mode" (sub-state "Ready Sleep State"). The network is released by the test-application SWC.	

7.3.7 [ATS_COMM_00042] Transition Prepare Bus-Sleep Mode to Network Mode on Network Request

Test Objective	Transition Prepare Bus-Sleep Mode to Network Mode on Network Request		
ID	ATS_COMM_00042	**AUTOSAR Releases**	3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CANNm, Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00122		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00087 CANNetworkManagement: SWS_CanNm_00092 CANNetworkManagement: SWS_CanNm_00094 CANNetworkManagement: SWS_CanNm_00315 CANNetworkManagement: SWS_CanNm_00100 CANNetworkManagement: SWS_CanNm_00102 CANNetworkManagement: SWS_CanNm_00103 CANNetworkManagement: SWS_CanNm_00115 CANNetworkManagement: SWS_CanNm_00123 CANNetworkManagement: SWS_CanNm_00247 CANNetworkManagement: SWS_CanNm_00248 CANNetworkManagement: SWS_CanNm_00237 CANNetworkManagement: SWS_CanNm_00032 CANNetworkManagement: SWS_CanNm_00040		
Requirements / Reference to Test Environment	Test SWC shall be able to read network state. e.g.: by creating a BSWM mode rule and action list Test environment shall be capable to measure CAN signal timings….Test logging and reporting, timing measurements, access capabilities, injection capabilities…		
Configuration Parameters	Parameter set: PS001, PS002		
Summary	It shall be tested that the SUT switches from "Prepare Bus-Sleep Mode" to "Network Mode" (sub-state "Repeat Message State")[SWS_CanNm_00123][SWS_CanNm_00315] when the test-application requests network communication. After requesting network communication the SUT starts transmitting cyclically NM messages [SWS_CanNm_00100] with a cycle time of <CANNM_MSG_CYCLE_TIME> [SWS_CanNm_00032][SWS_CanNm_00040][SWS_CanNm_00237]. After transmitting NM messages by the SUT for <CANNM_REPEAT_MESSAGE_TIME> [SWS_CanNm_00102] the SUT will switch into the sub-state "Normal Operation State" [SWS_CanNm_00103]. NM messages are still transmitted by the SUT.		
Needed Adaptation to other Releases			
Pre-conditions	The SUT is in "Prepare Bus-Sleep Mode" for less than "CANNM_WAIT_BUS_SLEEP_TIME".		

	The network is released by the test-application SWC. No NM messages are on the bus.
Main Test Execution	
Test Steps	**Pass Criteria**
Step 1	[CP] Wait for <CANNM_MSG_CYCLE_OFFSET> [SWS_CanNm_00005].	[LT] During this time, there is still no NM message on the bus. After this time, the SUT starts sending NM messages [SWS_CanNm_00100] on the bus with a cycle time of <CANNM_MSG_CYCLE_TIME>> [SWS_CanNm_00032][SWS_CanNm_00040].
Step 2	[CP] Wait for <CANNM_REPEAT_MESSAGE_TIME> [SWS_CanNm_00102]	[LT] During this time, the SUT keeps sending NM messages [SWS_CanNm_00100] on the bus with a cycle time of <CANNM_MSG_CYCLE_TIME>> [SWS_CanNm_00032][SWS_CanNm_00040]. After this time, the SUT enters the “Normal Operation State" [SWS_CanNm_00103] and keeps sending NM messages a cycle time of <CANNM_MSG_CYCLE_TIME>> [SWS_CanNm_00032][SWS_CanNm_00040].
Post-conditions	The SUT is in "Network Mode" (sub-state "Normal Operation State"). The network is requested by the test-application SWC.

7.3.8 [ATS_COMM_00044] Transition Repeat Message State to Normal Operation State + busload reduction inactive

Test Objective	Transition Repeat Message State to Normal Operation State + busload reduction inactive		
ID	ATS_COMM_00044	**AUTOSAR Releases**	3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CANNm, Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00122		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00087 CANNetworkManagement: SWS_CanNm_00092 CANNetworkManagement: SWS_CanNm_00094 CANNetworkManagement: SWS_CanNm_00314 CANNetworkManagement: SWS_CanNm_00100 CANNetworkManagement: SWS_CanNm_00102 CANNetworkManagement: SWS_CanNm_00103 CANNetworkManagement: SWS_CanNm_00107 CANNetworkManagement: SWS_CanNm_00129 CANNetworkManagement: SWS_CanNm_00247		

	CANNetworkManagement: SWS_CanNm_00237 CANNetworkManagement: SWS_CanNm_00005 CANNetworkManagement: SWS_CanNm_00032 CANNetworkManagement: SWS_CanNm_00040 CANNetworkManagement: SWS_CanNm_00052
Requirements / Reference to Test Environment	Test SWC shall be able to read network state. e.g.: by creating a BSWM mode rule and action list. Test environment shall be capable to measure CAN signal timings….Test logging and reporting, timing measurements, access capabilities, injection capabilities…
Configuration Parameters	Parameter set: PS001
Summary	It shall be tested that the SUT switches from "Repeat Message State" to "Normal Operation State" when network is requested (with busload reduction inactive). By observing the NM messages sent by the SUT the correct states and the correct timings of state switching are measured. TC_CANNM_119: In the NM messages the Repeat Message Bit is cleared.
Needed Adaptation to other Releases	
Pre-conditions	No NM messages are on the bus The network is released by the test-application SWC The SUT is in "Bus-Sleep Mode"

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Perform a Nm_NetworkRequest to go to CanNm Repeat Message State by calling ComM_RequestComMode with COMM_FULL_COMMUNICATION	[LT] After <CANNM_MSG_CYCLE_OFFSET> NM messages are transmitted cyclically by the SUT with the cycle time <CANNM_MSG_CYCLE_TIME>.
Step 2	[CP] Wait for <CANNM_REPEAT_MESSAGE_TIME>	[LT] During this time, the SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME> After this time, the SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME>
Step 3	[LT] Send NM messages on the bus with the cycle time 0,8*<CANNM_MSG_REDUCED_TIME>	[LT] The SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME>
Post-conditions	The network is requested by the test-application SWC The SUT is in "Network Mode" (sub-state "Normal Operation State")	

7.3.9 [ATS_COMM_00045] Transition Repeat Message State to Normal Operation State + busload reduction active

Test Objective	Transition Repeat Message State to Normal Operation State + busload reduction active		
ID	ATS_COMM_00045	**AUTOSAR Releases**	3.2.2|4.0.3|4.1.1|4.2.1|4.2.2

Affected Modules	CANNm, Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00122		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00087 CANNetworkManagement: SWS_CanNm_00092 CANNetworkManagement: SWS_CanNm_00094 CANNetworkManagement: SWS_CanNm_00314 CANNetworkManagement: SWS_CanNm_00100 CANNetworkManagement: SWS_CanNm_00102 CANNetworkManagement: SWS_CanNm_00103 CANNetworkManagement: SWS_CanNm_00129 CANNetworkManagement: SWS_CanNm_00247 CANNetworkManagement: SWS_CanNm_00237 CANNetworkManagement: SWS_CanNm_00238 CANNetworkManagement: SWS_CanNm_00005 CANNetworkManagement: SWS_CanNm_00032 CANNetworkManagement: SWS_CanNm_00040 CANNetworkManagement: SWS_CanNm_00052 CANNetworkManagement: SWS_CanNm_00156 CANNetworkManagement: SWS_CanNm_00157 CANNetworkManagement: SWS_CanNm_00069		
Requirements / Reference to Test Environment	Test SWC shall be able to read network state. e.g.: by creating a BSWM mode rule and action list. Test environment shall be capable to send CanNM messages. Test environment shall be capable to measure CAN signal timings….Test logging and reporting, timing measurements, access capabilities, injection capabilities…		
Configuration Parameters	Parameter set: PS002		
Summary	It shall be tested that the SUT switches from "Repeat Message State" to "Normal Operation State" when network is requested (with busload reduction active). By observing the NM messages sent by the SUT the correct states and the correct timings of state switching are measured.		
Needed Adaptation to other Releases			
Pre-conditions	No NM messages are on the bus The network is released by the test-application SWC The SUT is in "Bus-Sleep Mode"		
Main Test Execution			
Test Steps		**Pass Criteria**	
Step 1	[SWC] Go to ComM_Full_Communication by calling ComM_RequestComMode with COMM_FULL_COMMUNICATION	[SWC] The SUT is in COMM_FULL_COMMUNICATION	
Step 2	-	[LT] After <CANNM_MSG_CYCLE_OFFSET> NM messages are transmitted cyclically by the SUT with the cycle time <CANNM_MSG_CYCLE_TIME>	

Step 3	[CP] Wait for <CANNM_REPEAT_MESSAGE_TIME>	[LT] During this time, the SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME>
Step 4	[LT] Send NM messages on the bus with the cycle time 1,8*<CANNM_MSG_REDUCED_TIME>	[LT] The SUT sends NM messages <CANNM_MSG_REDUCED_TIME> after reception of the NM messages on the bus (with the cycle time 1,8*<CANNM_MSG_REDUCED_TIME>)
Step 5	[LT] Send NM messages on the bus with the cycle time 0,8*<CANNM_MSG_REDUCED_TIME>	[LT] The SUT sends no NM messages
Step 6	[LT] Stop sending NM messages on the bus	[LT] The SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME>
Post-conditions	The network is requested by the test-application SWC The SUT is in "Network Mode" (sub-state "Normal Operation State")	

7.3.10 [ATS_COMM_00046] Transition Repeat Message State to Normal Operation State + immediate nm transmissions

Test Objective	Transition Repeat Message State to Normal Operation State + immediate nm transmissions		
ID	ATS_COMM_00046	**AUTOSAR Releases**	3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CANNm, Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00122		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00087 CANNetworkManagement: SWS_CanNm_00092 CANNetworkManagement: SWS_CanNm_00094 CANNetworkManagement: SWS_CanNm_00314 CANNetworkManagement: SWS_CanNm_00100 CANNetworkManagement: SWS_CanNm_00102 CANNetworkManagement: SWS_CanNm_00103 CANNetworkManagement: SWS_CanNm_00129 CANNetworkManagement: SWS_CanNm_00247 CANNetworkManagement: SWS_CanNm_00237 CANNetworkManagement: SWS_CanNm_00238 CANNetworkManagement: SWS_CanNm_00005 CANNetworkManagement: SWS_CanNm_00334 CANNetworkManagement: SWS_CanNm_00335 CANNetworkManagement: SWS_CanNm_00032 CANNetworkManagement: SWS_CanNm_00040 CANNetworkManagement: SWS_CanNm_00052 CANNetworkManagement: SWS_CanNm_00156 CANNetworkManagement: SWS_CanNm_00157		

	CANNetworkManagement: SWS_CanNm_00069	
Requirements / Reference to Test Environment	Test SWC shall be able to read network state. e.g.: by creating a BSWM mode rule and action list. Test environment shall be capable to send CanNM messages. Test environment shall be capable to measure CAN signal timings….Test logging and reporting, timing measurements, access capabilities, injection capabilities…	
Configuration Parameters	Parameter set: PS003	
Summary	It shall be tested that the SUT switches from "Repeat Message State" to "Normal Operation State" when network is requested (with busload reduction active). By observing the NM messages sent by the SUT the correct states and the correct timings of state switching are measured.	
Needed Adaptation to other Releases		
Pre-conditions	No NM messages are on the bus The network is released by the test-application SWC The SUT is in "Bus-Sleep Mode"	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Go to ComM_Full_Communication by calling ComM_RequestComMode with COMM_FULL_COMMUNICATION	[SWC] The SUT is in COMM_FULL_COMMUNICATION.
Step 2	-	[LT] After <CANNM_MSG_CYCLE_OFFSET> NM messages are transmitted cyclically by the SUT with the cycle time <CANNM_MSG_CYCLE_TIME>
Step 3	[CP] Wait <CANNM_IMMEDIATE_NM_TRANSMISSIONS> * <CANNM_IMMEDIATE_NM_CYCLE_TIME>	[LT] During this time NM messages are sent cyclically by the SUT with the cycle time <CANNM_IMMEDIATE_NM_CYCLE_TIME>
Step 4	[CP] Wait <CANNM_MSG_CYCLE_TIME>	[LT] After this time, the SUT start to send NM messages with the cycle time <CANNM_MSG_CYCLE_TIME>
Step 5	[CP] Wait CANNM_REPEAT_MESSAGE_TIME relative to Main 1	[LT] During this time, the SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME>
Step 6	[LT] Stop sending NM messages on the bus	[LT] The SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME>
Post-conditions	The network is requested by the test-application SWC The SUT is in "Network Mode" (sub-state "Normal Operation State")	

7.3.11 [ATS_COMM_00047] Transition Repeat Message State to Ready Sleep State + no immediate nm transmissions

Test Objective	Transition Repeat Message State to Ready Sleep State + no immediate nm transmissions		
ID	ATS_COMM_00047	**AUTOSAR Releases**	3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CANNm, Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00122		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00314 CANNetworkManagement: SWS_CanNm_00100 CANNetworkManagement: SWS_CanNm_00102 CANNetworkManagement: SWS_CanNm_00106 CANNetworkManagement: SWS_CanNm_00247 CANNetworkManagement: SWS_CanNm_00237 CANNetworkManagement: SWS_CanNm_00334		
Requirements / Reference to Test Environment	Test SWC shall be able to read network state. e.g.: by creating a BSWM mode rule and action list. Test environment shall be capable to measure CAN signal timings….Test logging and reporting, timing measurements, access capabilities, injection capabilities…		
Configuration Parameters	Parameter set: PS002		
Summary	It shall be tested that the SUT switches from "Repeat Message State" to "Ready Sleep State" when network is released. By observing the NM messages sent by the SUT the correct states and the correct timings of state switching are measured.		
Needed Adaptation to other Releases			
Pre-conditions	No NM messages are on the bus The network is released by the test-application SWC The SUT is in "Bus-Sleep Mode"		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[LT] The test environment start sending NM message on the bus with a cycle time of <CANNM_MSG_CYCLE_TIME>.		[LT] After <CANNM_MSG_CYCLE_OFFSET NM> NM messages are transmitted cyclically by the SUT with the cycle time <CANNM_MSG_CYCLE_TIME>.
Step 2	[CP] Wait for <CANNM_REPEAT_MESSAGE_TIME>		[LT] After this time: The SUT sends no NM messages
Post-conditions	The network is requested by the test-application SWC The SUT is in "Network Mode" (sub-state "Ready Sleep State")		

7.3.12 [ATS_COMM_00048] Transition Normal Operation State to Repeat Message State +repeat message bit received

Test Objective	Transition Normal Operation State to Repeat Message State +repeat message bit received		
ID	ATS_COMM_00048	**AUTOSAR Releases**	4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CANNm, Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00122		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00100 CANNetworkManagement: SWS_CanNm_00102 CANNetworkManagement: SWS_CanNm_00103 CANNetworkManagement: SWS_CanNm_00119 CANNetworkManagement: SWS_CanNm_00247 CANNetworkManagement: SWS_CanNm_00237 CANNetworkManagement: SWS_CanNm_00238 CANNetworkManagement: SWS_CanNm_00005 CANNetworkManagement: SWS_CanNm_00334 CANNetworkManagement: SWS_CanNm_00335 CANNetworkManagement: SWS_CanNm_00052 CANNetworkManagement: SWS_CanNm_00156 CANNetworkManagement: SWS_CanNm_00069 CANNetworkManagement: SWS_CanNm_00135 CANNetworkManagement: SWS_CanNm_00014		
Requirements / Reference to Test Environment	Test SWC shall be able to read network state. e.g.: by creating a BSWM mode rule and action list. Test environment shall be capable to measure CAN signal timings….Test logging and reporting, timing measurements, access capabilities, injection capabilities…		
Configuration Parameters	Parameter set: PS003		
Summary	Coming from "Normal Operation State" with busload reduction enabled the SUT switches to "Repeat Message State" on reception of Repeat Message Request Bit in received NM messages. By observing the NM messages sent by the SUT the correct states and the correct timings of state switching are measured.		
Needed Adaptation to other Releases	Needed adaptation for release [3.2.2] *Configuration:* Parameter set **PS002** *Test steps:* Expected Results for **Steps 2, 3** and **4** • Step 2 - Expected Results: The SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME> and Nm_RepeatMessageIndication() is called • Step 3 - Expected Results: The SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME> • Step 4 - Expected Results: During this time, the SUT sends NM messages with the cycle time CANNM_MSG_CYCLE_TIME and Nm_RepeatMessageIndication() is called. After this time, the SUT is in Normal Operation State		
Pre-conditions	No NM messages are on the bus The network is requested by the test-application SWC The SUT is in "Normal Operation State"		

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] Send NM messages with Repeat Message Request Bit cleared on the bus with the cycle time 0,8*<CANNM_MSG_REDUCED_TIME>	[LT] The SUT sends no NM messages
Step 2	[LT] Send one NM message with Repeat Message Request Bit set on the bus	[LT] The SUT starts sending NM messages with the cycle time <CANNM_MSG_CYCLE_TIME>
Step 3	[LT] Send NM messages with Repeat Message Request Bit cleared on the bus with the cycle time 0,8*<CANNM_MSG_REDUCED_TIME>	[LT] The SUT keep sending NM messages with the cycle time <CANNM_IMSG_CYCLE_TIME> .
Step 4	[CP] Wait <CANNM_REPEAT_MESSAGE_TIME> relative to Main2	[LT] During this time, the SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME>
Step 5	[SWC] Get status of NM via Get Status API	[SWC] After this time, the SUT is in Normal Operation State
Post-conditions	The network is requested by the test-application SWC The SUT is in "Network Mode" (sub-state "Normal Operation State")	

7.3.13 [ATS_COMM_00049] Transition Ready Sleep State to Repeat Message State + repeat message bit received

Test Objective	Transition Ready Sleep State to Repeat Message State + repeat message bit received		
ID	ATS_COMM_00049	**AUTOSAR Releases**	4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CANNm, Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00122		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00100 CANNetworkManagement: SWS_CanNm_00102 CANNetworkManagement: SWS_CanNm_00103 CANNetworkManagement: SWS_CanNm_00111 CANNetworkManagement: SWS_CanNm_00247 CANNetworkManagement: SWS_CanNm_00237 CANNetworkManagement: SWS_CanNm_00005 CANNetworkManagement: SWS_CanNm_00334 CANNetworkManagement: SWS_CanNm_00335 CANNetworkManagement: SWS_CanNm_00052 CANNetworkManagement: SWS_CanNm_00156 CANNetworkManagement: SWS_CanNm_00069 CANNetworkManagement: SWS_CanNm_00135 CANNetworkManagement: SWS_CanNm_00014		

Requirements / Reference to Test Environment	Test SWC shall be able to read network state. e.g.: by creating a BSWM mode rule and action list. Test environment shall be capable to measure CAN signal timings….Test logging and reporting, timing measurements, access capabilities, injection capabilities…
Configuration Parameters	Parameter set: PS003
Summary	Coming from "Normal Operation State" the SUT switches to "Ready Sleep State" when calling ComM_RequestComMode with COMM_NO_COMMUNICATION and to "Repeat Message State" on reception of Repeat Message Request Bit in received NM messages (with busload reduction active). By observing the NM messages sent by the SUT the correct states and the correct timings of state switching are measured.
Needed Adaptation to other Releases	Needed adaptation for release [3.2.2] *Configuration:* Parameter set **PS001** *Test steps:* **Step 1**; expected results for **Step 3** • Step 1 - Step Action: Send NM message with Repeat Message Request Bit cleared with cycle time <CANNM_MSG_CYCLE_TIME> on the bus • Step 1 - Expected Results: The SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME> • Step 3 - Expected Results: After <CANNM_MSG_CYCLE_OFFSET> the SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME> and NmRepeatMessageIndication() is called.
Pre-conditions	No NM messages are on the bus The network is requested by the test-application SWC The SUT is in "Normal Operation State"

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] Send NM message with Repeat Message Request Bit cleared with cycle time 1,8*<CANNM_MSG_REDUCED_TIME> on the bus	[LT] The SUT sends NM messages <CANNM_MSG_REDUCED_TIME> after reception of the NM messages on the bus (with the cycle time 1,8*<CANNM_MSG_REDUCED_TIME>)
Step 2	[SWC] Release the Network by calling ComM_RequestComMode with COMM_NO_COMMUNICATION. Continue sending NM messages.	[LT] The SUT sends no NM messages
Step 3	[LT] Send one NM message with Repeat Message Request Bit set on the bus	[LT] The SUT starts sending NM messages with the cycle time CANNM_MSG_CYCLE_TIME
Step 4	[CP] Wait <CANNM_REPEAT_MESSAGE_TIME> relative to Main3.	[LT] During this time, the SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME>
Step 5	[LT] Continue sending NM messages with Repeat Message Request Bit cleared.	[LT] After this time, the SUT sends no NM

		messages.
Post-conditions	The network is released by the test-application SWC The SUT is in "Network Mode" (sub-state " Ready Sleep State")	

7.3.14 [ATS_COMM_00280] Transition Prepare Bus Sleep Mode to Network Mode on NM Rx Indication

Test Objective	Transition Prepare Bus Sleep Mode to Network Mode on NM Rx Indication		
ID	ATS_COMM_00280	**AUTOSAR Releases**	3.2.2|4.0.3|4.1.1|4.2.1|4.2.2
Affected Modules	CanNm, Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00122		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00092 CANNetworkManagement: SWS_CanNm_00094 CANNetworkManagement: SWS_CanNm_00315 CANNetworkManagement: SWS_CanNm_00100 CANNetworkManagement: SWS_CanNm_00102 CANNetworkManagement: SWS_CanNm_00106 CANNetworkManagement: SWS_CanNm_00108 CANNetworkManagement: SWS_CanNm_00115 CANNetworkManagement: SWS_CanNm_00124 CANNetworkManagement: SWS_CanNm_00247 CANNetworkManagement: SWS_CanNm_00248 CANNetworkManagement: SWS_CanNm_00237 CANNetworkManagement: SWS_CanNm_00238 CANNetworkManagement: SWS_CanNm_00032 CANNetworkManagement: SWS_CanNm_00040 CANNetworkManagement: SWS_CanNm_00051 CANNetworkManagement: SWS_CanNm_00052 CANNetworkManagement: SWS_CanNm_00156		
Requirements / Reference to Test Environment	Test SWC shall be able to call ComM_RequestComMode to release or request the communication Test environment shall be capable to measure CAN signal timings….Test logging and reporting, timing measurements, access capabilities, injection capabilities…		
Configuration Parameters	Parameter set: PS002		
Summary	It shall be tested that the SUT does not perform any bus load reduction while entering to "Network Mode" (sub-state "Repeat Message State") from "Prepare Bus-Sleep Mode" [SWS_CanNm_00124][SWS_CanNm_00315] due to NM messages on the CAN bus.		

	After reception of NM messages from the bus (with modified cycle time) the SUT starts transmitting cyclically NM messages [SWS_CanNm_00100] itself with a cycle time of <CANNM_MSG_CYCLE_TIME> [SWS_CanNm_00032][SWS_CanNm_00040][SWS_CanNm_00237]. The bus load reduction is disabled [SWS_CanNm_00156], so the SUT sends with standard cycle time. After transmitting NM messages by the SUT for <CANNM_REPEAT_MESSAGE_TIME> [SWS_CanNm_00102] the SUT will switch into the sub-state "Ready Sleep State" of the mode "Network Mode" [SWS_CanNm_00106] and no NM messages are transmitted anymore by the SUT [SWS_CanNm_00108].
Needed Adaptation to other Releases	
Pre-conditions	The SUT is in "Prepare Bus-Sleep Mode" for less than "CANNM_WAIT_BUS_SLEEP_TIME". The network is released by the test-application SWC. No NM messages are on the bus.

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] The test environment sends NM messages on the bus with a cycle time 0,8*<CANNM_MSG_REDUCED_TIME>.	[LT] The SUT sends NM messages [SWS_CanNm_00100] on the bus with a cycle time of <CANNM_MSG_CYCLE_TIME> [SWS_CanNm_00032][SWS_CanNm_00040][SWS_CanNm_00237] [SWS_CanNm_00156] (the absence of NM messages from SUT would indicate a running bus load reduction).
Step 2	[CP] Wait for <CANNM_REPEAT_MESSAGE_TIME> [SWS_CanNm_00102] to trigger transition "Repeat Message State" to “Ready Sleep State” [SWS_CanNm_00106].	[LT] After this time, there are just the NM messages from the test environment on the bus and no messages from SUT anymore [SWS_CanNm_00108].
Post-conditions	The SUT is in "Network Mode" (sub-state "Ready Sleep State"). The network is released by the test-application SWC.	

7.3.15 [ATS_COMM_00315] Transition Normal Operation State to Repeat Message State + Repeat Msg State request from SWC

Test Objective	Transition Normal Operation State to Repeat Message State + Repeat Msg State request from SWC		
ID	ATS_COMM_00315	**AUTOSAR Releases**	4.2.1|4.2.2
Affected Modules	CanNm, Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00122		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00100 CANNetworkManagement: SWS_CanNm_00102 CANNetworkManagement: SWS_CanNm_00103		

	CANNetworkManagement: SWS_CanNm_00119 CANNetworkManagement: SWS_CanNm_00247 CANNetworkManagement: SWS_CanNm_00237 CANNetworkManagement: SWS_CanNm_00238 CANNetworkManagement: SWS_CanNm_00005 CANNetworkManagement: SWS_CanNm_00334 CANNetworkManagement: SWS_CanNm_00335 CANNetworkManagement: SWS_CanNm_00052 CANNetworkManagement: SWS_CanNm_00156 CANNetworkManagement: SWS_CanNm_00069 CANNetworkManagement: SWS_CanNm_00135 CANNetworkManagement: SWS_CanNm_00014
Requirements / Reference to Test Environment	Test SWC shall be able to read network state. e.g.: by creating a BSWM mode rule and action list Test environment shall be capable to measure CAN signal timing, Test logging and reporting, timing measurements, access capabilities, injection capabilities.
Configuration Parameters	Parameter set: PS003
Summary	Coming from "Normal Operation State" with busload reduction enabled the SUT switches to "Repeat Message State" on reception of Repeat Message Request from application (SWC). By observing the NM messages sent by the SUT the correct states and the correct timings of state switching are measured.
Needed Adaptation to other Releases	Needed adaptation for release [3.2.2] *Configuration:* Parameter set **PS002** *Test steps:* Expected Results for **Steps 2, 3** and **4** • Step 2 - Expected Results: The SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME> and Nm_RepeatMessageIndication() is called • Step 3 - Expected Results: The SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME> • Step 4 - Expected Results: During this time, the SUT sends NM messages with the cycle time CANNM_MSG_CYCLE_TIME and Nm_RepeatMessageIndication() is called. After this time, the SUT is in Normal Operation State
Pre-conditions	No NM messages are on the bus The network is requested by the test-application SWC The SUT is in "Normal Operation State"

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] Send NM message with Repeat Message Request Bit cleared with cycle time 0.8*<CANNM_MSG_REDUCED_TIME> on the bus	[LT] The SUT sends no NM message.
Step 2	[SWC] Request NM for Repeat Message using Nm_RepeatMessageRequest	[LT] The SUT starts sending NM messages with the cycle time <CANNM_MSG_CYCLE_TIME> with repeat message bit set
Step 3	[LT] Send NM messages with Repeat Message	[LT] The SUT continues sending NM message every

	Request Bit Cleared on the bus with the cyle time 0.8*<CANNM_MSG_REDUCED_TIME>	<CANNM_MSG_CYCLE_TIME> with repeat message bit set
Step 4	[CP] Wait <CANNM_REPEAT_MESSAGE_TIME> relative to Step 2	[LT] During this time, SUT continues sending NM messages with the cycle time <CANNM_MSG_CYCLE_TIME> with repeat message bit set
Step 5	-	[SWC] After this time, the SUT is in Normal Operation state
Post-conditions		

7.3.16 [ATS_COMM_00316] Transition Ready Sleep State to Repeat Message State + repeat message state requested from SWC

Test Objective	Transition Ready Sleep State to Repeat Message State + repeat message state requested from SWC		
ID	ATS_COMM_00316	**AUTOSAR Releases**	4.2.1|4.2.2
Affected Modules	CanNm, Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document	ATR: ATR_ATR_00102 ATR: ATR_ATR_00122		
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00100 CANNetworkManagement: SWS_CanNm_00102 CANNetworkManagement: SWS_CanNm_00103 CANNetworkManagement: SWS_CanNm_00111 CANNetworkManagement: SWS_CanNm_00247 CANNetworkManagement: SWS_CanNm_00237 CANNetworkManagement: SWS_CanNm_00005 CANNetworkManagement: SWS_CanNm_00334 CANNetworkManagement: SWS_CanNm_00335 CANNetworkManagement: SWS_CanNm_00052 CANNetworkManagement: SWS_CanNm_00156 CANNetworkManagement: SWS_CanNm_00069 CANNetworkManagement: SWS_CanNm_00135 CANNetworkManagement: SWS_CanNm_00014		
Requirements / Reference to Test Environment	Test SWC shall be able to read network state. e.g.: by creating a BSWM mode rule and action list Test environment shall be capable to measure CAN signal timings….Test logging and reporting, timing measurements, access capabilities, injection capabilities…		
Configuration Parameters	Parameter set: PS003		
Summary	Coming from "Normal Operation State" the SUT switches to "Ready Sleep State" when calling ComM_RequestComMode with COMM_NO_COMMUNICATION and to "Repeat Message State" on reception of Repeat Message Request from SWC. By observing the NM messages sent by the SUT the correct states and the correct timings of state switching are measured.		

Needed Adaptation to other Releases	Needed adaptation for release [3.2.2] *Configuration:* Parameter set **PS001** *Test steps:* **Step 1**; expected results for **Step 3** • Step 1 - Step Action: Send NM message with Repeat Message Request Bit cleared with cycle time <CANNM_MSG_CYCLE_TIME> on the bus • Step 1 - Expected Results: The SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME> • Step 3 - Expected Results: After <CANNM_MSG_CYCLE_OFFSET> the SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME> and NmRepeatMessageIndication() is called.	
Pre-conditions	No NM messages are on the bus The network is requested by the test-application SWC The SUT is in "Normal Operation State"	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] Send NM message with Repeat Message Request Bit cleared with cycle time 1,8*<CANNM_MSG_REDUCED_TIME> on the bus	[LT] The SUT sends NM messages <CANNM_MSG_REDUCED_TIME> after reception of the NM messages on the bus (with the cycle time 1,8*<CANNM_MSG_REDUCED_TIME>)
Step 2	[SWC] Release the Network by calling ComM_RequestComMode with COMM_NO_COMMUNICATION.	[LT] The SUT sends no NM messages
Step 3	[LT] Continue sending NM messages	[SWC] The SUT is in Ready Sleep State
Step 4	[SWC] Request NM for Repeat Message using Nm_RepeatMessageRequest	[LT] The SUT starts sending NM messages with repeat message bit set for every <CANNM_MSG_CYCLE_TIME>
Step 5	[CP] Wait until <CANNM_REPEAT_MESSAGE_TIME>	[LT] During this time, the SUT sends NM messages with the cycle time <CANNM_MSG_CYCLE_TIME> with repeat message bit set.
Step 6	-	[SWC] After this time, the SUT is in Ready Sleep State
Post-conditions		

7.3.17 [ATS_COMM_00779] Evaluation Of The Received NM-PDU And Checking For Enabling And Disabling The Particular PNC

Test Objective	Evaluation Of The Received NM-PDU And Checking For Enabling And Disabling The Particular PNC		
ID	ATS_COMM_00779	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00415 CANNetworkManagement: SWS_CanNm_00416		
Requirements / Reference to Test Environment	none		
Configuration Parameters	CanNmAllNmMessagesKeepAwake = FALSE CANNM_PN_E-BLED = TRUE CANNM_COM_USER_DATA_SUPPORT = TRUE CanNmPnInfoOffset = 2 CanNmPnInfoLength = 1 CanNmPnFilterMaskByteIndex = 2 CanNmPnFilterMaskByteValue = 0x01		
Summary	CanSM confirms the PN availability to CanNm by calling CanNm_ConfirmPnAvailability API so NM-PDU filter algorithm of PN messages shall be available on the indicated channel. One PNC is configured mapping to Tx and Rx PDU groups		
Needed Adaptation to other Releases			
Pre-conditions	DUT shall be initialized		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[LT] Transmit a valid frame from the tester to the DUT (passive wakeup)		-
Step 2	[LT] Transmit a NM frame from the tester with Repeat Message Request (CBV bit 0) is set periodically to DUT		[LT] NM frames related to PNC1 shall not be observed
Step 3	[LT] Transmit a NM frame from the tester with Repeat Message Request (CBV bit 0) is not set periodically to DUT		[LT] NM frames related to PNC1 shall not be observed
Post-conditions	None		

7.3.18 [ATS_COMM_00780] Evaluation Of The Received NM-PDU For Enabling And Disabling Particular PNC And Also Filtering The Irrelevant Bit In The Received NM Frame

Test Objective	Evaluation Of The Received NM-PDU For Enabling And Disabling Particular PNC And Also Filtering The Irrelevant Bit In The Received NM Frame		
ID	ATS_COMM_00780	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00417 CANNetworkManagement: SWS_CanNm_00418 CANNetworkManagement: SWS_CanNm_00419 CANNetworkManagement: SWS_CanNm_00436		
Requirements / Reference to Test Environment	none		
Configuration Parameters	CanNmAllNmMessagesKeepAwake = FALSE CANNM_PN_E-BLED = TRUE CanNmPnInfoOffset = 2 CanNmPnInfoLength = 1 CanNmPnFilterMaskByteIndex = 2 CanNmPnFilterMaskByteValue = 0x01		
Summary	CanSM confirms the PN availability to CanNm by calling CanNm_ConfirmPnAvailability API so NM-PDU filter algorithm of PN messages shall be available on the indicated channel. To check the behaviour of filter algorithm when receiving PN bit as set and which is not configured in this ECU		
Needed Adaptation to other Releases			
Pre-conditions	DUT shall be initialized		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[LT] Transmit a valid frame from the tester to the DUT (passive wakeup)		-
Step 2	[LT] Transmit a NM frame from the tester with 2ndbyte first bit is set periodically to DUT		[LT] NM frames related to PNC1 shall be observed
Step 3	[LT] Transmit NM frame from the tester with 2ndbyte first bit cleared periodically to DUT		[LT] NM frames related to PNC1 shall not be observed
Step 4	[LT] Transmit a NM frame from the tester with 2ndbyte second bit is set which is not configured for this ECU periodically to DUT		[LT] NM frames related to PNC1 shall not be observed
Post-conditions	None		

7.3.19 [ATS_COMM_00781] Setting The Source Node Identifier In NM-PDU

Test Objective	Setting The Source Node Identifier In NM-PDU		
ID	ATS_COMM_00781	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00013		
Requirements / Reference to Test Environment	none		
Configuration Parameters	CANNM_PDU_NID_POSITION = CANNM_PDU_BYTE_0 CANNM_NODE_ID = 0x03 CANNM_PASSIVE_MODE_ENABLED = FALSE CANNM_NODE_DETECTION_ENABLED = TRUE CANNM_USER_DATA_ENABLED = TRUE CanNmNodeIdEnabled = TRUE		
Summary	Configure the Source Node Identifier and check the NM frames for the configured Source Node Identifier on the bus.		
Needed Adaptation to other Releases			
Pre-conditions	DUT shall be initialized		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[SWC] Request ComM to be in COMM_FULL_COMMUNICATION mode		[SWC] RTE shall return E_OK
Step 2	[SWC] Check the current ComM mode for the channel		[SWC] Successfully changed to the COMM_FULL_COMMUNICATION mode
Step 3	-		[LT] NM Frame shall be observed on the Bus with configured source node
Post-conditions	None		

7.3.20 [ATS_COMM_00782] Getting The NM-PDU Data Out Of The Most Recently Received NM Message

Test Objective	Getting The NM-PDU Data Out Of The Most Recently Received NM Message		
ID	ATS_COMM_00782	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2

Affected Modules	CANNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00138		
Requirements / Reference to Test Environment	none		
Configuration Parameters	CANNM_NODE_DETECTION_E-BLED = TRUE CANNM_USER_DATA_E-BLED = TRUE CANNM_PDU_NID_POSITION = CANNM_PDU_BYTE_0		
Summary	Nm will get the whole Pdu Data out of the most recently received NM message by calling CanNm. This is from OEM specific extensions of the NM stack which shall be used to analyse PDU Data.		
Needed Adaptation to other Releases			
Pre-conditions	ComM module shall be in FULL communication		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[LT] Transmit NM frame from the tester periodically to DUT with Node ID1 CBV and User Data		-
Step 2	[SWC] Request the whole PDU data (Node ID Control Bit Vector and User Data) of the most recently received NM PDU using NM Get PDU data interface		[SWC] PDU data must be received with node ID1 from the most recently received NM message
Post-conditions	None		

7.3.21 [ATS_COMM_00783] Status Of The Control Bit Vector Set Or Reset For The Active Wake Up Bit (Bit 4)

Test Objective	Status Of The Control Bit Vector Set Or Reset For The Active Wake Up Bit (Bit 4)		
ID	ATS_COMM_00783	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00401 CANNetworkManagement: SWS_CanNm_00402		
Requirements / Reference	none		

to Test Environment		
Configuration Parameters	CanNmActiveWakeupBitEnabled = TRUE	
Summary	If the active wakeup bit is enabled in CanNm (by a COMM_FULL_COMMUNICATION request) then CanNm shall set the active wakeup bit in the CBV due to an active wakeup. When the CanNm leaves the network mode check if the active wakeup bit is reset.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Request ComM to be in COMM_FULL_COMMUNICATION	[SWC] RTE shall return E_OK
Step 2	[SWC] Request the current ComM Mode	[SWC] ComM mode shall be COMM_FULL_COMMUNICATION
Step 3	-	[LT] NM Frame shall be observed on the Bus with bit 4 of CBV byte as set
Step 4	[SWC] Request ComM to be in COMM_NO_COMMUNICATION	[SWC] RTE shall return E_OK
Step 5	[SWC] Request the current ComM Mode	[SWC] ComM mode shall be COMM_NO_COMMUNICATION
Step 6	-	[LT] Application frames and NM frames shall not be observed
Post-conditions	None	

7.3.22 [ATS_COMM_00784] Enabling Of Communication Functionality In Case Of NM-PDU Transmission Is Disabled

Test Objective	Enabling Of Communication Functionality In Case Of NM-PDU Transmission Is Disabled		
ID	ATS_COMM_00784	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00176 CANNetworkManagement: SWS_CanNm_00295		
Requirements / Reference	none		

to Test Environment		
Configuration Parameters	CANNM_PASSIVE_MODE_E-BLED = FALSE CANNM_COM_CONTROL_E-BLED = TRUE	
Summary	Using Diagnostic Service 0x28 for Communication Control the Enabling and disabling the PDU is achieved.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Request ComM to be in COMM_FULL_COMMUNICATION	[SWC] RTE shall return E_OK
Step 2	[SWC] Request the current ComM mode	[SWC] ComM shall be in FULL_COMMUNICATION mode
Step 3	-	[LT] NM Frame shall be observed on the Bus
Step 4	[LT] Communication Control (0x28) service for disabling the NM PDU is invoked	[LT] NM frames shall not be observed only application frames shall be observed
Step 5	[LT] Communication Control (0x28) service for enabling the NM PDU is invoked	[LT] NM frames and application frames shall be observed
Post-conditions	None	

7.3.23 [ATS_COMM_00785] Getting The Node Id Of The Recently Received NM Message And Of The Local Node Identifier For The Local Host Node

Test Objective	Getting The Node Id Of The Recently Received NM Message And Of The Local Node Identifier For The Local Host Node		
ID	ATS_COMM_00785	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00132 CANNetworkManagement: SWS_CanNm_00220 CANNetworkManagement: SWS_CanNm_00133		
Requirements / Reference to Test Environment	none		
Configuration Parameters	NmNodeIdEnabled = TRUE CANNM_PASSIVE_MODE_E-BLED = FALSE		

	CANNM_NODE_DETECTION_E-BLED = TRUE CANNM_NODE_ID = 0x03 CANNM_USER_DATA_E-BLED = TRUE CANNM_USER_DATA_LENGTH = 0x06 CANNM_PDU_CBV_POSITION = CANNM_PDU_BYTE_1 CANNM_PDU_NID_POSITION = CANNM_PDU_BYTE_0 CANNM_PDU_RX_INDICATION_E-BLED = True
Summary	Nm will get the node identifier out of the last successfully received NM message by calling CanNm to get the node Id and Nm will get the node identifier configured for the local node by calling CanNm for that purpose.
Needed Adaptation to other Releases	
Pre-conditions	ComM shall be in full communication
Main Test Execution	

Test Steps		**Pass Criteria**
Step 1	[LT] Transmit Nm frames periodically from the tester with node identifier and user data	-
Step 2	[SWC] Get the node identifier in DUT out of the most recently received Network Management PDU	[SWC] The Nm node id pointer shall be loaded with the node identifier out of the last successfully received NM message with user data
Step 3	[SWC] Get the node identifier configured for the local host node using Get Local Node Identifier	[SWC] The node id configured for the local node shall be read
Step 4	-	[LT] Local ID shall be observed in the NM Frame transmitted by the DUT
Post-conditions	NONE	

7.3.24 [ATS_COMM_00786] Behavioral Check When Application Is Requesting For Repeat Message State In Repeat Message State

Test Objective	Behavioral Check When Application Is Requesting For Repeat Message State In Repeat Message State		
ID	ATS_COMM_00786	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00137		
Requirements / Reference to Test Environment	none		

Configuration Parameters	NM_NODE_DETECTION_E-BLED = TRUE CANNM_REPEAT_MESSAGE_TIME = 0.1 CANNM_REPEAT_MSG_IND_E-BLED = TRUE CANNM_NODE_DETECTION_E-BLED = TRUE	
Summary	Nm will set the repeat message request bit for the NM messages transmitted next on the bus and for that purpose the service of CanNm shall be used.	
Needed Adaptation to other Releases		
Pre-conditions	ComM module shall be in FULL communication	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Request ComM to be in COMM_FULL_COMMUNICATION	[SWC] E_OK shall be returned
Step 2	[SWC] Request for the NM states	[SWC] NM mode shall be in network mode and NM state shall be in Repeat message
Step 3	[SWC] Before the CanNm repeat message time expires request Nm to set the repeat message request bit	[SWC] Request is discarded with a return value E_NOT_OK
Post-conditions	NONE	

7.3.25 [ATS_COMM_00787] Behavioural Check For Reception Path When CarWakeUp Is Enabled And To Get The PDU Data For The NM-PDU Data That Causes The CarWakeUp Indication

Test Objective	Behavioural Check For Reception Path When CarWakeUp Is Enabled And To Get The PDU Data For The NM-PDU Data That Causes The CarWakeUp Indication		
ID	ATS_COMM_00787	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00405 CANNetworkManagement: SWS_CanNm_00406 CANNetworkManagement: SWS_CanNm_00407		
Requirements / Reference to Test Environment	none		
Configuration Parameters	NmCarWakeUpRxEnabled = TRUE CANNM_CAR_WAKE_UP_RX_E-BLED = TRUE CANNM_CAR_WAKE_UP_FILTER_E-BLED = FALSE CANNM_CAR_WAKE_UP_BYTE_POSITION = 2 CANNM_CAR_WAKE_UP_BIT_POSITION = 0		

Summary	Every ECUs of the Car shall be able to wake up all other ECUs of the Car and keep them awake. This includes all ECUs independent of the used bus system and the position in the topology of the Car. To provide the request for a “CarWakeUp” (CWU) in an efficient and standardized way the request shall be contained in the NM Message of the ECU. The requesting ECU sends its own NM message with a CWU request. To check the nmPduData when CanNm will call Nm to get the Pdu data which causes the call of car wakeup indication in Nm.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be in sleep mode	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] Transmit a valid NM frame periodically from tester to DUT with car wakeup bit set	[SWC] Car WakeUp Indication shall be invoked by CanNm
Step 2	[SWC] Get Pdu data out of the last successfully received NM message using Get PDU Data interface	[SWC] NM PDU with the Car wake up Indication as set shall be provided
Step 3	[SWC] Request to get current ComM mode	[SWC] ComM shall be in COMM_FULL_COMMUNICATION mode
Post-conditions	NONE	

7.3.26 [ATS_COMM_00788] Behavioural Check For Reception Path When CarWakeUpFilter Is Enabled

Test Objective	Behavioural Check For Reception Path When CarWakeUpFilter Is Enabled		
ID	ATS_COMM_00788	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00408		
Requirements / Reference to Test Environment	none		
Configuration Parameters	NmCarWakeUpRxEnabled = TRUE CANNM_CAR_WAKE_UP_RX_E-BLED = TRUE CANNM_CAR_WAKE_UP_FILTER_E-BLED = TRUE CANNM_CAR_WAKE_UP_FILTER_NODE_ID = 0x00 CANNM_CAR_WAKE_UP_BYTE_POSITION = 2 CANNM_CAR_WAKE_UP_BIT_POSITION = 0		

Summary	Every ECUs of the Car shall be able to wake up all other ECUs of the Car and keep them awake. This includes all ECUs independent of the used bus system and the position in the topology of the Car. To provide the request for a “CarWakeUp” (CWU) in an efficient and standardized way the request shall be contained in the NM Message of the ECU. The requesting ECU sends its own NM message with a CWU request. This Request is evaluated by the central Gateway. The Gateway wakes up all other connected communication channels. Additionally the Gateway provides the CWU request in its own NM message to enable sub Gateways to wake up the connected sub busses.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be in sleep mode	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT] Transmit a valid NM frame periodically from tester to DUT with the CanNm car wake up filter node id	[SWC] Nm will indicate about a reception of a CWU only when the CanNm car wake up filter node id is equal to that node id in the received NM PDU
Step 2	[SWC] Get Pdu data out of the last successfully received NM message and check the parameter nmPduData	[SWC] NM PDU with the Car wake up Indication as set shall be provided
Step 3	[SWC] Request to get current ComM mode	[SWC] ComM shall be in COMM_FULL_COMMUNICATION mode
Post-conditions	NONE	

7.3.27 [ATS_COMM_00789] Transition From Normal Operation To Ready Sleep State And Stays In Ready Sleep State

Test Objective	Transition From Normal Operation To Ready Sleep State And Stays In Ready Sleep State		
ID	ATS_COMM_00789	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00118		
Requirements / Reference to Test Environment	none		
Configuration Parameters	NmStateChangeIndEnabled = TRUE CANNM_REPEAT_MSG_IND_E-BLED = TRUE		

	CANNM_PDU_RX_INDICATION_E-BLED = TRUE CANNM_USER_DATA_E-BLED = TRUE CANNM_REPEAT_MESSAGE_TIME = 0.1 CANNM_TIMEOUT_TIME = 0.2	
Summary	Nm will return the state of the network management by calling CanNm for getting the states.	
Needed Adaptation to other Releases		
Pre-conditions	ComM module shall be in No communication	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Request ComM to be in COMM_FULL_COMMUNICATION	[SWC] E_OK shall be returned
Step 2	[CP] Wait until the configured CanNm repeat message time expires	-
Step 3	[SWC] Get the current NM state and mode	[SWC] NM mode shall be in network mode and NM state shall be normal operation
Step 4	[SWC] Request ComM to be in COMM_NO_COMMUNICATION	[SWC] E_OK shall be returned
Step 5	[SWC] Before the configured CanNm timeout time expires get the current NM state and mode	[SWC] NM mode shall be in network mode and NM state shall be ready sleep
Step 6	-	[LT] NM frames shall not be observed
Post-conditions	NONE	

7.3.28 [ATS_COMM_00790] Transition From Normal Operation To Ready Sleep State When The Application Releases The Network

Test Objective	Transition From Normal Operation To Ready Sleep State When The Application Releases The Network		
ID	ATS_COMM_00790	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	CANNM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	CANNetworkManagement: SWS_CanNm_00106 CANNetworkManagement: SWS_CanNm_00118		
Requirements / Reference to Test Environment	none		

Configuration Parameters	CANNM_PASSIVE_MODE_E-BLED = FALSE CANNM_NODE_DETECTION_E-BLED = TRUE CANNM_REPEAT_MSG_IND_E-BLED = TRUE CANNM_REPEAT_MESSAGE_TIME = 0.1 CANNM_TIMEOUT_TIME = 0.2	
Summary	Nm will set the repeat message request bit for the NM messages transmitted next on the bus and for that it will call CanNm.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Request ComM to be in COMM_FULL_COMMUNICATION	[SWC] E_OK shall be returned
Step 2	[SWC] Get the current ComM mode	[SWC] ComM shall be in COMM_FULL_COMMUNICATION
Step 3	[SWC] After the CanNm repeat message time expires Get the NM state and mode	[SWC] NM mode shall be in network mode and NM state shall be normal operation
Step 4	[SWC] Requests for setting repeat message request bit (Bit 0 in the CBV)	-
Step 5	[SWC] Before the CanNm repeat message timer expires get the NM state and mode	[SWC] NM mode shall be in network mode and NM state shall be repeat message
Step 6	[SWC] Request ComM to keep the DUT in COMM_NO_COMUNICATION	[SWC] E_OK shall be returned
Step 7	[SWC] Get the current ComM mode	[SWC] ComM shall be in COMM_NO_COMMUNICATION
Step 8	[SWC] Before the configured CANNM timeout time expires get the NM state and mode	[SWC] NM mode shall be in network mode and NM state shall be ready sleep
Post-conditions	NONE	

8 Miscellaneous features

8.1 General Test Objective and Approach

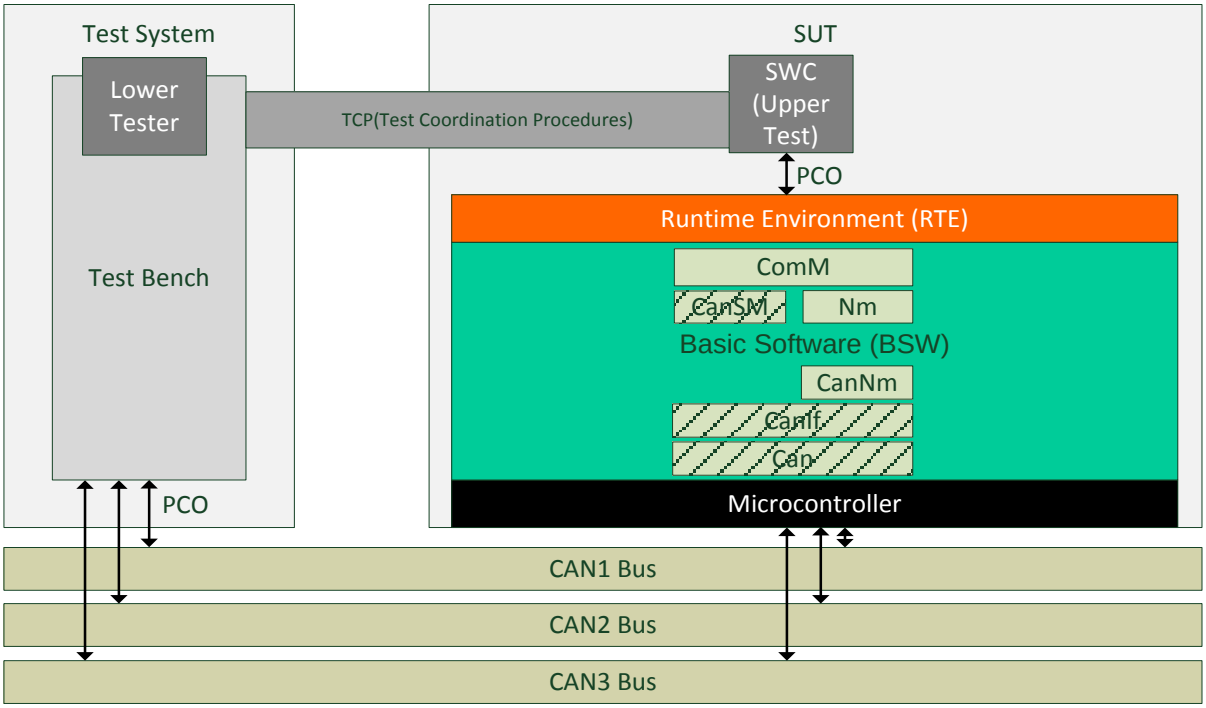
This test suite provides additional test cases for miscellaneous features of the Communication Management features, when they do not require a complete test suite on their own.

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

8.1.1 Test System

8.1.1.1 Overview on Architecture

The basic test setup is depicted in the following figure:



The test cases require a SWC as Upper Tester. This SWC indirectly stimulates the CAN NM functionality under test, and the behavior is observed through the CAN Bus.

8.1.1.2 Specific Requirements

None

8.1.1.3 Test Coordination Requirements

Not Applicable

8.1.2 Test Configuration

The configuration required to implement and execute the test cases is described in the “Configuration Parameters” field of each test case.

8.2 Re-usable Test Steps

No common test steps subsequence used in this test specification has been assessed as long enough for being considered as a test steps group.

8.3 Test Cases

8.3.1 [ATS_COMM_00759] Verify NmCarWakeUpCallback Notification

Test Objective	Verify NmCarWakeUpCallback Notification		
ID	ATS_COMM_00759	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	NetworkManagementInterface: SWS_Nm_00252 NetworkManagementInterface: SWS_Nm_00253		
Requirements / Reference to Test Environment	none		
Configuration Parameters	CoreTopology::CommunicationCluster = NM_BUSNM_CANNM NmStateChangeIndEnabled = TRUE NmCarWakeUpRxEnabled = TRUE NmCarWakeUpCallback = Nm_CarWakeUp_App NmUserDataEnabled = TRUE Signal configuration: Fibex::FibexCore::CoreCommunication::TransmissionModeTiming = DIRECT Fibex::FibexCore::CoreCommunication::Timing.EventControlledTiming.numberOfRepeats = Ex: 2 (user configurable) Fibex::FibexCore::CoreCommunication::EventControlledTiming.repetitionPeriod = Ex: 100 ms (user configurable) ComMBusType = CAN Callback function:NmCarWakeUpCallback		
Summary	If the CWU bit within any received NM Message is 1 and CanNmCarWakeUpRxEnabled is TRUE CanNm shall call Nm_CarWakeUpIndication.		
Needed Adaptation to other Releases			
Pre-conditions	DUT shall be initialized		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[LT] Transmit NM frame periodically to DUT with CWU (Car Wake Up) bit as set.		[SWC] NmCarWakeUpCallback shall be invoked.
Step 2	[LT] Stop transmitting NM frame.		[LT] NM frames shall not appear on bus.

Post-conditions	None

8.3.2 [ATS_COMM_00760] Setting the User Data In Nm Frame From SWC

Test Objective	Setting the User Data In Nm Frame From SWC		
ID	ATS_COMM_00760	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	NetworkManagementInterface: SWS_Nm_00035		
Requirements / Reference to Test Environment	none		
Configuration Parameters	CoreTopology::CommunicationCluster = NM_BUSNM_CANNM NmStateChangeIndEnabled = TRUE NmUserDataEnabled = TRUE NmPassiveModeEnabled = FALSE NmComControlEnabled = FALSE		
Summary	Using Rte_Write set the NM User data for NM messages transmitted next on the bus. Using Nm_GetState which is available as extra services provided by NM Interface the Nm_StateType and Nm_ModeType shall be probed. Hint: Nm will invoke the current state to NM (e.g. Notification that the network management has entered Network 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] Call Rte_Write to set NM User data.		[LT] observe NM frame on CAN channel with User data written by SWC.
Step 2	[SWC] SendComM_GetCurrentComMode request for current mode.		[SWC] ComM shall be in Full communication.
Step 3	[SWC] Send Nm_GetState request for NM states.		[SWC] NM shall be in Network Mode and state shall be in Repeat Message State
Post-conditions	NONE		

8.3.3 [ATS_COMM_00761] Getting the User Data From Received Nm Frame

Test Objective	Getting the User Data From Received Nm Frame		
ID	ATS_COMM_00761	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	NetworkManagementInterface: SWS_Nm_00036		
Requirements / Reference to Test Environment	none		
Configuration Parameters	CoreTopology::CommunicationCluster = NM_BUSNM_CANNM NmStateChangeIndEnabled = TRUE NmUserDataEnabled = TRUE NmPduRxIndicationEnabled = TRUE NmComControlEnabled = FALSE Callback function: Nm_PduRxIndication		
Summary	Using Nm_GetUserData get data out of the last successfully received NM message from SWC. Using Nm_GetState which is available as extra services provided by NM Interface the Nm_StateType and Nm_ModeType shall be probed. Hint : Nm will invoke the current state to NM (e.g. Notification that the network management has entered Network Mode).		
Needed Adaptation to other Releases			
Pre-conditions	ComM shall be in Full communication state		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[LT] Transmit NM frame periodically to DUT.		-
Step 2	[SWC] Send Nm_GetState request for NM states.		[SWC] NM shall be in Network Mode and Normal Operation state
Step 3	[SWC] Send Nm_GetUserData request to get user data received on NM frame in DUT.		[SWC] Received NM user data shall be the same as transmitted
Post-conditions	NONE		

8.3.4 [ATS_COMM_00762] Get Whole Pdu Data Out Of The Most Recently Received Nm Message

Test Objective	Get Whole Pdu Data Out Of The Most Recently Received Nm Message

ID	ATS_COMM_00762	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	NetworkManagementInterface: SWS_Nm_00037		
Requirements / Reference to Test Environment	none		
Configuration Parameters	CoreTopology::CommunicationCluster = NM_BUSNM_CANNM NM_COMM_NETWORK_HANDLE_REF = 0 NmStateChangeIndEnabled = TRUE NmPassiveModeEnabled = FALSE NmUserDataEnabled = TRUE NmNodeIdEnabled = TRUE Callback function: Nm_PduRxIndication		
Summary	To get the whole PDU data out of the most recently received NM message.		
Needed Adaptation to other Releases			
Pre-conditions	ComM shall be in Full communication state		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[LT] Transmit NM frame periodically to DUT.		[SWC] Nm_PduRxIndication callback function shall be invoked indicating reception of NM frame.
Step 2	[SWC] Send Nm_GetPduData request to get whole PDU data from latest received on NM frame in DUT.		[SWC] Received PDU data shall be the same as transmitted
Post-conditions	NONE		

8.3.5 [ATS_COMM_00764] Get Local Node Identifier for local node

Test Objective	Get Local Node Identifier for local node		
ID	ATS_COMM_00764	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	Nm	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	NetworkManagementInterface: SWS_Nm_00040 NetworkManagementInterface: SWS_Nm_00112		

Requirements / Reference to Test Environment	none		
Configuration Parameters	CoreTopology::CommunicationCluster = NM_BUSNM_CANNM NM_COMM_NETWORK_HANDLE_REF = 0 NmStateChangeIndEnabled = TRUE NmPassiveModeEnabled = FALSE NmUserDataEnabled = TRUE NmNodeIdEnabled = TRUE NmNodeDetectionEnabled = TRUE NmPduRxIndicationEnabled = TRUE Callback function: Nm_PduRxIndication		
Summary	Using Nm_GetLocalNodeIdentifier to get the node identifier configured for local node.		
Needed Adaptation to other Releases			
Pre-conditions	DUT shall be initialized		
Main Test Execution			
Test Steps		**Pass Criteria**	
Step 1	[SWC] Invoke Rte_Call to request ComM in COMM_FULL_COMMUNICATION	[SWC] Rte_Call shall return RTE_E_OK.	
Step 2	[SWC] Send Nm_GetLocalNodeIdentifier request to get the node identifier from local node.	[SWC] Node identifier value shall be same as the configured Node Id (eg. 0x03).	
Post-conditions	NONE		

8.3.6 [ATS_COMM_00765] Successful Wakeup Validation For Wakeup On A Sleeping Bus When The DUT Is In RUN State

Test Objective	Successful Wakeup Validation For Wakeup On A Sleeping Bus When The DUT Is In RUN State		
ID	ATS_COMM_00765	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	EcuMfEcuMComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00893 ECUStateManager: SWS_EcuM_02963 ECUStateManagerFixed: SWS_EcuMf_00027		
Requirements / Reference to Test Environment	none		

Configuration Parameters	BswM Mode CAN channel 1 Wakeup on bus on EcuM	
Summary	Validate the Wakeup event on the bus When the DUT is in RUN state.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized Bus has to be in Sleep State.	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Invoke RTE in order to request Active mode from BswM	[SWC] RTE response shall return E_OK.
Step 2	[LT] Tester shall transmit a CAN frame on channel 1	[LT] Shall observe the Application messages with CAN ID on channel 1.
Step 3	[SWC] Invoke ComM_GetCurrentComMode to get the current mode of the Channel	[SWC] ComM_GetCurrentComMode shall return E_OK
Step 4	[SWC] Invoke a runnable to know about the current mode of EcuM	[SWC] EcuM shall be in APP RUN state. Application messages observed on the bus periodically.
Post-conditions	NONE	

8.3.7 [ATS_COMM_00766] Wakeup Inhibition By SWC When Channel Is In NO Communication Mode

Test Objective	Wakeup Inhibition By SWC When Channel Is In NO Communication Mode		
ID	ATS_COMM_00766	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00218		
Requirements / Reference to Test Environment	none		
Configuration Parameters	ComM/ComMGeneral – ComMWakeupInhibitionEnabledChannel 1EcuM		
Summary	When wake up inhibition is enabled on a channel any request for FULL communication for that channel shouldn't be accepted. Once the inhibition is disabled request for FULL communication for that channel should be accepted.		

Needed Adaptation to other Releases		
Pre-conditions	Channel is in No Communication mode	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Invoke ComM_PreventWakeUp with parameter channel and status as TRUE	[SWC] ComM_PreventWakeUp shall return E_OK Wake up Inhibition is switched on
Step 2	[SWC] Invoke ComM_GetInhibitionStatus to get the inhibition status of ComM channel	[SWC] ComM_GetInhibitionStatus shall return E_OK
Step 3	[SWC] Invoke ComM_RequestComMode to change the ComM mode to Full Communication mode.	[SWC] An error shall be notified saying Mode Limitation
Step 4	[SWC] Invoke ComM_GetCurrentComMode to get the current mode of the channel	[SWC] ComM_GetCurrentComMode shall return E_OK Com mode shall be in NO_COMMUNICATION
Step 5	-	[LT] Frames shall not be observed on the tester
Step 6	[SWC] Invoke ComM_PreventWakeUp with parameter channel and status as FALSE	[SWC] ComM_PreventWakeUp shall return E_OK Wake up inhibition is switched off
Step 7	[SWC] Invoke ComM_GetInhibitionStatus to get the inhibition status of ComM channel	[SWC] ComM_GetInhibitionStatus shall return E_OK Status shall be updated successfully
Step 8	[SWC] Invoke ComM_GetCurrentComMode to get the current mode of the channel	[SWC] ComM_GetCurrentComMode shall return E_OK.Com Mode shall be updated with Full Communication
Step 9	-	[LT] Application messages with CAN IDs shall be observed on CAN channel 1 periodically at every 10ms
Post-conditions	NONE	

8.3.8 [ATS_COMM_00767] Wakeup Inhibition By SWC When Channel Is In FULL Communication Mode

Test Objective	Wakeup Inhibition By SWC When Channel Is In FULL Communication Mode		
ID	ATS_COMM_00767	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed

Trace to Requirement on Acceptance Test Document		
Trace to SWS Item	COMManager: SWS_ComM_00219	
Requirements / Reference to Test Environment	none	
Configuration Parameters	BswM ComM/ComMGeneral – ComMWakeupInhibitionEnabled Channel 1EcuM	
Summary	Wakeup inhibition on a channel can be enabled only if the channel is in COMM_NO_COMMUNICATION mode. When a channel is COMM_FULL_COMMUNICATION mode any request for prevent wake up should not be accepted.	
Needed Adaptation to other Releases		
Pre-conditions	DUT shall be initialized	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[SWC] Invoke ComM_RequestComMode to change the ComM mode to Full Communication mode.	[SWC] Mode FULL_COMMUNICATION shall be updated
Step 2	[SWC] Invoke ComM_GetCurrentComMode to get the current mode of the channel	[SWC] Com Mode shall be in Full Communication
Step 3	-	[LT] Application messages with CAN IDs shall be observed on CAN channel 1 periodically at every 10ms
Step 4	[SWC] Invoke ComM_PreventWakeUp with parameter channel and status as TRUE	[SWC] Wake up status for the channel shall be changed successfully (Inhibition shall not be active if the channel is already active)
Step 5	[SWC] Invoke ComM_GetCurrentComMode to get the current mode of the channel	[SWC] Com Mode shall be updated with Full Communication
Step 6	-	[LT] Application messages with CAN IDs shall be observed on CAN channel 1 periodically at every 10ms
Post-conditions	NONE	

8.3.9 [ATS_COMM_00768] Wakeup Inhibition By SWC When Bus Activity Is Detected On The Channel

Test Objective	Wakeup Inhibition By SWC When Bus Activity Is Detected On The Channel

ID	ATS_COMM_00768	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00066		
Requirements / Reference to Test Environment	none		
Configuration Parameters	ComM Channel 0 : CAN		
Summary	Never inhibit Passive wake up. If a wake up inhibition is enabled on a channel and if a wake up event occurs on that channel it should be processed and the mode of the channel should be COMM_FULL_COMMUNICATION.		
Needed Adaptation to other Releases			
Pre-conditions	Channel 1 is in No communication state		
Main Test Execution			
Test Steps		**Pass Criteria**	
Step 1	[SWC] Invoke ComM_PreventWakeUp with parameter channel and status	[SWC] Wake up status for the channel shall be changed successfully	
Step 2	[SWC] Invoke ComM_RequestComMode to change the ComM mode to Full Communication mode.	[SWC] An error shall be notified saying Mode Limitation	
Step 3	[SWC] Invoke ComM_GetCurrentComMode to get the current mode of the channel	[SWC] Com Mode shall be updated with No Communication	
Step 4	-	[LT] Frames shall not be observed on the tester	
Step 5	[LT] Tester shall transmit application CAN frame with CAN ID from tester on channel 1	[LT] Application messages with CAN IDs shall be observed on CAN channel 1 periodically at every 10ms	
Step 6	[SWC] Invoke ComM_GetCurrentComMode to get the current mode of the channel	[LT] Com Mode shall be updated with Full Communication	
Post-conditions	NONE		

8.3.10 [ATS_COMM_00769] Wakeup Inhibition By SWC During Active Diagnostic Session

Test Objective	Wakeup Inhibition By SWC During Active Diagnostic Session

ID	ATS_COMM_00769	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00182		
Requirements / Reference to Test Environment	none		
Configuration Parameters	CanTP BswM Channel 1 EcuM ComM/ComMGeneral – ComMWakeupInhibitionEnabled		
Summary	Never inhibit Diagnostic frames. When a diagnostic frame is received on the channel which is enabled for wakeup inhibition it should be processed and the mode of the channel should be set to COMM_FULL_COMMUNICATION.		
Needed Adaptation to other Releases			
Pre-conditions	Channel 1 is in No communication state		
Main Test Execution			
Test Steps		**Pass Criteria**	
Step 1	[SWC] Invoke ComM_PreventWakeUp with parameter channel and status as TRUE	[SWC] Wake up status for the channel shall be changed successfully	
Step 2	[SWC] Invoke ComM_RequestComMode to change the ComM mode to Full Communication mode.	[SWC] An error shall be notified saying Mode Limitation	
Step 3	[SWC] Invoke ComM_GetCurrentComMode to get the current mode of the channel	[SWC] Com Mode shall be updated with No Communication	
Step 4	-	[LT] Frames shall not be observed on the tester	
Step 5	[LT] Tester shall transmit a diagnostic frame with CAN-ID DLC value as 8 bytes and data on CAN channel 1	[LT] The DUT shall send a response frame Application messages with CAN IDs shall be observed on CAN channel 1 periodically at every configured time	
Step 6	[SWC] Invoke ComM_GetCurrentComMode to get the current mode of the channel	[LT] Com Mode shall be updated with Full Communication (The communication inhibition shall get temporarily inactive during an active diagnostic session)	
Post-conditions	NONE		

8.3.11 [ATS_COMM_00811] Behavior of the ComM channel state machine when configuration parameter commNoCom=True

Test Objective	Behavior of the ComM channel state machine when configuration parameter commNoCom=True		
ID	ATS_COMM_00811	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00110		
Requirements / Reference to Test Environment	none		
Configuration Parameters	COMM_NM_VARIANT = FULL COMM_MODE_LIMITATION_E-BLED =TRUE COMM_NO_COM = TRUE		
Summary	Configure one user and map a channel to it. And limit the channel to No Communication. Send the request to change the COM MODE to Full communication but the channel will not change its mode and it remains in the No communication only.		
Needed Adaptation to other Releases			
Pre-conditions	DUT shall be initialized		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[SWC] Request ComM to be in Full Communication		[SWC] RTE shall return E_OK
Step 2	[SWC] Request ComM for the current mode		[SWC] ComM mode shall be in Full communication
Step 3	[SWC] request ComM to limit channel 0 to No communication		[SWC] Successfully changed to the No Communication mode
Step 4	[SWC] Request ComM to be in Full Communication		[SWC] Failed to change the mode and RTE returns COMM_E_MODE_LIMITATION
Step 5	[SWC] Request ComM for the current mode		[SWC] ComM shall be in No communication mode
Step 6	-		[SWC] DUT shall not send any NM Frames on the bus
Post-conditions	NONE		

8.3.12 [ATS_COMM_00815] Transmission And Reception On The Affected Physical Channels For SILENT_COMMUNICATION

Test Objective	Transmission And Reception On The Affected Physical Channels For SILENT_COMMUNICATION		
ID	ATS_COMM_00815	**AUTOSAR Releases**	4.0.3|4.2.1|4.2.2
Affected Modules	ComM	**State**	reviewed
Trace to Requirement on Acceptance Test Document			
Trace to SWS Item	COMManager: SWS_ComM_00071		
Requirements / Reference to Test Environment	none		
Configuration Parameters	COMM_NM_VARIANT = FULL		
Summary	Bring the Integrated Stack to Full communication and requests for No communication. A callout is created to indicate silent communication. Check the configured ComM channels.		
Needed Adaptation to other Releases			
Pre-conditions	DUT shall be initialized		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[SWC] Send the request for changing COM mode to Full communication for user0		[SWC] RTE shall return E_OK
Step 2	[SWC] Send the request for the current mode of ComM for user0		[SWC] ComM mode shall be in Full communication
Step 3	[SWC] Send the request for changing COM mode to Full communication for user0		[SWC] RTE shall return E_OK BswM user callout shall be invoked for ComM current state notification indicating Silent communication
Step 4	[SWC] Get the NM mode and state		[SWC] NM mode shall be in “PREPARE BUS SLEEP” and NM state shall be “PREPARE BUS SLEEP
Post-conditions	NONE		