Elektrobit

EB tresos® AutoCore Generic 8 LIN Stack documentation

release notes update for the LinIf 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.

Table of Contents

1. Overview 4
2. LinIf module release notes 5
   2.1. Change log 5
   2.2. New features 19
   2.3. Elektrobit-specific enhancements 19
   2.4. Deviations 20
   2.5. Limitations 30
   2.6. Open-source software 31

1. Overview

This document provides you with the release notes to accompany an update to the `LinIf` 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: 29.2.0
- AUTOSAR R4.0 Rev 3
- Build number: B577598

2. LinIf module release notes

- AUTOSAR R4.0 Rev 3
- AUTOSAR SWS document version: 4.0.0
- Module version: 5.8.40.B577598
- Supplier: Elektrobit Automotive GmbH

2.1. Change log

This chapter lists the changes between different versions.

Module version 5.8.40

2022-11-04

- Add support for LinIfScheduleChangeNextTimeBase
- ASCLINIF-1359 Fixed known issue: LinIfBusIdleTimeoutPeriod cannot be configured to the maximum value

Module version 5.8.39

2022-10-28

- ASCLINIF-1359 Fixed known issue: LinIfBusIdleTimeoutPeriod cannot be configured to the maximum value
- Add support for LinIfScheduleChangeNextTimeBase

Module version 5.8.38

2022-09-16

- ASCLINIF-1340 Fixed known issue: LinIf slave node does not go to sleep after bus idle timeout
- Node response tolerance 0-40% for SAE standard (instead of ISO 40% fixed)
- Updated requirement Id format in module documentation and source code tracing comments. Note: This does not change the Baseline, nor functionality.

Module version 5.8.37

2022-08-19

- Add Tx and Rx Bus-Adapter user specific callout functions

Module version 5.8.36

2022-06-10

- ASCLINIF-1310 Fixed known issue: Compilation error occurs due to inclusion of inexistent header

Module version 5.8.35

2022-05-13

- ASCLINIF-1300 Fixed known issue: LinIf_Cbk.h singleton inclusion (by Lin driver) causes compiler warning/error

Module version 5.8.34

2022-03-18

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

Module version 5.8.33

2022-02-18

- ASCLINIF-1294 Fixed known issue: Det is wrongfully called for LIN_TX_HEADER_ERROR and LIN_RX_ERROR
- Extended configurable user callout to SRF and MFR

Module version 5.8.32

2022-01-28

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

Module version 5.8.31

2021-11-12

- ASCLINIF-1281 Fixed known issue: LinIf_Cbk exported functions not available to some drivers

Module version 5.8.30

2021-10-08

- ASCLINIF-1276 Fixed known issue: LinIf Slave is affected by shared data race

Module version 5.8.29

2021-09-17

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

Module version 5.8.28

2021-08-20

- ASCLINIF-1272 Fixed known issue: End of schedule notification is not called if retrieving data of the last frame failed
- ASCLINIF-1263 Fixed known issue: LinIf_SetPIDTable() and LinIf_SetConfiguredNAD() fail if Det is enabled and masters are configured

Module version 5.8.27

2021-06-25

- Added LIN Slave support

Module version 5.8.26

2021-05-28

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

Module version 5.8.25

2021-04-30

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

Module version 5.8.24

2021-04-09

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

Module version 5.8.23

2021-03-05

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

Module version 5.8.22

2021-01-22

- ASCLINIF-1241 Fixed known issue: Duplicate frame priority possible

Module version 5.8.21

2020-12-18

- ASCLINIF-1237 Fixed known issue: Transceiver function list is wrongly populated in LinIf_Cfg.c

Module version 5.8.20

2020-10-23

- Increased upper-bound configuration limit of the LinTpP2Max and LinTpP2Timing parameters

Module version 5.8.19

2020-09-25

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

Module version 5.8.18

2020-07-31

- ASCLINIF-1218 Fixed known issue: LinIfEbGeneralBswmdImplementation raises an error if transceiver support is disabled

Module version 5.8.17

2020-06-19

- ASCLINIF-1210 Fixed known issue: LinIf switches to operational before time
- Schedule table switch behavior when same schedule table is called refined

Module version 5.8.16

2020-04-24

- ASCLINIF-1207 Fixed known issue: LinIf confirms sleep to LinSM even though a CDD is configured

Module version 5.8.15

2020-03-25

- ASCLINIF-1203 Fixed known issue: Wakeup during sleep transition does not work as expected for ASR 4.2.2 and above drivers

Module version 5.8.14

2020-02-21

- ASCLINIF-1191 Fixed known issue: LinIf does not confirm a schedule switch to NULL_SCHEDULE caused by a sleep request

Module version 5.8.13

2020-01-24

- ASCLINIF-1189 Fixed known issue: Transceiver list is wrongly populated in LinIf_Macros.m

Module version 5.8.12

2019-12-06

- ASCLINIF-1187 Fixed known issue: The LinTp_GetAvailablePduRTxBufferLength does not initialize pduInfo.SduDataPtr

Module version 5.8.11

2019-11-08

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

Module version 5.8.10

2019-10-11

- ASCLINIF-1165 Fixed known issue: Module configuration pointer access occurs before checking for uninitialized access of the function

Module version 5.8.9

2019-09-06

- Add 4.0 and 4.2 Lin driver initialization support

Module version 5.8.8

2019-07-12

- ASCLINIF-1150 Fixed known issue: LinIf_ScheduleRequest uses LinIf Channel ID as ComM Channel ID
- ASCLINIF-1155 Fixed known issue: End of Schedule Notification erroneously called before the last entry's status check

Module version 5.8.7

2019-06-14

- ASCLINIF-1136 Fixed known issue: LinTp does not notify PduR that functional/physical transmission was aborted because schedule table change failed

Module version 5.8.6

2019-05-17

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

Module version 5.8.5

2019-04-18

- ASCLINIF-1130 Fixed known issue: Frame reporting to Mirror during transmission non-functional

Module version 5.8.4

2019-03-22

- ASCLINIF-1127 Fixed known issue: LinIf accesses the post-build configuration without checking the channel ID

Module version 5.8.3

2019-02-15

- Internal module improvement. This module version update does not affect module functionality
- ASCLINIF-1119 Fixed known issue: LinTp_Transmit()/LinIf_Transmit() do not notify the upper layer if the Lin channel is in NO_COMM.

Module version 5.8.2

2019-01-25

- Added Support for NMoE (BusMirroring).

Module version 5.8.1

2018-12-21

- ASCLINIF-1112 Fixed known issue: Symbolic name values for LinIfChannels are erroneously taken from ComM.

Module version 5.8.0

2018-10-26

- ASCLINIF-1101 Fixed known issue: LinIf assigns slave-to-slave frames to incorrect slots
- Changed LinIf APIs incorrectly expecting ComM handle IDs

Module version 5.7.5

2018-08-24

- Added support for forwarding the status from `Lin_GetStatus()` to the user callout
- Added support for `Lin Confirmation Notification` and `LIN_RX_NO_REPONSE` handling in the user callout

Module version 5.7.4

2018-06-22

- Added support referenceable NULL_SCHEDULE LinIfScheduleTable

Module version 5.7.3

2018-05-25

- Added support for configurable upper layer

Module version 5.7.2

2018-04-20

- Add support for UINT32 PduLengthType.
- Added support for custom end-of-schedule notifications

Module version 5.7.1

2017-09-22

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

Module version 5.7.0

2017-07-28

- Fine grained DEM reporting
- Comply to MISRA-C:2012

Module version 5.6.3

2017-06-30

Module version 5.6.2

2017-06-02

Module version 5.6.1

2017-05-05

- ASCLINIF-1041 Fixed known issue: LinIf_LinDriverConfig[ ] is generated empty if Lin configuration name is not LinGlobalConfig_0
- ASCLINIF-1042 Fixed known issue: If the VendorApiInfix parameter is not present in the Lin driver, the LinIf will not generate
- ASCLINIF-1043 Fixed known issue: If LinIfLinDriverAPI is 'REV42' and LinIfCheckWakeupSupported is not activated, LinIf_LinDriverWakeupIntFctPtrType is not available

Module version 5.6.0

2017-03-31

- Internal module improvement. This module version update does not affect module functionality
- Add proper name mangling for header files and API functions of Lin and LinTrcv
- Implement Lin transceiver support

Module version 5.5.0

2017-03-10

- Internal module improvement. This module version update does not affect module functionality
- Implement support for 4.2.x Lin drivers

Module version 5.4.9

2017-02-03

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

Module version 5.4.8

2016-11-04

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

Module version 5.4.7

2016-09-09

- ASCLINIF-1005 Fixed known issue: Config parameter NumberOfRespPendingFrames is used in a wrong way. Decrement NumberOfRespPendingFrames by one in order to keep the same (erroneous) behavior as before.

Module version 5.4.6

2016-08-05

- ASCLINIF-1004 Fixed known issue: NRC response pending frame does not restart P2 timer

Module version 5.4.5

2016-05-25

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

Module version 5.4.4

2016-02-05

- ASCLINIF-990 Fixed known issue: Nested MemMap section if TS_MERGED_COMPILE is activated
- ASCLINIF-991 Fixed known issue: LinIfSupplierId cannot be set to 32767
- Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeaderFile`

Module version 5.4.3

2015-11-06

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

Module version 5.4.2

2015-06-19

- Fixed error reported by broken ENABLE xdm check of the LinIfCollisionResolvingRef parameter
- Adapted source code comments with ReqM2 tags to conventions
- Removed misra deviation comment 19.1 from source code

Module version 5.4.1

2015-02-20

- Removed configuration parameter LinIfTrcvWakeupNotification (LINIF048_Conf)
- Changed parameter range for LinTpP2Timing, LinTpP2Max, LinIfFunctionId
- Modified LinIf to cancel a go-to-sleep command request if wakeup is requested before the go-to-sleep command is transmitted
- Added configuration check for maximum Pdu length
- Modified LinIf to call LinSM_ScheduleRequestConfirmation() even if the current run continuous schedule table is requested

Module version 5.4.0

2014-10-03

- ASCLINIF-930 Fixed known issue: If more than 255 unconditional frames are configured, a schedule table might process an unexpected frame
- Added an optional callout which is called in case of Lin bus errors for user error handling
- ASCLINIF-939 Fixed known issue: It is not possible to send MRF and receive SRF frames without LinTp
- Removed obsolete legacy symbolic name values
- ASCLINIF-946 Fixed known issue: LinIf_GotoSleep, LinIf_Wakeup, LinIf_ScheduleRequest may access configuration data of not initialized module

Module version 5.3.3

2014-04-25

- Removed xdm check which verifies that `LinIfEntryIndex`must start from `0`and be consecutive within one schedule table
- Added xdm check which verifies that `LinIfDelay`is bigger than the maximum frame transmission duration + `LinIfJitter`
- ASCLINIF-909 Fixed known issue: LinIf may call Lin API functions with an incorrect channel ID if `LinIfMapChannelIdDirect`is set to true
- ASCLINIF-913 Fixed known issue: LinIf may call ComM API functions with an incorrect channel ID if `LinIfMapComMChannelIdDirect`is set to true
- ASCLINIF-912 Fixed known issue: LinIf BSWMD is generated with invalid information causing RTE to report an error
- ASCLINIF-923 Fixed known issue: Build error due to missing file `LinIf/LinTp_PBcfg.c`if code generation for LinIf/LinTp is disabled and only post-build configuration is compiled

Module version 5.3.2

2013-10-11

- Removed compiler warning about unused variable `ScheduleChange`if `LINTP_SCHEDULE_CHANGE_DIAG_API == STD_OFF`
- Added defensive programming instrumentation for unreachable code fragments
- ASCLINIF-837 Fixed known issue: Physical transmission might not properly abort if a new physical transmission is invoked on the same channel
- ASCLINIF-838 Fixed known issue: `LinTp_Transmit()` is rejected if a previous transmission has been requested on the same LIN channel, but the `LinIf_Mainfunction()` has not executed in between these requests

- ASCLINIF-836 Fixed known issue: LinTp does not expect response for user-defined diagnostic messages
- Removed compiler warning about unused variable `invalidWakeupSource`if `LINIF_DEV_ERROR_DETECT == STD_OFF`
- Added xdm check which verifies that LinSM confirmation timeout is greater than the time it takes to execute a goto-sleep command
- ASCLINIF-853 Fixed known issue: A compiler error occurs if `PbCfgM`is used for passing a post-build time configuration to LinTp, but not to LinIf
- ASCLINIF-856 Fixed known issue: `LinIf_Init()` uses const void* for post-build config instead of `const LinIf_ConfigType*`
- Convert enum type definitions to `uint8`types
- ASCLINIF-866 Fixed known issue: If the master request frame (MRF) for a functional transmission fails, `PduR_LinTpTxConfirmation()` is called with a wrong `TxPduId`value
- ASCLINIF-868 Fixed known issue: LinTp might call `BswM_LinTp_RequestMode()` with `LINTP_APPLICATIVE_SCHEDULE`even if LinTp communication is no longer active
- ASCLINIF-869 Fixed known issue: Wrong memory might be accessed when evaluating configuration parameter value `LinTpScheduleChangeDiag`in case of P2 timeout
- Updated symbolic name value naming schema according to AUTOSAR 4.0 Rev 3
- ASCLINIF-870 Fixed known issue: If `LinTp_Transmit()` is called for an uninitialized LinTp, an illegal memory is accessed even if Det is enabled
- Extended MCG to generate XML code for Binary Code Generation

Module version 5.3.1

2013-06-21

- ASCLINIF-755 Fixed known issue: Configuration parameters `LinTpNumberOfRxNSdu`and `LinTpNumberOfTxNSdu`have invalid default values
- ASCLINIF-758 Fixed known issue: LinIf passes wrong `HandleId`when calling `PduR_LinIfRxIndciation`for unconditional Rx-frames
- Added checking of configuration and platform-specific signature to prevent loading of incompatible post-build configuration
- Added checking of published information signature to prevent loading of incompatible post-build configuration
- ASCLINIF-788 Fixed known issue: It is not possible to receive messages with a payload length larger than 255 bytes
- ASCLINIF-789 Fixed known issue: `LinTp_CancelReceive()` does not work if the value of parameter `LinTpRxSduId`is larger than 255

- ASCLINIF-804 Fixed known issue: PbcfgMcannot differentiate LinTp and LinIf configuration
- ASCLINIF-801 Fixed known issue: LinIf post-build time configuration does not compile if used by PBcfgM
- ASCLINIF-797 Fixed known issue: LinTp ignores receive messages containing 7 bytes payload length
- ASCLINIF-796 Fixed known issue: LinTp passes the wrong value for the networkparameter when calling BswM_LinTp_RequestMode()
- ASCLINIF-817 Fixed known issue: Memory mapping macros incorrectly define both variables and constants with the same memory section name
- ASCLINIF-808 Fixed known issue: Processing of empty schedule tables may cause transmission of unexpected frames
- ASCLINIF-821 Fixed known issue: LinTp does not call BswM_LinTp_RequestMode()with parameter LINTP_DIAG_REQUESTat the beginning of a functional transmission

Module version 5.3.0

2013-02-14

- Registered HandleIdwizard for ScheduleTableIdxgeneration
- Updated reference paths of LinIf- ComMChannelreference for the introduction of ComMConfigSetcontainer
- Added relocatability to post-build configuration

Module version 5.2.0

2012-10-24

- ASCLINIF-653 Fixed known issue: Post-build configuration of LinIf and LinTp references external symbols when used with post-build configuration manager
- ASCLINIF-651 Fixed known issue: The configuration name is different from the name of the MULTIPLE-CONFIGURATIONcontainer
- Implemented Tp-API according to AUTOSAR 4.0 Rev 3
- Implemented Handle ID policy according to AUTOSAR 4.0 Rev 3
- Changed the top-level structure of the software-component description in the ARXML files from /AUTOSAR/LinIfto /AUTOSAR_LinIf
- Updated to Lin Specification Package Revision 2.1
- ASCLINIF-702 Fixed known issue: Wrong ComMChannelIdis used if LinIfMapComMChannelIdDirectis enabled, but LinIfChannelIddoes not match ComMChannelId

Module version 5.1.0

2012-06-20

- Introduced post-build data structures

Module version 5.0.0

2012-03-16

- Initial AUTOSAR 4.0 version
- Updated naming scheme for #defines for symbolic name values to AUTOSAR 4.0 Rev 3 naming scheme
- Updated config to AUTOSAR 4.0 Rev 3 schema
- Added support of AUTOSAR 4.0 Rev 3 Lin MCAL module

2.2. New features

- LinIf supports LinIfScheduleChangeNextTimeBase.

2.3. Elektrobit-specific enhancements

This chapter lists the enhancements provided by the module.

- Configurable support of AUTOSAR 4.0 Rev 3, and 4.2 Lin MCAL Module

  The configuration parameter `LinIfLinDriverAPI` allows to configure the LIN Interface module to support a specific Lin MCAL Module.

  `LinIfLinDriverAPI`:

  - Rev 2: Use Lin according to AUTOSAR Specification of LIN Driver V1.4.0 R4.0 Rev 2.
  - Rev 3: Use Lin according to AUTOSAR Specification of LIN Driver V1.5.0 R4.0 Rev 3.
  - 4.2: Use Lin according to AUTOSAR Specification of LIN Driver 4.2.1/4.2.2.
  - 4.3.1: Use Lin according to AUTOSAR Specification of LIN Driver 4.3.1.
  - 4.4: Use Lin according to AUTOSAR Specification of LIN Driver 4.4.0.
- Implementation of receive cancellation

  Contrary to the AUTOSAR 4.0 Rev 3 specification, cancellation of ongoing receptions by a call to `LinTp_CancelReceive` is implemented.

- Callout for Lin bus error-handling

  EB LinIf implements the two additional configuration parameters `LinIfLinErrorCalloutName` and `LinIfLinErrorCalloutHeaderFile` which enable LinIf to call a user-definable callout function in case of Lin bus communication errors.

- Vendor specific configuration parameters were introduced to support configurable reporting of the production errors "Bit-Error `(LINIF_E_TX_BIT_ERROR)` ", "Checksum-Error `(LINIF_E_RX_CHECKSUM_ERROR)` " and "Slave-Not-Responding-Error `(LINIF_E_RX_NO_RESPONSE_ERROR)` ".

  Description:

  Vendor specific configuration parameters `LinIfTxBitErrorReportToDem` , `LinIfTxBitErrorDemDetErrorId` , `LinIfTxBitErrorDebounceMethod` , `LinIfRxChecksumErrorReportToDem` , `LinIfRxChecksumErrDemDetErrorId` , `LinIfRxChecksumDebounceMethod` , `LinIfRxNoRespErrorReportToDem` , `LinIfRxNoRespErrDemDetErrorId` and `LinIfRxNoRespDebounceMethod` ,were introduced to support configurable reporting of the production errors above.

- Vendor specific configuration parameters: `LinIfScheduleTableEndNotificationSupported`, `LinIfScheduleTableEndNotificationCallout`, `LinIfScheduleTableEndNotificationRef` allow having custom end-of-schedule notifications.

- Added support referenceable NULL_SCHEDULE LinIfScheduleTable.

- Added support for forwarding the status from `Lin_GetStatus()` to the user callout.

- Added support for `Lin Confirmation Notification` and `LIN_RX_NO_REPONSE` handling in the user callout.

- LinIf now supports referencing BSWMD for Lin driver/transceiver from which to extract the Vendor ID and Vendor API Infix.

- Added support for solving the inconsistency between the LinIf and Lin drivers with an autosar version lower than 4.3. (check https://bugzilla.autosar.org/show_bug.cgi?id=73095). If the LinIf channel starts in SLEEP, at initialization LinIf forces the driver channel into sleep. If calling `Lin_GoToSleepInternal()` returns *E_NOT_OK*, a DET is called. `LINIF_DRIVER_CHANNEL_NOT_IN_SLEEP` was chosen for this purpose with reserved ID `0xFF`.

- Added support for requesting the same schedule table. If the same schedule table is requested (as the one that is running) the schedule table will be restarted.

- Increased upper-bound configuration limit of the LinTpP2Max and LinTpP2Timing parameters to 65535s.

2.4. Deviations

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

- LinTp does not provide the API function `LinTp_Shutdown()` (reference to product description: ASCPD-96)

  Description:

  The API function `LinTp_Shutdown()` is not implemented in the LinTp module.

  Rationale:

  There is no AUTOSAR internal user for the API function `LinTp_Shutdown()` and the behavior and operating constraints are not clearly specified in the AUTOSAR SWS. Using the function might be risky since expectations and actual behavior might differ, so it was decided to skip the function implementation.

  Requirements:

  SWS_LinIf_00355, SWS_LinIf_00356, SWS_LinIf_00433, SWS_LinIf_00357, SWS_LinIf_00482, SWS_LinIf_00484, SWS_LinIf_00683

- The `LinIf_Transmit()` function does not reject transmission requests of non-sporadic frames

  Description:

  If an upper layer requests to transmit an unconditional frame which is not associated to a sporadic frame slot, the function `LinIf_Transmit()` returns `E_OK`.

  Rationale:

  This deviation in behavior (i.e., the fact that the `LinIf_Transmit()` function does not reject transmission requests of non-sporadic frames) is required in order to support gateway operation. Because in gateway mode, the upper layer (i.e., the PduR) does not know about sporadic frames and calls `LinIf_Transmit()` unconditionally. If the `LinIf_Transmit()` returns `E_NOT_OK` in that case, unconditional frames might get lost. See http://www.autosar.org/bugzilla/show_bug.cgi?id=51794.AUTOSAR 4.1.1 [SWS_LinIf_00700]

  Requirements:

  SWS_LinIf_00341

- ASCCCB-1403: Initialization check in `LinIf_MainFunction()`

  Description:

  If `LinIf_MainFunction()` is called while the module is not yet initialized, `LinIf_MainFunction()` returns immediately without performing any functionality and without raising any Det error. This initialization check is always performed independent of the development error detection setting.

  Rationale:

The SchM module may schedule the modules main function before the module is initialized. This would result in lots of Det errors during startup. Therefore the module's main function does not throw a Det error if the module is not yet initialized and simply returns in this case.

Requirements:

SWS_LinIf_00535

- The function `LinIf_CheckWakeup()` is reentrant only for different LIN channels

  Description:

  `LinIf_CheckWakeup()` cannot be interrupted by another `LinIf_CheckWakeup()` call.

  Rationale:

  `LinIf_CheckWakeup()` calls `Lin_CheckWakeup` which is non-reentrant, therefore `LinIf_CheckWakeup()` also needs to be non-reeantrant.

  Requirements:

  SWS_LinIf_00378

- Call of `LinIf_ScheduleRequest()` within 100ms after `LinIf_Wakeup()` may lead to an unexpected behavior

  Description:

  If `LinIf_ScheduleRequest()` is called after `LinIf_Wakeup()` within 100ms, it could be that a slave neither receives data nor transmits a response.

  Rationale:

  After a wake-up signal is sent to a LIN cluster in sleep mode, the slaves may take up to 100ms before they can communicate. Only if the slaves are ready, the master shall start communication again (LIN Protocol Specification, Revision 2.0, Section 5.1). The LIN Interface does not enforce this delay, so if frames are sent immediately after the wake-up, slaves might miss them.

- Race conditions might lead to a wrong schedule table being active during sleep mode.

  Description:

  Issuing a schedule request (via `LinIf_ScheduleRequest()`) while the LinIf is performing the transition into sleep mode (due to a `LinIf_GotoSleep()`) call. might cause the LinIf to end up in sleep mode with another schedule table than the NULL schedule being active due to internal race conditions. Note: If the LIN State Manager (LinSM) is used as upper layer for the LinIf (as designed by AUTOSAR) the module takes care that `LinIf_ScheduleRequest()` is not called during transition into sleep mode.

- Only one frame reference per schedule table entry supported

Description:

A schedule table entry does not allow configuration of more than one frame reference.

Requirements:

ECUC_LinIf_00016

- LinIfPublicCddHeaderFile parameter

  Description:

  The configuration parameter LinIfPublicCddHeaderFile besides CDDs is used for user defined end-of-schedule notifications as well.

  Requirements:

  LinIf.ASR40.ECUC_LinIf_00631

- Deviating post-build implementation

  Description:

  The `PbcfgM` offers the opportunity to initialize the LinIf and LinTp with different configurations during run-time. Therefore it is possible to call `LinIf_Init()` and `LinTp_Init()` more than once.

  Requirements:

  SWS_LinIf_00562, SWS_LinIf_00593, SWS_LinIf_00376

- Development error code

  `LINIF_E_NC_NO_RESPONSE` is not reported

  Description:

  If a SRF is put in a schedule table after a node configuration frame and a slave does not answer the runtime error code, `LINIF_E_NC_NO_RESPONSE` is not reported.

  Requirements:

  SWS_LinIf_00405, SWS_LinIf_00376

- No support of configuration parameter

  `LinIfNcOptionalRequestSupported` (reference to product description: ASCPD-61)

  Description:

  The configuration parameter `LinIfNcOptionalRequestSupported` is not supported. Node configuration frames cannot be disabled.

Rationale:

Configuration node frames are sent as fixed frames and they are not distinguished.

Requirements:

SWS_LinIf_00310

- No support of configuration parameter

  `LinIfNcOptionalRequestSupported` (reference to product description: ASCPD-61)

  Description:

  The configuration parameter `LinIfNcOptionalRequestSupported` is not supported. Support for NAD assignment (SID 0xB0) is always enabled.

  Rationale:

  The same configuration parameter is used for masters and slaves.

  Requirements:

  SWS_LinIf_00810

- Range of `LinIfLinProtocolVersion` restricted

  Description:

  The configuration parameter `LinIfLinProtocolVersion` has the range restricted to ISO17987.

  Rationale:

  The SWS refers to ISO17987. ISO 17987 is compatible with LIN 2.1 spec, on which the Master implementation is based.

  Requirements:

  ECUC_LinIf_00647

- Reception is aborted if `PduR_StartOfReception()` returns `BUFREQ_E_BUSY`

  Description:

  If `PduR_StartOfReception()` returns `BUFREQ_E_BUSY` and a buffer size smaller than the payload of the SF or FF, the LIN interface does not retry to copy data to PduR.

  Rationale:

  The LinTp does not support buffering of received data from the Lin driver.

Requirements:

SWS_LinIf_00679

- LinTrcv.h header inclusion

  Description:

  The `LinTrcv.h`header is included via the `LinIf_TrcvTypes.h` header, not directly in the main source file. Also, the name depends on the configuration parameters - `LinIfSingleLinTrcvAPIInfixEnable`- `LinIfMultipleTrcvDriverSupported` If any of the above parameters is set to TRUE, the naming is according to http://www.autosar.org/bugzilla/show_bug.cgi?id=53325 .

  Requirements:

  SWS_LinIf_00555

- Parameter type differs from specified

  Description:

  The configuration parameter LinIfCddRef isn't implemented as a having the type of a foreign reference but as a choice reference with values limited to [ECUC-MODULE-CONFIGURATION-VALUES].

  Requirements:

  ECUC_LinIf_00637

- Parameter existence criteria

  Description:

  The requirement from the SWS states that LinIfCddRef is only needed when LinIfWakeupConfirmationUL, LinIfScheduleRequestConfirmationUL and/or LinIfGotoSleepConfirmationUL is set to CDD. This enumeration is extended by LinIfUserRxIndicationUL and LinIfUserTxUL.

  Requirements:

  ECUC_LinIf_00637

- Parameter existence criteria

  Description:

  The requirement from the SWS list the `LinIfRxIndicationUL`, `LinIfTxConfirmationUL` and `LinIfTxTriggerTransmitUL` parameters as having the type `EcucFunctionNameDef`.

  Due to the fact that parent container is PB, the type was changed to `EcucReferenceDef.`

  Requirements:

ECUC_LinIf_00055

ECUC_LinIf_00054

ECUC_LinIf_00628

- Upper Limit for LinIfBusIdleTimeoutPeriod is changed

  Description:

  The allowed configurable upper limit for LinIfBusIdleTimeoutPeriod has been changed from Infinity to 4294967295s divided by LinIfTimeBase.

  This has been done to allow timeout values that are greater than the ones specified in the ISO 17987-2:2016(E) standard without actually disabling it.

- Lack of support for reliable TxConfirmation

  Description:

  &lt;User_TxConfirmation&gt; is called whenever LinIf_TxConfirmation gets called.

  No negative confirmations are provided in order to maintain backwards (AUTOSAR 4.0 - 4.2) compatibility.

- LinTpTsNsdu parameter LinTpMaxBufReq not supported

  Description:

  When `PduR_LinTpCopyTxData()` does not return `BUFREQ_OK`, the transmission of the SRF is aborted and `PduR_LinTpTxConfirmation()` is called with `NTFRSLT_E_NOT_OK`.

  Requirements:

  LinIf.SWS_LinIf_00330.Slave

  LinTp.ECUC_LinTp_00637

  LinTp.ASR20-11.CopyTxErrorSlave

- New functionalities

  Description:

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

  Rationale:

  Requirements:

  SWS_LinIf_00709,SWS_LinIf_00710,SWS_LinIf_00723,SWS_LinIf_00724

- LinIf_MainFunction_LinIfChannel.ShortName not implemnted

  Description:

  Actual implementation does not support `LinIf_MainFunction_LinIfChannel.ShortName`.

  Rationale:

  Requirements:

  SWS_LinIf_00712,SWS_LinIf_00043,SWS_LinIf_00053,SWS_LinIf_00286,SWS_LinIf_00287,SWS_LinIf_00293,SWS_LinIf_00384,SWS_LinIf_00453,SWS_LinIf_00454,SWS_LinIf_00455,SWS_LinIf_00473,SWS_LinIf_00597

- Changes in error types

  Description:

  According to AR20-11 there is some changes in error types.

  Rationale:

  AR20-11 functionality changes

  Requirements:

  SWS_LinIf_00539,SWS_LinIf_00546,SWS_LinIf_00549,SWS_LinIf_00570,SWS_LinIf_00571,SWS_LinIf_00573,SWS_LinIf_00574,SWS_LinIf_00575,SWS_LinIf_00595,SWS_LinIf_00639,SWS_LinIf_00640

- Change in name of LinIf reference for time

  Description:

  Name of LinIf reference for time (time base or main function period) changed, but delay functionality is supported.

  Rationale:

  AR20-11 functionality changes

  Requirements:

  SWS_LinIf_00261

- Requirment Changes in AR20-11

  Description:

  The listed requirements are Functionality changed in AR20-11

Rationale:

AR20-11 functionality changes

Requirements:

SWS_LinIf_00613,SWS_LinIf_00614,SWS_LinIf_00642,SWS_LinIf_00643,SWS_LinIf_00648,SWS_LinIf_00655,SWS_LinIf_00658,SWS_LinIf_00662,SWS_LinIf_00666,SWS_LinIf_00201,SWS_LinIf_00081,SWS_LinIf_00086,ECUC_LinIf_00042,SWS_LinIf_00501,SWS_LinIf_00351,ECUC_LinIf_00021

- Implementation currently based on LinIfTimeBase

  Description:

  Rationale:

  AR20-11 functionality changes

  Requirements:

  ECUC_LinIf_00639

- Partly implemented functionalities

  Description:

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

  Rationale:

  Requirements:

  ECUC_LinIf_00040,SWS_LinIf_00877,SWS_LinIf_00879,SWS_LinIf_00653

- Unsupported functionalities

  Description:

  The listed requirements are not supported.

  Rationale:

  Requirements:

  SWS_LinIf_00702,SWS_LinIf_00707,SWS_LinIf_00708,SWS_LinIf_00719,SWS_LinIf_00725,SWS_LinIf_00726,SWS_LinIf_00728,SWS_LinIf_00878

- User_RxIndication new in parameter.

  Description:

User_RxIndication is implemented according to Autosar 4.0.3 and it only takes RxPduId as parameter. According to ASR20-11 it should also take result as parameter.

Rationale:

AR20-11 functionality changes

Requirements:

SWS_LinIf_00530

- User_TxConfirmation new in parameter.

  Description:

  User_TxConfirmation is implemented according to Autosar 4.0.3 and it only takes TxPduId as parameter. According to ASR20-11 it should also take result as parameter.

  Rationale:

  AR20-11 functionality changes

  Requirements:

  SWS_LinIf_00529,SWS_LinIf_00128

- New LinTp Configuration

  Description:

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

  Rationale:

  Requirements:

  ECUC_LinTp_00638,ECUC_LinTp_00635,ECUC_LinTp_00636,ECUC_LinTp_00073

- Changed LinTp Configuration

  Description:

  The listed requirements are Functionality changed in AR20-11

  Rationale:

  AR20-11 functionality changes

  Requirements:

ECUC_LinTp_00056,ECUC_LinTp_00624,ECUC_LinTp_00072,ECUC_LinTp_00428,ECUC_LinTp_00061,ECUC_LinTp_00060,ECUC_LinTp_00511,ECUC_LinTp_00065,ECUC_LinTp_00064

- Slave nodes do not support SAE.

  Description:

  Slave nodes do not support SAE_J2602 standard.

- Master nodes do not support 0x3E for MRF when SAE is used.

  Description:

  SAE_J2602 standard uses the id 0x3E for MRF and it is not supported.

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.

- Limitation: Link time Cdd support

  Description:

  The configuration container LinIfChannel is post-build capable but the ConfigurationClass of upper layer Cdd support parameters is VARIANT-LINK-TIME.

  Rationale:

  The function pointers aren't generated within the post-build data structure (limitation).

  It is considered to be the integrator's responsibility when modifying the LinIfChannel container to ensure that the set of references to LinSM, PduR or Cdds are identical to one present during LinkTime configuration.

  Limitation: Compatibility with LinSM module

  Description:

  If used with a LinSM module from Elektrobit, the supported minimum LinSM version is 3.4.0.

  Rationale:

  APIs from earlier versions expect ComM, instead of LinIf handle IDs.

  Limitation: Bus Mirroring number of channels

Description:

Maximum number of channels that are mirrored is 16

Rationale:

Implementation constraint from using uint16. In case of channel ID greater than the maximum mirrored channels, there will be an error reported to DET (error ID LINIF_E_INVALID_MIRROR_CHANNEL 0x70U).

Limitation: Drivers of different Autosar version

Description:

LinIf cannot use drivers of different Autosar version.

Rationale:

The configuration parameter LinIfLinDriverAPI specifies what version of Autosar the driver is expected to be. All other drivers of different Autosar versions are ignored.

Limitation: LinIfBusIdleTimeoutPeriod

Description:

The value that can be configured for LinIfBusIdleTimeoutPeriod as an upper limit depends on LinIfTimeBase. The division between the two of them needs to be smaller that 4294967295s (i.e for a LinIfTimeBase of 0.005, the max value LinIfBusIdleTimeoutPeriod can have is 21474836s) .

Rationale:

To be able to monitor the timeout, a transformation into Number of Main Functions needed to be done. The decision is to limit it to uint32, since the Idle Timeout is supposed to be between 4s and 10s . uint32 seems like a reasonable value to allow also some flexibility. Bigger values were not considered as an Idle Timeout greater than 10s is anyhow not ISO compliant.

2.6. Open-source software

LinIf does not use open-source software.