



Elektrobit

EB tresos[®] AutoCore Generic 8 CAN Stack documentation

release notes update for the CanNm module

product release 8.8.7



Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen, Germany
Phone: +49 9131 7701 0
Fax: +49 9131 7701 6333
Email: info.automotive@elektrobit.com

Technical support

<https://www.elektrobit.com/support>

Legal disclaimer

Confidential information.

ALL RIGHTS RESERVED. No part of this publication may be copied in any form, by photocopy, microfilm, retrieval system, or by any other means now known or hereafter invented without the prior written permission of Elektrobit Automotive GmbH.

All brand names, trademarks, and registered trademarks are property of their rightful owners and are used only for description.

Copyright 2023, Elektrobit Automotive GmbH.



Table of Contents

- 1. Overview 4
- 2. CanNm module release notes 5
 - 2.1. Change log 5
 - 2.2. New features 15
 - 2.3. Elektrobit-specific enhancements 15
 - 2.4. Deviations 16
 - 2.5. Limitations 29
 - 2.6. Open-source software 29

1. Overview

This document provides you with the release notes to accompany an update to the `CanNm` module. Refer to the changelog [Section 2.1, “Change log”](#) for details of changes made for this update.

Release notes details

- ▶ EB tresos AutoCore release version: 8.8.7
- ▶ EB tresos Studio release version: 29.2.1
- ▶ AUTOSAR R4.0 Rev 3
- ▶ Build number: B614323

2. CanNm module release notes

- ▶ AUTOSAR R4.0 Rev 3
- ▶ AUTOSAR SWS document version: 3.3.0
- ▶ Module version: 6.20.3.B614323
- ▶ Supplier: Elektrobit Automotive GmbH

2.1. Change log

This chapter lists the changes between different versions.

Module version 6.20.3

2023-02-22

- ▶ ASCCANNM-1328 Fixed known issue: Impossible to use tresos ImportEcuConfig Unattended Wizard to import a CanNm module configuration.
- ▶ ASCCANNM-1339 Fixed known issue: If support for distribution of CAN stack along network boundaries is enabled, then CanNm module might not compile.
- ▶ ASCCANNM-1342 Fixed known issue: CanNm module does not compile if partial networking is used and CanNmPnEiraCalcEnabled is set to FALSE.

Module version 6.20.2

2022-10-12

- ▶ ASCCANNM-1306 Updated requirement Id format in module documentation and source code tracing comments. Note: This does not change the Baseline, nor functionality.

Module version 6.20.1

2022-07-04

- ▶ Increase CanNm support for 504 PNCs.
- ▶ Implemented Support for Synchronized PNC shutdown.
- ▶ ASCCANNM-1308 Fixed known issue: CanNm reports Det errors with incorrect error ID when an invalid PDU-ID is used.

Module version 6.20.0

2022-03-09

- ▶ Added support for distribution of CAN stack along network boundaries

Module version 6.19.8

2021-10-27

- ▶ Improved support for calling CanNm APIs that rely on the current state from the context of the state change notification

Module version 6.19.7

2021-06-25

- ▶ ASCCANNM-1236 Fixed known issue: CanNm might unexpectedly not transmit NM messages after (re)entering Normal Operation State or Repeat Message State.

Module version 6.19.6

2021-03-05

- ▶ Added support for postbuild selectable config of CanNmMsgCycleOffset

Module version 6.19.5

2020-10-23

- ▶ Improved Active wakeup Bit functionality

Module version 6.19.4

2020-06-19

- ▶ ASCCANNM-1194 Fixed known issue: First NM message sent on the bus carries outdated state change information and additional messages are sent on the same main function.

Module version 6.19.3

2020-02-21

- ▶ Internal module improvement. This module version update does not affect module functionality.

Module version 6.19.2

2019-10-11

- ▶ ASCCANNM-1142 Fixed known issue: Compilation error for CanNm module if CanNmStateChangeIndEnabled is set to true
- ▶ Changed maximum value for CanNmPnInfoOffset to 31 and minimum value for CanNmPnInfoLength to 1
- ▶ ASCCANNM-1160 Fixed known issue: Linker errors are reported due to incorrect memory mapping
- ▶ ASCCANNM-1159 Fixed known issue: Active wake-up bit in CBV is incorrectly set when Network Mode is re-entered.

Module version 6.19.1

2019-06-14

- ▶ Support transition from PrepareBusSleepMode to NetworkMode when CanNm_PassiveStartUp is triggered

Module version 6.19.0

2019-02-15

- ▶ Changed usage of message cycle offset
- ▶ Improved robustness check for references, optional parameters property and enable parameters property
- ▶ ASCCANNM-1107 Fixed known issue: First NM message sent on the bus carries outdated state change information
- ▶ ASCCANNM-1116 Fixed known issue: State change notification can be sent repeatedly from RepeatMessage state
- ▶ Changed structure of generated templates for post-build selectable support

Module version 6.18.0

2018-10-26

- ▶ Implemented Multi-core support
- ▶ ASCCANNM-1096 Fixed known issue: CanNm generates an invalid basic software module description if no configuration set is provided

Module version 6.17.3

2018-06-22

- ▶ Internal module improvement. This module version update does not affect module functionality.

Module version 6.17.2

2018-02-16

- ▶ Implemented Post-build selectable support
- ▶ Removed AUTOSAR 3.x compliant symbolic name value macros and updated the logic to only provide AUTOSAR 4.0.2 compliant macros

Module version 6.17.1

2017-09-22

- ▶ Implemented support for car wake up
- ▶ Changed number of calls for Nm_StateChangeNotification during Prepare Bus Sleep Mode to Repeat Message State. Reverted ASCCANNM-923 Wrong state change information sent on CanBus on transition from Prepare Bus Sleep Mode to Repeat Message State
- ▶ ASCCANNM-943 Fixed known issue: Compilation error occurs if all CanNmPnFilterMaskByteValues are set to zero
- ▶ Added CanNmNodeDetectionEnabled, CanNmNodeIdEnabled and CanNmRepeatMsgIndEnabled as per channel configurable.
- ▶ ASCCANNM-944 Fixed known issue: Out of bounds access in case at post-build more PNCs are configured than at precompile time
- ▶ Implemented Support for PDU Length greater than 8 Bytes
- ▶ ASCCANNM-959 Fixed known issue: Source code does not compile if CanNmPnEraCalcEnabled is set to false and for at least one channel CanNmPnEraCalcEnabled is set to true
- ▶ ASCCANNM-972 Fixed known issue: Code generation error for user data with length zero
- ▶ ASCCANNM-973 Fixed known issue: API CanNm_CheckRemoteSleepIndication indicates wrong state if the API interrupts the execution of the main function
- ▶ ASCCANNM-961-973 Fixed known issue: Existence of PDU referenced by CanNmPnEraRxNSduRef in PduR is not checked
- ▶ ASCCANNM-960 Fixed known issue: CanNmPnEraRxNSduRef shall be available based on parameter CanNmPnEraCalcEnabled

- ▶ ASCCANNM-963 Fixed known issue: Tx timeout exception is generated for a channel which works correctly

Module version 6.17.0

2017-03-31

- ▶ ASCCANNM-900 Fixed known issue: CanNm causes the ECU to become asynchronous with other ECUs on the network
- ▶ Added/corrected missing memory sections and compiler abstraction
- ▶ ASCCANNM-885 Fixed known issue: Incorrect consistency check of CanNmComUserDataSupport against CanNmUserDataEnabled
- ▶ ASCCANNM-881 Fixed known issue: The user data transmitted in the NM PDU could be inconsistent
- ▶ Removed CanNmRepeatMessageTime - CanNmMsgCycleTime multiplicity constrain
- ▶ Changed UserTxConfPduId member in the CanNm_ChannelDataType structure
- ▶ ASCCANNM-915 Fixed known issue: Wrong dependency CanNmStateChangeIndEnabled - CanNmPassiveModeEnabled. Improved CanNmStateChangeIndEnabled parameter description.
- ▶ ASCCANNM-923 Fixed known issue: Wrong state change information sent on CanBus on transition from Prepare Bus Sleep Mode to Repeat Message State

Module version 6.16.0

2016-10-31

- ▶ Added aggregation of internal and external requested partial networks also if the NM-PDU filter algorithm is disabled
- ▶ Implement CanNm_Transmit(...) for sponaneous transmission of NM frames
- ▶ ASCCANNM-870 Fixed known issue: Tx timeout timer called incorrectly

Module version 6.15.0

2016-05-23

- ▶ ASCCANNM-871 Fixed known issue: Repeat Message time is not editable if Passive mode is enabled

Module version 6.14.0

2016-02-10

- ▶ ASCCANNM-842 Fixed known issue: Handle ID wizard error computing CanNmRxPduIds when multiple CanNmRxPdu per channel are configured
- ▶ Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeader-File`

Module version 6.13.0

2015-11-06

- ▶ Fixed In case RMS timer is zero parameter `NmRetryFirstMessageRequest` should not be active
- ▶ Fixed incorrect handling of RSI timer in case of disabled communication
- ▶ Implemented `NmRetryFirstMessageRequest` functionality to Normal operation state if Repeat Message Time is zero
- ▶ Fixed the NM-Timeout and ReducedTime timer restart if Communication is disabled

Module version 6.12.0

2015-07-28

- ▶ ASCCANNM-807 Fixed known issue: Generation error when `CanNmNodeIdEnabled = TRUE` and `CanNmPassiveModeEnabled = TRUE`
- ▶ ASCCANNM-818 Fixed known issue: Changed implementation class of `CanNmNodeId` to `PostBuild`
- ▶ ASCCANNM-780 Fixed known issue: RMS time can be 1 cycle shorter if Immediate transmission is set to true
- ▶ ASCCANNM-812 Fixed known issue: Unintended trigger of immediate messages when `CanNmImmediateNmTransmissions > 0` but `CanNmImmediateNmCycleTime` is disabled.
- ▶ ASCCANNM-798 Fixed known issue: In case of passive mode `CanNmUserDataTxPdu` should not be forced to exist

Module version 6.11.0

2015-06-24

- ▶ ASCCANNM-771 Fixed known issue: The Nm Message Tx Timeout Timer is not started when entering Network Mode from Prepare Bus Sleep Mode
- ▶ ASCCANNM-770 Fixed known issue: The `Nm_RemoteSleepIndication()` can be triggered by a transition from Ready Sleep to Normal Operation
- ▶ ASCCANNM-751 Fixed known issue: TX Pdu is still transmitted after API `CanNm_DisableCommunication` is called

- ▶ ASCCANNM-765 Fixed known issue: CanNm_CheckRemoteSleepIndication has an incorrect return value

Module version 6.10.0

2015-02-20

- ▶ ASCCANNM-776 Fixed known issue: Statemachine remains in RepeatMessageState if CanNmRepeatMessageTime is 0
- ▶ CanNm_Checks.m misleading error message

Module version 6.9.0

2015-01-07

- ▶ Fixed MISRA warning
- ▶ Update multiplicity of CanNmRxPdu elements according to resolution in bugzilla 54555.
- ▶ Add the CanNmNodeIdEnabled parameter.

Module version 6.8.0

2014-10-02

- ▶ Removed dependency that if CanNmRemoteSleepIndEnabled is false, then CanNmRemoteSleepIndTime needs to be 0
- ▶ ASCCANNM-686 Fixed known issue: Incomplete initialization of EIRA timer array
- ▶ ASCCANNM-674 Fixed known issue: NM messages are lost in case of external wake-ups
- ▶ Loosen the dependency between parameters CanNmRemoteSleepIndTime and CanNmMsgCycleTime
- ▶ Implemented support for Side Allocation
- ▶ Refactor CanNm.h to CanNm_Api.h
- ▶ Add a test to prove that the module compiles without any post build information
- ▶ ASCCANNM-744 Fixed known issue: CanNmRemoteSleepIndTime cannot be 0 even if CanNmRemoteSleepIndEnabled is false
- ▶ Implement support for the aggregation of external requested partial networks (ERA)

Module version 6.7.0

2014-04-25

- ▶ ASCCANNM-648 Fixed known issue: Wrong memory section for an array of type `NetworkHandleType`
- ▶ ASCCANNM-663 Fixed known issue: `PreCompile` parameters referencing `Postbuild` parameters in invalid conditions in CanNm's XDM schema
- ▶ ASCCANNM-670 Fixed known issue: ECU may be prevented from entering SLEEP mode due to incorrect handling of EIRA
- ▶ ASCCANNM-675 Fixed known issue: Compile error in `CanNm_HandleTimerTick` function if Debug and Trace module is enabled
- ▶ ASCCANNM-668 Fixed known issue: In case of active wake-ups the last immediate and first cyclic message shall not be transmitted in the same cycle if `CanNmMsgCycleOffset` is configured to 0

Module version 6.6.0

2013-12-11

- ▶ ASCCANNM-624 Fixed known issue: TX timeout exception is not reported if `CanIf_Transmit()` returns `E_NOT_OK`
- ▶ ASCCANNM-623 Fixed known issue: User data is initialized incorrectly to 0x00U instead of 0xFFU if partial networking is enabled
- ▶ ASCCANNM-608 Fixed known issue: An incorrect DET error is reported when an invalid PDU ID is passed to `CanNm_TxConfirmation()` or `CanNm_RxIndication()`
- ▶ Implemented support of `VARIANT-POST-BUILD` for CanNm

Module version 6.5.0

2013-10-18

- ▶ ASCCANNM-587 Fixed known issue: NM Messages are stopped after the transmission of immediate NM messages
- ▶ ASCCANNM-586 Fixed known issue: Unexpected behavior when `CanNmPduCbvPosition` is configured to `CANNM_PDU_OFF`
- ▶ Implemented support of spontaneous transmission of NM PDUs via calls of API function `CanNm_NetworkRequest()`
- ▶ Changed functionality to release a network even if transmission of NM messages is disabled
- ▶ Implemented support for notification of transmission timeouts to CanSM
- ▶ ASCCANNM-609 Fixed known issue: CanNm erroneously reports TX timeout exception if feature *Bus Load Reduction* is enabled
- ▶ Changed the default value of parameter `CanNmRemoteSleepIndEnabled` to false

- ▶ Added support for function tracing via AUTOSAR Debugging

Module version 6.4.0

2013-06-25

- ▶ Improved the robustness of the finite state machine design by revising the event handling; removed configuration parameter `CanNmEventQueueSize`
- ▶ ASCCANNM-548 Fixed known issue: A compiler errors when `CanNmNodeDetectionEnabled` is set to false and '`CanNmPduNidPosition` is not set to `CANNM_PDU_OFF`'
- ▶ ASCCANNM-439 Fixed known issue: CanNm does not switch to repeat message state if communication is disabled

Module version 6.3.0

2013-02-15

- ▶ ASCCANNM-504 Fixed known issue: Error occurs during code generation when `CanNmTxPd` is disabled and `CanNmPassiveMode` is set to false
- ▶ Changed the reference path of `ComMChannel` in parameter `CanNmComMNetworkHandleRef` to `/AUTOSAR/EcuDefs/ComM/ComMConfigSet/ComMChannel`
- ▶ Memory allocation keywords were implemented in compliance to ASR 4.0.3
- ▶ ASCCANNM-474 Fixed known issue: The API functions `CanNm_GetNodeIdentifier()` / `CanNm_GetLocalNodeIdentifier()` are also available when parameter `CanNmPduNidPosition` is set to off
- ▶ ASCCANNM-507 Fixed known issue: Compiler errors when symbolic names according to AR4.0.3 are used
- ▶ ASCCANNM-512 Fixed known issue: `PduR_CanNmTxConfirmation()` and `PduR_CanNmTriggerTransmit()` are called with the wrong handle ID

Module version 6.2.0

2012-10-12

- ▶ ASCCANNM-463 Fixed known issue: Immediate Nm messages are not sent when `CanNmComControlEnabled` is set to false
- ▶ ASCCANNM-464 Fixed known issue: Incorrect number of immediate Nm messages
- ▶ ASCCANNM-461 Fixed known issue: Extra NM message is sent while leaving Repeat Message State when `CanNmMsgCycleOffset` is zero

- ▶ ASCCANNM-465 Fixed known issue: Compiler warning `statement not reached`
- ▶ ASCCANNM-467 Fixed known issue: CanNm schema does not prevent an invalid configuration of `CanNmRepeatMessageTime`
- ▶ Migrated to ASR 4.0 ComStack HandleId Policy
- ▶ Support for `ActiveWakeUp` bit in CBV added
- ▶ The top-level structure of the software-component description in the ARXML files changed from `/AUTOSAR/CanNm` to `/AUTOSAR_CanNm`
- ▶ ASCCANNM-486 Fixed known issue: Error during the PduR code generation when Com User Data support is enabled

Module version 6.1.1

2012-06-27

- ▶ Update to AUTOSAR 4.0.3 version
- ▶ ASCCANNM-413 Fixed known issue: No configuration constant for `CanNm_Init` is available
- ▶ ASCCANNM-422 Fixed known issue: CanNm may ignore valid PN messages
- ▶ ASCCANNM-397 Fixed known issue: Transition from Ready Sleep State to Prepare Bus-Sleep Mode takes longer than `CanNmTimeoutTime`
- ▶ ASCCANNM-425 Fixed known issue: EIRA contains PN requests which are not relevant for the ECU
- ▶ Corrected the invalid MemMap usage in `CanNm_HsmCanNmFunct.c`

Module version 6.1.0

2012-03-15

- ▶ ASCCANNM-364 Fixed known issue: Node never goes to Sleep state if `CanNm_DisableCommunication` is called in Repeat message state
- ▶ ASCCANNM-339 Fixed known issue: Event queue overflow
- ▶ ASCCANNM-385 Fixed known issue: Production error `CANNM_E_NETWORK_TIMEOUT` is erroneously reported

Module version 6.0.2

2011-12-08

- ▶ COM User Data Support added

Module version 6.0.1

2011-09-28

- ▶ EBACANNM-219 Fixed known issue: Bus Load Reduction Mechanism might increase the bus load

Module version 6.0.0

2011-09-02

- ▶ Initial AUTOSAR 4.0 version

2.2. New features

- ▶ No new features have been added since the last release.

2.3. Elektrobit-specific enhancements

This chapter lists the enhancements provided by the module.

- ▶ COM Rx user data

Description:

COM Support for Rx user data is added.

- ▶ New container `CanNmUserDataRxPdu` is added to configure the Rx Pdu of received user data.
- ▶ User can enable or disable this container.

When this feature is enabled, then user must configure the respective Pdu in EcuC and provide correct routing path in PduR. When this feature is disabled, the user can still receive data using `CanNm_GetUserData()` API.

Rationale:

User has freedom of receiving the user data over COM.

- ▶ Channel with no user data with user data support enabled

Description:

As per AUTOSAR requirement CANNM086, when `CanNmUserDataEnabled` is enabled, the `CanNmUserDataLength` should not be zero.

The module deviates from this requirement. The module allows a user to configure a mixture of channels where some channels support user data and some channels doesn't support user data.

If CanNm_GetUserData or CanNm_SetUserData API is called for a channel with user data length as 0, DET error CANNM_E_INVALID_FUNCTION_ARG will be registered.

Rationale:

More flexibility and freedom of configuration for user is achieved.

► Support for Side Allocation

Description:

The Side Allocation feature allow flashing of two different ECUs with the same software. The behaviour of each ECU will differ at runtime based on a flag(eg: stored in EEPROM or the level of a pin).

The following parameter differ between the two variants: `CanNmNodeId` CanNm supports configuring a callout function to be called everytime the CanNm module needs to retrieve a NodeId for an ECU.

► Extended the number of PNCs from 56 to 504

Description:

Starting from AUTOSAR R20-11 requirements ECUC_CanNm_00060 and ECUC_CanNm_00061, CanNmPnInfoLength range has changed from 1..7 to 1..63 and CanNmPnInfoOffset range has changed from 1..31 to 1..63.

Rationale:

More flexibility and freedom of configuration for user is achieved.

2.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

► COM user data zero length

Description:

As per the requirement CANNM086 / SWS_CanNm_00086, if `CANNM_USER_DATA_ENABLED` is enabled, `CANNM_USER_DATA_LENGTH` should not be zero. In contrast to this, the user is allowed to configure a channel with `CANNM_USER_DATA_LENGTH` as zero.

Rationale:

In reality we may have a use case, where in a CanNm channel user data is present and another channel is not having user data. This requirement puts restriction on such use case.

Requirements:

CANNM086, SWS_CanNm_00086

- Changes in symbolic name references

Description:

If the attribute `SHORT-NAME` is not specified for the container `CanNmRxPdu` the symbolic name macros for `CanNmRxPduId` are generated not according to the requirement `ecuc_sws_2108` but according to the naming pattern `CanNmConf_CanNmChannelConfig_<ChannelName>_CanNmRxPdu`. where `<ChannelName>` is the name of the channel containing `CanNmRxPdu`.

The above behavior is also applicable for the generation of symbolic name for the parameter `CanNmTx-ConfirmationPduId` which is located inside the container `CanNmTxPdu` and `CanNmTxUserDataPduId` which is located inside the container `CanNmUserDataTxPdu` whose symbolic name macros are generated in the following pattern: `CanNmConf_CanNmChannelConfig_<ChannelName>_CanNmTxPdu` and `CanNmConf_CanNmChannelConfig_<ChannelName>_CanNmUserDataTxPdu`.

Rationale:

If no short-name is specified, EB tresos Studio assumes the name of the corresponding schema node as a default. Thus, the symbolic name macros generated according to the requirement `ecuc_sws_2108` are not unique.

- No support for link-time configuration parameters

Description:

The following parameters are treated as pre-compile time parameters instead of as link-time parameters:

- `CanNmGlobalConfig/CanNmMainFunctionPeriod`
- `CanNmGlobalConfig/CanNmPnResetTime`
- `CanNmGlobalConfig/CanNmPnEiraRxNSduRef`
- `CanNmGlobalConfig/CanNmChannelConfig/CanNmAllNmMessagesKeepAwake`
- `CanNmGlobalConfig/CanNmChannelConfig/CanNmBusLoadReductionActive`
- `CanNmGlobalConfig/CanNmChannelConfig/CanNmCarWakeUpBitPosition`
- `CanNmGlobalConfig/CanNmChannelConfig/CanNmCarWakeUpBytePosition`
- `CanNmGlobalConfig/CanNmChannelConfig/CanNmCarWakeUpFilterEnabled`
- `CanNmGlobalConfig/CanNmChannelConfig/CanNmCarWakeUpFilterNodeId`
- `CanNmGlobalConfig/CanNmChannelConfig/CanNmCarWakeUpRxEnabled`

- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmImmediateNmCycleTime
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmImmediateNmTransmissions
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmMsgCycleTime
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmMsgReducedTime
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmMsgTimeoutTime
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmNodeId
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmPduCbvPosition
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmPduNidPosition
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmPnEraCalcEnabled
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmPnHandleMultipleNetworkRequests
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmRemoteSleepIndTime
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmRepeatMessageTime
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmTimeoutTime
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmUserDataLength
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmWaitBusSleepTime
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmPnEraRxNSduRef
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmComMNetworkHandleRef
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmRxPdu/CanNmRxPduId
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmTxPdu/CanNmTxConfirmationPduId
- ▶ CanNmGlobalConfig/CanNmChannelConfig/CanNmUserDataTxPdu/CanNmTxUserDataPduId
- ▶ CanNmGlobalConfig/CanNmPnInfo/CanNmPnInfoLength
- ▶ CanNmGlobalConfig/CanNmPnInfo/CanNmPnInfoOffset
- ▶ CanNmGlobalConfig/CanNmPnInfo/CanNmPnFilterMaskByte/CanNmPnFilterMaskByteIndex

Requirements:

CANNM300

- ▶ Changes in symbolic name references

Description:

If the attribute `SHORT-NAME` is specified for the container `CanNmRxPdu` the symbolic name macros for `CanNmRxPduId` are generated not according to the requirement `ecuc_sws_2108` but according to the

naming pattern `CanNmConf_CanNmChannelConfig_<ChannelName>_SHORT-NAME`. where `<ChannelName>` is the name of the channel containing `CanNmRxPdu`.

► Changes regarding `CanNmImmediateNmCycleTime`

Description:

Parameter `CanNmImmediateNmCycleTime` has a default value of 0.001 and the multiplicity is 1 and not 0..1 as in the Autosar 4.0.3 SWS and Autosar R20-11.

Requirements:

CANNM057_Conf, ECUC_CanNm_00057

► Changes regarding `CanNmMsgCycleOffset`

Description:

Parameter `CanNmMsgCycleOffset` is treated as post-build selectable parameter instead of as link-time parameter.

Requirements:

CANNM300, CANNM029_Conf, ECUC_CanNm_00029

► Changes regarding `CanNmPnFilterMaskByteIndex`

Description:

Parameter `CanNmPnFilterMaskByteIndex` can only be configured with a value between 0 and 62 since it represents the position within the mask byte arrays it shall always be smaller than `CanNmPnInfoLength` which has the range 1-63.

Requirements:

ECUC_CanNm_00063

► Configuration parameters post build multiplicity

Description:

The following configuration items do not support postbuild multiplicity true: `CanNmChannelConfig`, `CanNmMsgTimeoutTime`.

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

ECUC_CanNm_00030, ECUC_CanNm_00017

► CanNmWaitBusSleepTime multiplicity

Description:

Configuration parameter `CanNmWaitBusSleepTime` is mandatory in the current implementation.

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

ECUC_CanNm_00021

► Coordinator sync support

Description:

Coordination of nested sub-busses is not supported. `Nm_CoordReadyToSleepIndication()` is never called and the API `CanNm_SetSleepReadyBit()` is not provided.

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

SWS_CanNm_00338, SWS_CanNm_00340, SWS_CanNm_00342, SWS_CanNm_00341, ECUC_CanNm_00080, SWS_CanNm_00348

► Partial network learning

Description:

Partial network learning algorithm is not supported in the current implementation.

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

SWS_CanNm_00380, ECUC_CanNm_00094, SWS_CanNm_00382, SWS_CanNm_91004, SWS_CanNm_00381, SWS_CanNm_00014, ECUC_CanNm_00093

► Dependency on `PnSupport` when calling `Nm_TxTimeoutException`

Description:

In the current implementation there is no dependency on `PnSupport` when calling the `Nm_TxTimeoutException`.

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

SWS_CanNm_00064, SWS_CanNm_00066

- Unsupported config parameters

Description:

Following configuration parameters are not supported by current implementation: `CanNmGlobalPnSupport`.

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

ECUC_CanNm_00086, SWS_CanNm_00065, SWS_CanNm_00330, SWS_CanNm_00446, SWS_CanNm_00346

- `CanNmStayInPbsEnabled` not supported

Description:

`CanNmStayInPbsEnabled` and related functionality is not supported by the current implementation.

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

SWS_CanNm_00088, SWS_CanNm_00115, ECUC_CanNm_00092

- Reliable TX confirmation

Description:

Handling of (reliable) Tx Confirmation is not implemented according to Autosar 20-11.

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

SWS_CanNm_00228, SWS_CanNm_00099

► First Nm message after enable communication

Description:

When `CanNm_EnableCommunication` is called, `MsgCycleOffset` is loaded in order to transmit first Nm message and transmission may not be started latest within the next main processing function.

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

SWS_CanNm_00178

► `CanNm_SetUserData` API reentrancy

Description:

Following APIs are specified as reentrant for different NM channels: `CanNm_SetUserData`. The APIs are provided as non-reentrant.

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

SWS_CanNm_00217

► `MsgCycleOffset` when transitioning from `ReadySleep` to `NormalOperation`

Description:

When `NormalOperation` is entered from `ReadySleep` state, the transmission of NM PDU is not started immediately. `MsgCycleOffset` is applied in this case.

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

SWS_CanNm_00006

► `PduR_CanNmTriggerTransmit` return value

Description:

Current implementation ignores returned value of `PduR_CanNmTriggerTransmit`.

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

SWS_CanNm_00450

- ▶ Autosar R21-11 Synchronized PNC Shutdown requirements

Description:

SynchronizedPncShutdown functionality is implemented according to requirements from Autosar R21-11.

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

SWS_CanNm_00506, SWS_CanNm_91003, ECUC_CanNm_00097, SWS_CanNm_00504, ECUC_CanNm_00096, ECUC_CanNm_00098, SWS_CanNm_00509, SWS_CanNm_00465, SWS_CanNm_00464, SWS_CanNm_00462, SWS_CanNm_00467, SWS_CanNm_00461, SWS_CanNm_00472, SWS_CanNm_00460, SWS_CanNm_00471, SWS_CanNm_00470

- ▶ Obsolete Synchronized PNC Shutdown requirements

Description:

Requirement is obsolete in Autosar R21-11. Implementation of PN sync shutdown was done according to Autosar R21-11.

Requirements:

SWS_CanNm_00463, SWS_CanNm_00466

- ▶ CBV Byte location

Description:

Default value was removed in Autosar R20-11, but default value is still set in implementation as implemented according to Autosar 4.0.3.

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

SWS_CanNm_00075

► Header filename changed

Description:

Header file name is changed. Generally all APIs shall be aligned with the header files where they are provided (this is specified for each API in Autosar R20-11).

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

SWS_CanNm_00307

► Missing API `Nm_CoordReadyToSleepCancellation`

Description:

`Nm_CoordReadyToSleepCancellation` API is not implemented.

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

SWS_CanNm_00325

► DET error when receiving a NM PDU in state `BusSleep` is done only if `CANNM_DEV_ERROR_DETECT` is TRUE.

Description:

Implementation is done according to requirement CANNM336 from Autosar 4.0.3 that differs from SWS_CanNm_00336 from R20.11 by the statement that DET error is reported only if `CANNM_DEV_ERROR_DETECT` is set to TRUE.

Rationale:

Reporting of DET errors should be guarded by `CANNM_DEV_ERROR_DETECT` set to TRUE.

Requirements:

SWS_CanNm_00336

► PN message filtering is disabled when entering `BusSleep`

Description:

Implementation is done according to requirement CANNM403 from Autosar 4.0.3 that differs from SWS_-CanNm_00403 from R20.11 by the statement that PN message filtering is disabled also when entering `BusSleep`

Rationale:

Implementation started with Autosar 4.0.3 specifications.

Requirements:

SWS_CanNm_00403

- ▶ `CanNmMainFunctionPeriod` range is from 0.001 to 0.255

Description:

Implementation is done according to requirement CANNM032_Conf from Autosar 4.0.3 where the range differs from ECUC_CanNm_00032 from R20.11

Rationale:

Implementation started with Autosar 4.0.3 specifications.

Requirements:

ECUC_CanNm_00032

- ▶ `CanNmPnHandleMultipleNetworkRequests` dependency

Description:

ECUC_CanNm_00073 requirement from Autosar R20-11 states that `CanNmPnHandleMultipleNetworkRequests` dependency is that `CanNmGlobalPnSupport` is `TRUE`, but implementation dependency is that `CanNmImmediateNmTransmissions` is 0, and `CanNmPnEnabled` is `TRUE`.

Requirements:

ECUC_CanNm_00073

Rationale:

Implementation started with Autosar 4.0.3 specifications. `CanNmGlobalPnSupport` is not supported.

- ▶ Transmission on entering `RepeatMessage` state

Description:

Requirement SWS_CanNm_00100 states that (re)transmission is started when entering `RepeatMessage` state, while requirement CANNM100 from Autosar 4.0.3 lists the states from which `RepeatMessage` state

was entered for the (re)transmission to occur. Implementation is done according to CANNM100 from Autosar 4.0.3.

Rationale:

Implementation started with Autosar 4.0.3 specifications.

Requirements:

SWS_CanNm_00100

- Configuration parameters contained in `CanNmGlobalConfig`

Description:

In contrast to `ECUC_CanNm_00001`, the implementation for `CanNmGlobalConfig` is missing the following parameters: `CanNmDynamicPncToChannelMappingSupport`, `CanNmGlobalPnSupport`. In contrast to `ECUC_CanNm_00001`, the implementation for `CanNmGlobalConfig` has the following additional parameters: `CanNmActiveWakeupBitEnabled`, `CanNmNodeIdCallback`, `CanNmNodeIdCallbackHeader`, `CanNmNumberOfChannels`, `CanNmPostBuildRamSize`, `CanNmPnShutdownMessageRetransmissionDuration`.

Rationale:

This is due to implementation starting with Autosar 4.0.3 and adding additional features from other releases and due to EB added requirements.

Requirements:

ECUC_CanNm_00001

- `CanNmActiveWakeupBitEnabled` configuration parameter parent container

Description:

In contrast to `ECUC_CanNm_00084`, the implemented parent container is `CanNmGlobalConfig` instead of `CanNmChannelConfig`.

Rationale:

The parameter was added with implementation of Autosar 4.0.3 but the schema file then didn't provide the parameter.

Requirements:

ECUC_CanNm_00084, ECUC_CanNm_00001

- UnInit development error name

Description:

`CANNM_E_NO_INIT` error define is used instead of `CANNM_E_UNINIT`.

Rationale:

The error name was added with implementation of Autosar 4.0.3 and backwards compatibility must be kept.

Requirements:

SWS_CanNm_00316

- Pointer parameter development error name

Description:

`CANNM_E_NULL_POINTER` error define is used instead of `CANNM_E_PARAM_POINTER`.

Rationale:

The error name was added with implementation of Autosar 4.0.3 and backwards compatibility must be kept.

Requirements:

SWS_CanNm_00316

- NotInBusSleep development error name

Description:

`CANNM_E_NOT_IN_BUS_SLEEP` error not implemented.

Rationale:

The error was not present in base implementation of Autosar 4.0.3.

Requirements:

SWS_CanNm_00316

- Nm message Tx Timeout timer stopping dependency

Description:

Stopping a Nm Message Tx Timeout Timer is not dependent of `CanNmGlobalPnSupport` configuration.

Rationale:

Implementation started from Autosar 4.0.3 and `CanNmGlobalPnSupport` is not supported.

Requirements:

SWS_CanNm_00061

► CanNmMsgTimeoutTime dependency and multiplicity

Description:

CanNmMsgTimeoutTime only dependency is that CanNmPassiveModeEnabled is set to FALSE, and multiplicity is 1.

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

ECUC_CanNm_00030

► CanNmRemoteSleepIndTime dependency and multiplicity

Description:

In current implementation, CanNmRemoteSleepIndTime is not dependent on CanNmRemoteSleepIndEnabled, and multiplicity is 1.

Rationale:

Implementation is inherited from 4.1.3 implementation.

Requirements:

ECUC_CanNm_00023

► Partly implemented functionalities

Description:

The listed changed requirements are partly not supported and can be requested on demand.

Requirements:

ECUC_CanNm_00029

► Unsupported functionalities

Description:

The listed requirements are currently not support and can be requested on demand.

Requirements:

SWS_CanNm_00384, SWS_CanNm_00350

► Configuration parameters inherited from Autosar 4.0.3

Description:

Following configuration parameters are still present in the CanNm configuration even if they have been removed from the specification: `CanNmNumberOfChannels`, `CanNmUserDataLength`.

Rationale:

Implementation is inherited from Autosar 4.0.3 implementation.

Requirements:

CANNM014_Conf, CANNM027_Conf

2.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- For this module no limitations are known.

2.6. Open-source software

CanNm does not use open-source software.