


SCOPE OF APPLICATION All Project/Engineering		SHT/SHTS 1 / 47
Responsibility: Classic AUTOSAR Team	LinIf Manual	DOC. NO
<h1>LinIf User Manual</h1>		

Document Change History			
Date (YYYY-MM-DD)	Ver.	Editor	Content
2020-12-22	1.0.0.0	SJ Kim	- Initial version
2021-04-20	1.0.1.0	LongTQB	- Second version
2022-02-28	1.0.2.0	HJ Seong	- Update Scope of the Release - Add Change Log
2022-04-29	1.0.3.0	DuyNHP2	- Update Scope of the Release - Add Change Log
2022-06-28	1.0.4.0	HJ Seong	- Update Scope of the Release - Add Change Log
2022-08-04	1.0.5.0	KhaLN1	- Update Scope of the Release - Add Change Log
2022-10-27	1.0.6.0	KhaLN1	- Update Scope of the Release - Add Change Log
2022-11-03	1.0.7.0	KhaLN1	- Update Scope of the Release - Add Change Log
2022-11-30	1.0.7.1	KhaLN1	- Add Change Log
2023-01-11	1.0.8.0	KhaLN1	- Update Scope of the Release - Add Change Log - Add parameter "LinIfBlockWakeupRepetitionMax"
2023-02-06	1.0.9.0	KhaLN1	- Update Scope of the Release - Add Change Log
2023-03-07	1.0.10.0	KhaLN1	- Update Scope of the Release - Add Change Log
2023-04-10	1.0.11.0	KhaLN1	- Update Scope of the Release - Add Change Log
2023-06-29	1.0.12.0	JH Jang	- Update Scope of the Release - Add Change Log
2023-08-15	1.0.13.0	KhaLN1	- Update Scope of the Release - Add Change Log
2023-11-20	1.0.14.0	ThuanLD5	- Update Scope of the Release - Add Change Log
2024-03-15	1.0.15.0	JH Jang	- Update UM to the latest template

Edition Date: 2024-03-15	File Name LinIf_UM.pdf	Creation JH Jang	Check JH Cho	Approval DJ Lee
Document Management System				
		2024-03-15	2024-03-15	2024-03-15

Table of Contents

1 Overview	5
2 Acronyms, abbreviations and term	5
3 Reference	7
4 AUTOSAR System	8
4.1 LinIf Module	8
5 Limitations and Deviations	9
5.1 Limitations	9
5.1.1 Multiple Driver Support Not Supported	9
5.1.2 Event-Triggered Frame not supported	9
5.1.3 Slave-To-Slave Frame not supported	9
5.1.4 Slave-To-Slave Frame not supported	9
5.1.5 Slave Node not supported	9
5.2 Deviations	9
6 Configuration Guide	10
6.1 LinIfGeneral	10
6.2 LinIfGlobalConfig-LinIfChannel	11
6.3 LinIfGlobalConfig-LinIfChannel-LinIfFrame	12
6.4 LinIfGlobalConfig-LinIfChannel-LinIfFrame-LinIfFixedFrameSdu	12
6.5 LinIfGlobalConfig-LinIfChannel-LinIfFrame-LinIfSubstitutionFrames	13
6.6 LinIfGlobalConfig-LinIfChannel-LinIfFrame-LinIfPduDirection-LinIfRxPdu	13
6.7 LinIfGlobalConfig-LinIfChannel-LinIfFrame-LinIfPduDirection-LinIfTxPdu	13
6.8 LinIfGlobalConfig-LinIfChannel-LinIfScheduleTable	14
6.9 LinIfGlobalConfig-LinIfChannel-LinIfScheduleTable-LinIfEntry	14
6.10 LinIfGlobalConfig-LinIfChannel-LinIfNodeType-LinIfMaster	15
6.11 LinIfGlobalConfig-LinIfChannel-LinIfNodeType-LinIfSlave	15
6.12 LinIfGlobalConfig-LinIfChannel-LinIfNodeType-LinIfSlave-LinIfNodeConfigurationIdentification	15
6.13 LinIfGlobalConfig-LinIfChannel-LinIfFrame-LinIfPduDirection-LinIfSlaveToSlavePdu	16
6.14 LinIfGlobalConfig-LinIfChannel-LinIfFrame-LinIfPduDirection-LinIfInternalPdu	16
6.15 LinIfGlobalConfig-LinIfChannel-LinIfTransceiverDrvConfig	16
6.16 LinTpGeneral	16
6.17 LinTpGlobalConfig	17
6.18 LinTpGlobalConfig-LinTpChannelConfig	17
6.19 LinTpGlobalConfig-LinTpRxNSdu	17
6.20 LinTpGlobalConfig-LinTpTxNSdu	18

7 Application Programming Interface (API)	20
7.1 Type Definitions	20
7.1.1 LinIf_SchHandleType	20
7.1.2 LinIf_ConfigType	20
7.1.3 LinTp_ConfigType	20
7.1.4 LinTp_Mode	21
7.2 Macro Constants	21
7.3 Functions	21
7.3.1 LinIf_Init	21
7.3.2 LinIf_GetVersionInfo	22
7.3.3 LinIf_Transmit	22
7.3.4 LinIf_ScheduleRequest	23
7.3.5 LinIf_GotoSleep	24
7.3.6 LinIf_Wakeup	25
7.3.7 LinIf_SetTrcvMode	26
7.3.8 LinIf_GetTrcvMode	27
7.3.9 LinIf_GetTrcvWakeupReason	27
7.3.10 LinIf_SetTrcvWakeupMode	28
7.3.11 LinIf_GetPIDTable	29
7.3.12 LinIf_SetPIDTable	30
7.3.13 LinIf_GetConfiguredNAD	30
7.3.14 LinIf_SetConfiguredNAD	31
7.3.15 LinTp_Init	32
7.3.16 LinTp_Transmit	32
7.3.17 LinTp_GetVersionInfo	33
7.3.18 LinTp_Shutdown	33
7.3.19 LinTp_ChangeParameter	34
7.3.20 LinIf_CheckWakeup	35
7.3.21 LinIf_EnableBusMirroring	35
7.3.22 LinIf_WakeupConfirmation	36
7.3.23 LinIf_HeaderIndication	37
7.3.24 LinIf_RxIndication	37
7.3.25 LinIf_TxConfirmation	38
7.3.26 LinIf_LinErrorIndication	39
7.3.27 LinIf_MainFunction_<LinIfChannel.ShortName>	39

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
4 / 47

7.3.28 Rte_Call_CallbackError_lin_cbk_err	40
7.3.29 Rte_Call_CallbackAfterSchedule_lin_after_schedule	40
8 Generator	41
8.1 Generator Option	41
8.2 Generator Error Message	41
8.2.1 Error Messages	41
8.2.2 Warning Messages	46
8.2.3 Information Messages	46
9 Appendix	47

Table of Figures

Figure 1 - AUTOSAR BSW software architecture - LIN stack scope	8
--	---

1 Overview

It is written based on AUTOSAR standard SRS / SWS. If more detailed functional explanation is needed when using the module, see the Reference Manual. The interpretation of setting related category is as follows:

- Changeable (C): Items that can be set by the user.
- Fixed (F): Items that cannot be changed by the user.
- Not Supported (N): Deprecated item.

2 Acronyms, abbreviations and term

Abbreviation / Acronym:	Description:
CF	Continuous Frame in LIN TP
FF	First Frame in LIN TP
ID	Identifier
LDF	LIN Description File
LIN TP	LIN Transport Protocol (Part of the LIN Interface)
MRF	Master Request Frame
NAD	Node Address. Each slave in LIN must have a unique NAD.
NC	Node Configuration
N_As	Time for transmission of the LIN frame (any N-PDU) on the sender side (see ISO 17987-2 [19]).
N_Cr	Time until reception of the next consecutive frame N-PDU (see ISO 17987-2 [19]).
N-Cs	Time until transmission of the next consecutive frame N-PDU (see ISO 17987-2 [19]).
P2	Time between reception of the last frame of a diagnostic request on the LIN bus and the slave node being able to provide data for a response.
P2*	Time between sending a response pending frame (0x78) and the LIN-slave being able to provide data for a response.
PID	Protected ID
RX	Reception
SID	Service Identifier (of node configuration service)
SF	Single Frame in LIN TP
SRF	Slave Response Frame

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
6 / 47

SRS	Software Requirement Specification
TX	Transmission

Term:	Description:
Slot Delay	The time between start of frames in a schedule table. The unit is in number of time-bases for the specific cluster.
Jitter	Difference between longest delay and shortest delay (e.g. Worst case execution time – Best case execution time)
Maximum frame length	The maximum frame length is the T_{FRAME_MAX} as defined in the ISO 17987-3 [19] (i.e. The nominal frame length plus 40 %).
Schedule entry is due	This means that the LIN Interface has arrived at a new entry in the schedule table and a frame (received or transmitted) will be initiated.
Slave-to-slave	From a LIN master node's point of view, there exist 3 different directions of frames on the LIN bus: Response transmitted by the master, Response received by the master and Response transmitted by one slave and received by another slave. The slave-to-slave is describing the last one. This is not described explicitly in the ISO 17987 specifications, but mentioned in Figure 14 in ISO 17987-3: Three unconditional frame transfers.
Irrelevant frame	From a LIN slave node point of view, there exist 3 different directions of frames on the LIN bus: Response transmitted by the slave, Response received by the slave and Response that is ignored by the slave (i.e. communication between master and another slave or between two other slaves). These ignored frames are named irrelevant frames in this specification. This is not described explicitly in the ISO17987 specifications.
Relevant frame	From a LIN slave node point of view, a frame that is transmitted or received by the slave. Opposite of irrelevant frame.
Sporadic frame	This is one of the unconditional frames that are attached to a sporadic slot.
Sporadic slot	This is a placeholder for the sporadic frames. The reason to name it slot is that it has no LIN frame ID.
Tick	The tick is the smallest time entity to handle the communication on all channels.
Bus idle timeout	Lapse of time duration with no bus activity

3 Reference

Sl. No.	Title	Version
1	AUTOSAR_SWS_LINInterface.pdf	4.4.0

4 AUTOSAR System

4.1 LinIf Module

- The main responsibilities for the LinIf are:
 - Executing the currently selected schedule for each LIN bus the ECU is connected to (transmitting headers and transmitting/receiving response).
 - Switching schedule tables when requested by the upper layer(s).
 - Accepting frame transmit requests from the upper layers and transmit the data part as response within the appropriate LIN frame.
 - Providing frame receive notification for the upper layer when the corresponding response is received within the appropriate frame.
 - Go to sleep and wake-up services.
 - Error handling.
 - Diagnostic Transport Layer services.

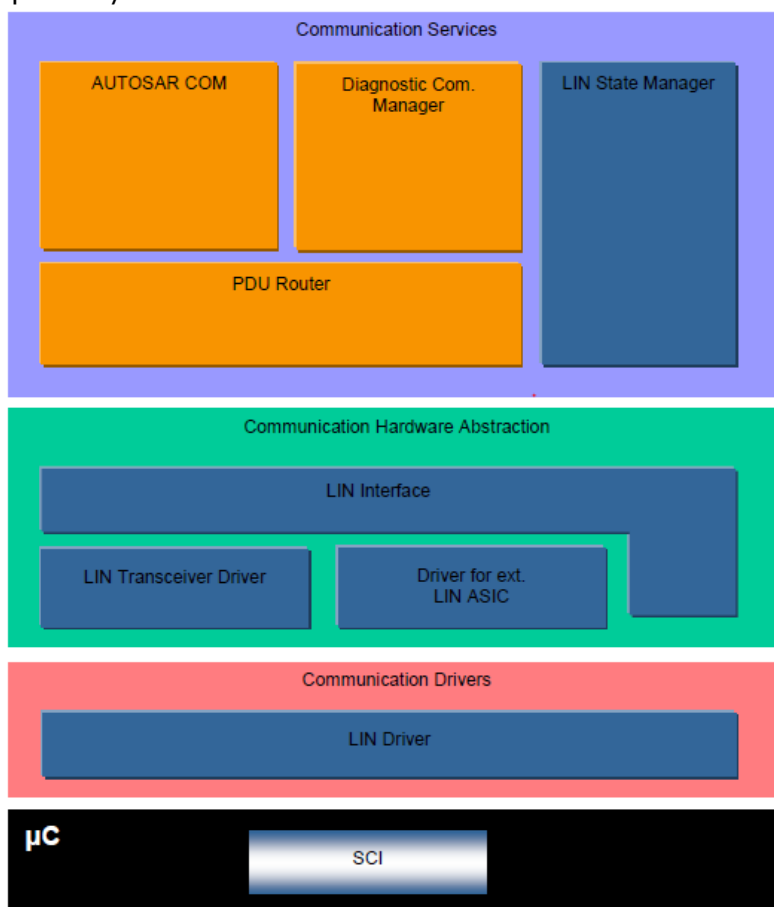


Figure 1 - AUTOSAR BSW software architecture - LIN stack scope

5 Limitations and Deviations

5.1 Limitations

5.1.1 Multiple Driver Support Not Supported

This function uses different LIN Driver (MCAL) for each channel. This function is not supported.

5.1.2 Event-Triggered Frame not supported

Function to receive response from multiple Slave Nodes for one Header. When responses are sent from multiple Slave Nodes at the same time, collision occurs and the Collision Resolving Table is performed. This function is not supported.

5.1.3 Slave-To-Slave Frame not supported

This is a function that communicates between slaves. Master Node transmits only Header and receives Response from Slave Node from other Slave Node. This method does not check frame errors in the master node, so even if a communication error occurs, the master node cannot know it. This function is not supported.

5.1.4 Slave-To-Slave Frame not supported

LinIf does not support LinIfNodeType Slave Node. It means the ECU itself cannot be LIN Slave Node. Currently LinIf is only supports Master Node.

5.1.5 Slave Node not supported

LinIf does not support LinIfNodeType Slave Node. It means the ECU itself cannot be LIN Slave Node. Currently LinIf is only support Master Node.

5.2 Deviations

LinIf does not support LIN TP for slave nodes.

6 Configuration Guide

The LinIf setting of the AUTOSAR platform distributed by Hyundai Auto is a setting reflecting Hyundai Auto Policy's policy. Therefore, you should consult with Hyundai Auto.

The following chapters summarize all configuration parameters of module.

6.1 LinIfGeneral

Parameter Name	Value	Category
¹⁾ LinIfBusMirroringSupported	True/False	N
²⁾ LinIfDevErrorDetect	True	F
³⁾ LinIfMultipleDriversSupported	True/False	N
⁴⁾ LinIfMultipleTrcvDriverSupported	False	F
⁵⁾ LinIfNcOptionalRequestSupported	True/False	N
⁶⁾ LinIfPublicCddHeaderFile	-	N
⁷⁾ LinIfResponseErrorSignalChangedCallout	-	N
⁸⁾ LinIfSaveConfigurationCallout	-	N
⁹⁾ LinIfTpSupported	True/False	C
¹⁰⁾ LinIfTrcvDriverSupported	True	F
¹¹⁾ LinIfVersionInfoApi	True	F
¹²⁾ LinIfBlockWakeupRepetitionMax	1 ... 255	C

- 1) LinIfBusMirroringSupported
 - States if Bus Mirroring is enabled in the LIN Interface or not.
 - The reason for this parameter is to reduce the size of LIN Interface if the Bus Mirroring is not used.
- 2) LinIfDevErrorDetect
 - Switches the development error detection and notification on or off.
- 3) LinIfMultipleDriversSupported
 - States if multiple drivers are supported by the LIN Interface or not.
 - The reason for this parameter is to reduce the size of LIN Interface if multiple drivers are not used.
- 4) LinIfMultipleTrcvDriverSupported
 - States if multiple transceiver drivers are supported by the LIN Interface or not.
 - The reason for this parameter is to reduce the size of LIN Interface if multiple transceiver drivers are not used.
- 5) LinIfNcOptionalRequestSupported
 - States if the node configuration commands Assign NAD and Conditional Change NAD are supported.
- 6) LinIfPublicCddHeaderFile
 - Defines header files for callback functions which shall be included in case of CDDs.
 - Range of characters is 1..32.
- 7) LinIfResponseErrorSignalChangedCallout
 - This parameter contains the name of the callout function that is called after a response error signal change.
 - Only applicable for LIN slave nodes.
- 8) LinIfSaveConfigurationCallout
 - This parameter contains the name of the callout function that is called when a save configuration node configuration command is processed by this slave node.

- The service is only supported when this parameter is configured.
- Only applicable for LIN slave nodes.
- 9) LinIfTpSupported
 - States if the TP is included in the LIN Interface or not.
 - The reason for this parameter is to reduce the size of LIN Interface if the TP is not used.
- 10) LinIfTrcvDriverSupported
 - States if transceiver driver support is included in the LIN Interface or not.
 - The reason for this parameter is to reduce the size of LIN Interface if transceiver drivers are not used.
- 11) LinIfVersionInfoApi
 - Switches the LinIf_GetVersionInfo function ON or OFF.
- 12) LinIfBlockWakeupRepetitionMax
 - Specifies the maximum number of block wake-up repetitions (One block consists of three wake-up signals) when wake-up failed.
 - Minimum: 1
 - Maximum: 255 (Infinitely repeat)

6.2 LinIfGlobalConfig-LinIfChannel

Parameter Name	Value	Category
¹⁾ LinIfBusIdleTimeoutPeriod	Automated	C
²⁾ LinIfGotoSleepConfirmationUL	LIN_SM	F
³⁾ LinIfGotoSleepIndicationUL	LIN_SM/CDD	N
⁴⁾ LinIfMainFunctionPeriod	0.005	F
⁵⁾ LinIfMaxFrameCnt	Automated	C
⁶⁾ LinIfScheduleChangeNextTimeBase	Automated	C
⁷⁾ LinIfScheduleRequestConfirmationUL	LIN_SM	F
⁸⁾ LinIfWakeupConfirmationUL	LIN_SM	F
⁹⁾ LinIfCddRef	-	N
¹⁰⁾ LinIfChannelRef	Automated	F
¹¹⁾ LinIfComMNetworkHandleRef	Automated	F

- 1) LinIfBusIdleTimeoutPeriod
 - Bus idle timeout in seconds.
- 2) LinIfGotoSleepConfirmationUL
 - This parameter defines the upper layer (UL) module to which the confirmation of the goto-sleep command shall be sent.
- 3) LinIfGotoSleepIndicationUL
 - This parameter defines the upper layer (UL) module to which the indication of the goto-sleep command shall be sent.
 - Only used for LIN Slave nodes, ignored for master nodes.
- 4) LinIfMainFunctionPeriod
 - Defines the interval of calls to main functions per channel in seconds.
- 5) LinIfMaxFrameCnt
 - Maximum number of Frames.
 - This parameter is needed only in case of post-build loadable implementation using static memory allocation.

- 6) LinIfScheduleChangeNextTimeBase
 - Enables/disables the switch to a new schedule table at the start of the next time base after status check.
 - True: LinIf selects a new schedule table in next main function.
 - Only applicable for LIN Master nodes.
- 7) LinIfScheduleRequestConfirmationUL
 - This parameter defines the upper layer (UL) module to which the confirmation of the successfully performed schedule table change shall be sent.
 - Only applicable to LIN master nodes.
- 8) LinIfWakeupConfirmationUL
 - This parameter defines the upper layer (UL) module to which the confirmation of the wake-up shall be sent.
- 9) LinIfCddRef
 - Reference to the CDD module description.
 - This parameter is only required when LinIfWakeupConfirmationUL, LinIfScheduleRequestConfirmationUL, LinIfGotoSleepConfirmationUL and/or LinIfGotoSleepIndicationUL is set to CDD.
- 10) LinIfChannelRef
 - Reference to the channel definition in the LIN driver.
- 11) LinIfComMNetworkHandleRef
 - Unique handle to identify one LIN network.
 - Reference to one of the network handles configured for the ComM.

6.3 LinIfGlobalConfig-LinIfChannel-LinIfFrame

Parameter Name	Value	Category
¹⁾ LinIfChecksumType	Automated	F
²⁾ LinIfFrameId	Automated	F
³⁾ LinIfFrameIndex	Automated	N
⁴⁾ LinIfFrameType	Automated	F

- 1) LinIfChecksumType
 - Type of checksum that the frame is using.
 - This parameter is optional because in case of sporadic frames it should not be set.
- 2) LinIfFrameId
 - ID of the LIN frame. The Protected ID including parity is calculated by the generation tool.
- 3) LinIfFrameIndex
 - PID index of the frame.
- 4) LinIfFrameType
 - This parameter defines the type of frame (e.g. sporadic frame).
 - For master nodes, all frame types are permitted.
 - A sporadic slot may be used by a set of unconditional frames in the role of substitution frames.
 - For slave nodes, only following types are permitted: Unconditional, MRF, SRF, Event-triggered.
 - An event-triggered slot may be used by a set of unconditional frames in the role of substitution frames.

6.4 LinIfGlobalConfig-LinIfChannel-LinIfFrame-LinIfFixedFrameSdu

Parameter Name	Value	Category
¹⁾ LinIfFixedFrameSduByte	0 ... 7	N
²⁾ LinIfFixedFrameSduByteVal	0 ... 255	N

- 1) LinIfFixedFrameSduByte
 - Index of the Byte in the SDU (response) 8 byte array.
- 2) LinIfFixedFrameSduByteVal
 - Byte value in the SDU (response) 8-byte array.

6.5 LinIfGlobalConfig-LinIfChannel-LinIfFrame-LinIfSubstitutionFrames

Parameter Name	Value	Category
¹⁾ LinIfFramePriority	0 ... 255	N
²⁾ LinIfSubstitutionFrameRef	-	N

- 1) LinIfFramePriority
 - Priority of sporadic frame in a master node or of event-triggered frame in slave node.
- 2) LinIfSubstitutionFrameRef
 - Reference to an unconditional Frame that is used as sporadic frame in a master node or event-triggered frame in a slave node.

6.6 LinIfGlobalConfig-LinIfChannel-LinIfFrame-LinIfPduDirection-LinIfRxPdu

Parameter Name	Value	Category
¹⁾ LinIfRxIndicationUL	-	N
²⁾ LinIfUserRxIndicationUL	PDUR	F
³⁾ LinIfRxPduRef	Automated	F

- 1) LinIfRxIndicationUL
 - This parameter defines the name of the <User_RxIndication>.
 - This parameter depends on the parameter LinIfUserRxIndicationUL.
 - If LinIfUserRxIndicationUL equals PDUR, the name of the <User_RxIndication> is fixed.
 - If LinIfUserRxIndicationUL equals CDD, the name of the <User_RxIndication> is selectable.
- 2) LinIfUserRxIndicationUL
 - This parameter defines the upper layer (UL) module to which the indication of the successfully received LinIfRxPdu has to be routed via <User_RxIndication>.
- 3) LinIfRxPduRef
 - Reference to the PDU that is received in this frame.

6.7 LinIfGlobalConfig-LinIfChannel-LinIfFrame-LinIfPduDirection-LinIfTxPdu

Parameter Name	Value	Category
¹⁾ LinIfTxConfirmationUL	-	N
²⁾ LinIfTxPduId	Automated	F
³⁾ LinIfTxTriggerTransmitUL	-	N
⁴⁾ LinIfUserTxUL	PDUR	F
⁵⁾ LinIfTxPduRef	Automated	F

- 1) LinIfTxConfirmationUL

- This parameter defines the name of the <User_TxConfirmation>.
 - This parameter depends on the parameter LinIfUserTxUL.
 - If LinIfUserTxUL equals PDUR, the name of the <User_TxConfirmation> is fixed.
 - If LinIfUserTxUL equals CDD, the name of the <User_TxConfirmation> is selectable.
- 2) LinIfTxPduId
- Identifier of the Pdu for the upper layer.
- 3) LinIfTxTriggerTransmitUL
- This parameter defines the name of the <User_TriggerTransmit>.
 - This parameter depends on the parameter LinIfUserTxUL.
 - If LinIfUserTxUL equals PDUR, the name of the <User_TriggerTransmit> is fixed.
 - If LinIfUserTxUL equals CDD, the name of the <User_TriggerTransmit> is selectable.
- 4) LinIfUserTxUL
- This parameter defines the upper layer (UL) module to which the trigger of the transmitted LinTxPdu (via the <User_TriggerTransmit>) or the confirmation of the successfully transmitted LinTxPdu has to be routed (via the <User_TxConfirmation>).
- 5) LinIfTxPduRef
- Reference to the PDU that is transmitted in this frame.

6.8 LinIfGlobalConfig-LinIfChannel-LinIfScheduleTable

Parameter Name	Value	Category
¹⁾ LinIfResumePosition	(Ref to the description below)	C
²⁾ LinIfRunMode	(Ref to the description below)	C
³⁾ LinIfScheduleTableIndex	Automated	F

- 1) LinIfResumePosition
- Defines where a RUN_CONTINUOUS schedule table shall proceed in case it has been interrupted by a RUN_ONCE table.
 - CONTINUE_AT_IT_POINT : Continue schedule table where it was interrupted.
 - START_FROM_BEGINNING : Start schedule table from the beginning.
- 2) LinIfRunMode
- The schedule table can be executed in two different modes.
 - RUN_CONTINUOUS
 - RUN_ONCE
- 3) LinIfScheduleTableIndex
- This is the unique index used by upper layers to identify a schedule.
 - Note that the NULL_SCHEDULE for each channel must have index 0.

6.9 LinIfGlobalConfig-LinIfChannel-LinIfScheduleTable-LinIfEntry

Parameter Name	Value	Category
¹⁾ LinIfDelay	Automated	F
²⁾ LinIfEntryIndex	Automated	F
³⁾ LinIfCollisionResolvingRef	-	N
⁴⁾ LinIfFrameRef	Automated	F

- 1) LinIfDelay

- Delay to next entry in schedule table in seconds.
- 2) LinIfEntryIndex
 - Position of the Frame Entry in the Schedule Table. The first entry index in the schedule table is 0.
- 3) LinIfCollisionResolvingRef
 - Reference to the schedule table, which resolves the collision.
 - This parameter is only used if the referenced frames are event triggered frames.
- 4) LinIfFrameRef
 - Reference to the frames that belong to this schedule table entry.

6.10 LinIfGlobalConfig-LinIfChannel-LinIfNodeType-LinIfMaster

Parameter Name	Value	Category
¹⁾ LinIfJitter	Automated	N

- 1) LinIfJitter
 - The jitter specifies the differences between the maximum and minimum delay from time base tick to the header sending start point in seconds.

6.11 LinIfGlobalConfig-LinIfChannel-LinIfNodeType-LinIfSlave

Parameter Name	Value	Category
¹⁾ LinIfLinProtocolVersion	Automated	N
²⁾ LinIfResponseErrorSignal	Automated	N

- 1) LinIfLinProtocolVersion
 - Defines the LIN protocol version of the slave node.
 - This information is relevant for the LIN conformance test execution.
- 2) LinIfResponseErrorSignal
 - Reference to the response_error signal.
 - Mandatory for all LIN 2.x and ISO LIN slave nodes, not relevant for LIN 1.3 slave nodes.

6.12 LinIfGlobalConfig-LinIfChannel-LinIfNodeType-LinIfSlave-LinIfNodeConfigurationIdentification

Parameter Name	Value	Category
¹⁾ LinIfConfiguredNAD	Automated	N
²⁾ LinIfFunctionId	Automated	N
³⁾ LinIfInitialNAD	Automated	N
⁴⁾ LinIfNasTimeout	Automated	N
⁵⁾ LinIfSupplierId	Automated	N
⁶⁾ LinIfVariantId	Automated	N

- 1) LinIfConfiguredNAD
 - Slave node configured NAD.
- 2) LinIfFunctionId

- LIN function Id.
- 3) LinIfInitialNAD
 - Slave node initial NAD.
- 4) LinIfNasTimeout
 - N_As timeout in seconds.
- 5) LinIfSupplierId
 - LIN consortium or ISO LIN supplier Id.
- 6) LinIfVariantId
 - LIN variant Id.

6.13 LinIfGlobalConfig-LinIfChannel-LinIfFrame-LinIfPduDirection-LinIfSlaveToSlavePdu

Parameter Name	Value	Category
-	-	-

Represents a slave-to-slave PDU/frame. Master does only send the header but doesn't receive the response. Only relevant for master nodes.

6.14 LinIfGlobalConfig-LinIfChannel-LinIfFrame-LinIfPduDirection-LinIfInternalPdu

Parameter Name	Value	Category
-	-	-

Represents a Diagnostic or Configuration frame : no Message ID (no PduId). Only applicable to LIN master nodes.

6.15 LinIfGlobalConfig-LinIfChannel-LinIfTransceiverDrvConfig

Parameter Name	Value	Category
¹⁾ LinIfTrcvIdRef	-	C

- 1) LinIfTrcvIdRef
 - Logical handle of the underlying LIN transceiver to be served by the LIN Interface.

6.16 LinTpGeneral

Parameter Name	Value	Category
¹⁾ LinTpChangeParameterApi	Automated	C
²⁾ LinTpVersionInfoApi	Automated	C

- 1) LinTpChangeParameterApi
 - This parameter, if set to true, enables the LinTp_ChangeParameter Api for this Module.
- 2) LinTpVersionInfoApi
 - Switches the LinTp_GetVersionInfo function ON or OFF.

6.17 LinTpGlobalConfig

Parameter Name	Value	Category
¹⁾ LinTpMaxRxNSduCnt	Automated	C
²⁾ LinTpMaxTxNSduCnt	Automated	C

- 1) LinTpMaxRxNSduCnt
 - Maximum number of NSdus.
 - This parameter is needed only in case of post-build loadable implementation using static memory allocation.
- 2) LinTpMaxTxNSduCnt
 - Maximum number of NSdus.
 - This parameter is needed only in case of post-build loadable implementation using static memory allocation.

6.18 LinTpGlobalConfig-LinTpChannelConfig

Parameter Name	Value	Category
¹⁾ LinTpDropNotRequestedNad	Automated	C
²⁾ LinTpMaxNumberOfRespPendingFrames	Automated	C
³⁾ LinTpP2Max	Automated	C
⁴⁾ LinTpP2Timing	Automated	C
⁵⁾ LinTpScheduleChangeDiag	Automated	C
⁶⁾ LinTpChannelRef	Automated	C

- 1) LinTpDropNotRequestedNad
 - Configures if TP Frames of not requested LIN-Slaves are dropped or not.
 - TRUE: Drop TP Frames of not requested LIN-Slaves
 - FALSE: Keep TP Frames of not requested LIN-Slaves
 - Only used for LIN Master nodes, ignored for slave nodes.
- 2) LinTpMaxNumberOfRespPendingFrames
 - Configures the maximum number of allowed response pending frames.
 - Only used for LIN Master nodes, ignored for slave nodes.
- 3) LinTpP2Max
 - P2*max timeout when a response pending frame is expected in seconds.
 - Note that the minimum value of LinTpP2Max shall be more than or equal to the value of LinTpP2Timing. Only used for LIN Master nodes, ignored for slave nodes.
- 4) LinTpP2Timing
 - Definition of the P2max timeout observation parameter in seconds.
 - Only used for LIN Master nodes, ignored for slave nodes.
- 5) LinTpScheduleChangeDiag
 - Enables or disables the call of BswM_LinTp_RequestMode() to diagnostic request/response schedule.
 - false: BswM is not called
 - true: BswM is called Only used for LIN Master nodes, ignored for slave nodes.
- 6) LinTpChannelRef
 - Index of the channel this LinTp channel belongs to.

6.19 LinTpGlobalConfig-LinTpRxNSdu

Parameter Name	Value	Category
¹⁾ LinTpNcr	Automated	C

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
18 / 47

²⁾ LinTpRxNSduId	Automated	C
³⁾ LinTpRxNSduNad	Automated	C
⁴⁾ LinTpRxNSduChannelRef	Automated	C
⁵⁾ LinTpRxNSduPduRef	Automated	C

- 1) LinTpNcr
 - Value in seconds of the N_Cr timeout.
 - N_Cr is the time until reception of the next Consecutive Frame N_PDU.
- 2) LinTpRxNSduId
 - The identifier of the Transport Protocol message.
 - This ID will be used by upper layers to call LinTp_ChangeParameter.
- 3) LinTpRxNSduNad
 - A N-SDU transported on LIN is identified using the NAD for the specific slave.
- 4) LinTpRxNSduChannelRef
 - Index of the channel this N-SDU belongs to.
- 5) LinTpRxNSduPduRef
 - Reference to the global PDU

6.20 LinTpGlobalConfig-LinTpTxNSdu

Parameter Name	Value	Category
¹⁾ LinTpMaxBufReq	Automated	C
²⁾ LinTpNas	Automated	C
³⁾ LinTpNcs	Automated	C
⁴⁾ LinTpTxNSduId	Automated	C
⁵⁾ LinTpTxNSduNad	Automated	C
⁶⁾ LinTpTxNSduChannelRef	Automated	C
⁷⁾ LinTpTxNSduPduRef	Automated	C

- 1) LinTpMaxBufReq
 - This parameter defines the maximum number of times the LinTp should request upper layer for the Tx Buffer.
 - It is also used to limit the number of retries for PduR_LinTpCopyTxData when no timer is active.
- 2) LinTpNas
 - Value in seconds of the N_As timeout.
 - N_As is the time for transmission of a LIN frame (any N_PDU) on the part of the sender.
- 3) LinTpNcs
 - Value in seconds of the performance requirement of N_Cs.
 - N_Cs is the time which elapses between the transmit request of a CF N-PDU until the transmit request of the next CF N-PDU.
- 4) LinTpTxNSduId
 - The identifier of the Transport Protocol message.
 - This ID will be the one that is communicated with upper layers.
- 5) LinTpTxNSduNad
 - A N-SDU transported on LIN is identified using the NAD for the specific slave.
- 6) LinTpTxNSduChannelRef
 - Index of the channel this N-SDU belongs to.
- 7) LinTpTxNSduPduRef
 - Reference to the global PDU

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
19 / 47

7 Application Programming Interface (API)

7.1 Type Definitions

7.1.1 LinIf_SchHandleType

This type is only applicable for LIN master nodes.

Type:	uint8		
Range:	NULL_SCHEDULE	0x00	The NULL_SCHEDULE.
	range	1..255	Index of the schedule table that is selectable and followed by LIN Interface. Value is unique per LIN channel/controller, but not per ECU.
Description:	Index of the schedule table that is selectable and followed by LIN Interface. Value is unique per LIN channel/controller, but not per ECU. The number of schedule tables is limited to 255.		
Available via:	LinIf.h		

7.1.2 LinIf_ConfigType

Type:	Structure	
Range:	implementation specific	--
Description:	A pointer to an instance of this structure will be used in the initialization of the LIN Interface. The outline of the structure is defined in chapter 10 Configuration Specification.	
Available via:	Linf.h	

7.1.3 LinTp_ConfigType

Type:	Structure	
Range:	implementation specific	--
Description:	This is the base type for the configuration of the LIN Transport Protocol A pointer to an instance of this structure will be used in the initialization of the LIN Transport	

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
21 / 47

	Protocol. The outline of the structure is defined in chapter 10 Configuration Specification.
Available via:	LinIf.h

7.1.4 LinTp_Mode

This type is only applicable for LIN master nodes.

Type:	Enumeration		
Range:	LINTP_APPLICATIVE_SCHEDULE	--	Applicative schedule is selected.
	LINTP_DIAG_REQUEST	--	Master request schedule table is selected.
	LINTP_DIAG_RESPONSE	--	Slave response schedule table is selected
Description:	This type denotes which Schedule table can be requested by LIN TP during diagnostic session.		
Available via:	LinIf.h		

7.2 Macro Constants

None

7.3 Functions

Describes all functionalities of modules, group by feature or listed following SWS chapter 8

7.3.1 LinIf_Init

Function Name	LinIf_Init
Syntax	void LinIf_Init(const LinIf_ConfigType* ConfigPtr)
Service ID [Hex]	0x01
Sync/Async	Synchronous
Reentrancy	Non-Reentrant

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
22 / 47

Parameters (In)	ConfigPtr	Pointer to the LIN Interface configuration
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	Initializes the LIN Interface. This function is used by BSW.	
Available via	LinIf.h	

7.3.2 LinIf_GetVersionInfo

Function Name	LinIf_GetVersionInfo	
Syntax	void LinIf_GetVersionInfo(Std_VersionInfoType* versioninfo)	
Service ID [Hex]	0x03	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	None	
Parameters (Inout)	None	
Parameters (Out)	versioninfo	Pointer to where to store the version information of this module.
Return Value	None	
Description	Returns the version information of this module. This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Available via	LinIf.h	

7.3.3 LinIf_Transmit

Function Name	LinIf_Transmit	
Syntax	Std_ReturnType LinIf_Transmit(PduIdType TxPduId, const PduInfoType* PduInfoPtr)	

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
23 / 47

Service ID [Hex]	0x49	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (In)	TxPduId	Identifier of the PDU to be transmitted
	PduInfoPtr	Length of and pointer to the PDU data and pointer to MetaData.
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	E_OK: Transmit request has been accepted. E_NOT_OK: Transmit request has not been accepted.
Description	Requests transmission of a PDU. This function is used by BSW.	
Available via	LinIf.h	

7.3.4 LinIf_ScheduleRequest

Function Name	LinIf_ScheduleRequest	
Syntax	Std_ReturnType LinIf_ScheduleRequest(NetworkHandleType Channel, LinIf_SchHandleType Schedule)	
Service ID [Hex]	0x05	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	Channel index.
	Schedule	Identification of the new schedule to be set.
Parameters (Inout)	None	
Parameters (Out)	mode	Returns the active mode, see ComM_ModeType for descriptions of the modes.
Return Value	Std_ReturnType	E_OK: Schedule table request has been accepted. E_NOT_OK: Schedule table switch request has not been accepted due to one of the following reasons:

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
24 / 47

		<ul style="list-style-type: none"> - LIN Interface has not been initialized - referenced channel does not exist (identification is out of range) - referenced schedule table does not exist (identification is out of range) - State is sleep
Description	<p>Requests a schedule table to be executed.</p> <p>Only used for LIN master nodes.</p> <p>This function is used by user. But it needs configuration. (It cannot be called directly by user)</p>	
Available via	LinIf.h	

7.3.5 LinIf_GotoSleep

Function Name	LinIf_GotoSleep	
Syntax	Std_ReturnType LinIf_GotoSleep(NetworkHandleType Channel)	
Service ID [Hex]	0x06	
Sync/Async	ASynchronous	
Reentrancy	Non Reentrant	
Parameters (In)	Channel	Identification of the LIN channel.
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	<p>E_OK: Request to go to sleep has been accepted or sleep transition is already in progress or controller is already in sleep state</p> <p>E_NOT_OK: Request to go to sleep has not been accepted due to one or more of the following reasons:</p> <ul style="list-style-type: none"> - LIN Interface has not been initialized - referenced channel does not exist (identification is out of range)

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
25 / 47

Description	Initiates a transition into the Sleep Mode on the selected channel. This function is used by BSW.
Available via	LinIf.h

7.3.6 LinIf_Wakeup

Function Name	LinIf_Wakeup	
Syntax	Std_ReturnType LinIf_Wakeup(NetworkHandleType Channel)	
Service ID [Hex]	0x07	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	Identification of the LIN channel.
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	<p>E_OK: Request to wake up has been accepted or the controller is not in sleep state.</p> <p>E_NOT_OK: Request to wake up has not been accepted due to one or more of the following reasons:</p> <ul style="list-style-type: none"> - LIN Interface has not been initialized - referenced channel does not exist (identification is out of range) - Lin_Wakeup has returned E_NOT_OK - Lin_WakeupInternal has returned E_NOT_OK
Description	Initiates the wake up process. This function is used by BSW.	
Available via	LinIf.h	

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
26 / 47

7.3.7 LinIf_SetTrcvMode

Function Name	LinIf_SetTrcvMode	
Syntax	Std_ReturnType LinIf_SetTrcvMode(NetworkHandleType Channel, LinTrcv_TrvcModeType TransceiverMode)	
Service ID [Hex]	0x08	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	Identification of the LIN channel.
	TransceiverMode	<p>E_OK: Will be returned, if the transceiver state has been changed to the requested mode.</p> <p>E_NOT_OK: Will be returned, if the transceiver state change has failed or the parameter is out of the allowed range. The previous state has not been changed.</p>
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	<p>E_OK: Request to wake up has been accepted or the controller is not in sleep state.</p> <p>E_NOT_OK: Request to wake up has not been accepted due to one or more of the following reasons:</p> <ul style="list-style-type: none"> - LIN Interface has not been initialized - referenced channel does not exist (identification is out of range) - Lin_Wakeup has returned E_NOT_OK - Lin_WakeupInternal has returned E_NOT_OK
Description	<p>Set the given LIN transceiver to the given mode.</p> <p>This function is used by BSW.</p>	
Available via	LinIf.h	

7.3.8 LinIf_GetTrcvMode

Function Name	LinIf_GetTrcvMode	
Syntax	Std_ReturnType LinIf_GetTrcvMode(NetworkHandleType Channel, LinTrcv_TrcvModeType* TransceiverModePtr)	
Service ID [Hex]	0x09	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	Identification of the LIN channel
Parameters (Inout)	None	
Parameters (Out)	TransceiverModePtr	Pointer to a memory location where output value will be stored.
Return Value	Std_ReturnType	<p>E_OK: The call of the LIN Transceiver Driver's API service has returned E_OK.</p> <p>E_NOT_OK: The call of the LIN Transceiver Driver's API service has returned E_NOT_OK or channel parameter is invalid or pointer is NULL.</p>
Description	<p>Returns the actual state of a LIN Transceiver Driver.</p> <p>This function is used by user. But it needs configuration. (It cannot be called directly by user)</p>	
Available via	LinIf.h	

7.3.9 LinIf_GetTrcvWakeupReason

Function Name	LinIf_GetTrcvWakeupReason	
Syntax	Std_ReturnType LinIf_GetTrcvWakeupReason(NetworkHandleType Channel, LinTrcv_TrcvWakeupReasonType* TrcvWuReasonPtr)	
Service ID [Hex]	0x0a	
Sync/Async	Synchronous	
Reentrancy	Reentrant	

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
28 / 47

Parameters (In)	Channel	Identification of the LIN channel
Parameters (Inout)	None	
Parameters (Out)	TrcvWuReasonPtr	Pointer to a memory location where output value will be stored.
Return Value	Std_ReturnType	E_OK: The call of the LIN Transceiver Driver's API service has returned E_OK. E_NOT_OK: The call of the LIN Transceiver Driver's API service has returned E_NOT_OK or channel parameter is invalid or pointer is NULL.
Description	Returns the reason for the wake up that has been detected by the LIN Transceiver Driver. This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Available via	LinIf.h	

7.3.10 LinIf_SetTrcvWakeupMode

Function Name	LinIf_SetTrcvWakeupMode	
Syntax	Std_ReturnType LinIf_SetTrcvWakeupMode(NetworkHandleType Channel, LinTrcv_TrvcWakeupModeType LinTrcvWakeupMode)	
Service ID [Hex]	0x0b	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	Identification of the LIN channel
	LinTrcvWakeupMode	Requested transceiver wake up reason.
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	E_OK: The call of the LIN Transceiver Driver's API service has returned E_OK. E_NOT_OK: The call of the LIN Transceiver Driver's API service has returned E_NOT_OK or channel or mode parameter is invalid.

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
29 / 47

Description	<p>This API enables, disables and clears the notification for wakeup events on the addressed network.</p> <p>This function is used by user. But it needs configuration. (It cannot be called directly by user)</p>
Available via	LinIf.h

7.3.11 LinIf_GetPIDTable

Function Name	LinIf_GetPIDTable	
Syntax	Std_ReturnType LinIf_GetPIDTable(NetworkHandleType Channel, Lin_FramePidType* PidBuffer, uint8* PidBufferLength)	
Service ID [Hex]	0x72	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	Identification of the LIN channel.
Parameters (Inout)	PidBuffer	Pointer to existing buffer to which the current assigned PID values are copied to
	PidBufferLength	Pointer to actual length of provided buffer. After successful return, it contains the number of copied PID values.
Parameters (Out)	None	
Return Value	Std_ReturnType	<p>E_OK: Request has been accepted.</p> <p>E_NOT_OK: Request has not been accepted, development or production error occurred.</p>
Description	<p>Retrieves all assigned PID values. The order is congruent to the LIN frame index. Only applicable for LIN slave nodes.</p> <p>This function is used by user. But it needs configuration. (It cannot be called directly by user)</p>	
Available via	LinIf.h	

7.3.12 LinIf_SetPIDTable

Function Name	LinIf_SetPIDTable	
Syntax	Std_ReturnType LinIf_SetPIDTable(NetworkHandleType Channel, Lin_FramePidType* PidBuffer, uint8 PidBufferLength)	
Service ID [Hex]	0x73	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	Identification of the LIN channel.
	PidBuffer	Pointer to buffer which contains the PID values to configure.
	PidBufferLength	Number of PID values in the provided buffer
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	E_OK: Request has been accepted.
		E_NOT_OK: Request has not been accepted, development or production error occurred.
Description	<p>Sets all assigned PID values. The order is congruent to the LIN frame index. Only applicable for LIN slave nodes.</p> <p>This function is used by user. But it needs configuration. (It cannot be called directly by user)</p>	
Available via	LinIf.h	

7.3.13 LinIf_GetConfiguredNAD

Function Name	LinIf_GetConfiguredNAD	
Syntax	Std_ReturnType LinIf_GetConfiguredNAD(NetworkHandleType Channel, uint8* Nad)	
Service ID [Hex]	0x70	
Sync/Async	Synchronous	

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
31 / 47

Reentrancy	Reentrant	
Parameters (In)	Channel	Identification of the LIN channel.
Parameters (Inout)	None	
Parameters (Out)	Nad	Configured NAD of slave
Return Value	Std_ReturnType	E_OK: Request has been accepted. E_NOT_OK: Request has not been accepted, development or production error occurred.
Description	Reports the current configured NAD. Only applicable for LIN slave nodes. This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Available via	LinIf.h	

7.3.14 LinIf_SetConfiguredNAD

Function Name	LinIf_SetConfiguredNAD	
Syntax	Std_ReturnType LinIf_SetConfiguredNAD(NetworkHandleType Channel, uint8 Nad)	
Service ID [Hex]	0x71	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	Identification of the LIN channel.
	Nad	Configured NAD to set as new slave NAD
Parameters (Inout)	None	
Parameters (Out)	Nad	Configured NAD of slave
Return Value	Std_ReturnType	E_OK: Request has been accepted. E_NOT_OK: Request has not been accepted, development or production error occurred.
Description	Sets the current configured NAD. Only applicable for LIN slave nodes.	

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
32 / 47

	This function is used by user. But it needs configuration. (It cannot be called directly by user)
Available via	LinIf.h

7.3.15 LinTp_Init

Function Name	LinTp_Init	
Syntax	void LinTp_Init(const LinTp_ConfigType* ConfigPtr)	
Service ID [Hex]	0x40	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	ConfigPtr	Pointer to the LIN Transport Protocol configuration.
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	Initializes the LIN Transport Layer. This function is used by BSW.	
Available via	LinIf.h	

7.3.16 LinTp_Transmit

Function Name	LinTp_Transmit	
Syntax	Std_ReturnType LinTp_Transmit(PduldType TxPduld, const PduInfoType* PduInfoPtr)	
Service ID [Hex]	0x49	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different Pdulds. Non reentrant for the same Pduld.	
Parameters (In)	TxPduld	Identifier of the PDU to be transmitted
	PduInfoPtr	Length of and pointer to the PDU data and pointer to MetaData.

User Manual

DOCUMENT NUMBER
(DOC NO)SHT/SHTS
33 / 47

Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	E_OK: Transmit request has been accepted. E_NOT_OK: Transmit request has not been accepted.
Description	Requests transmission of a PDU. This function is used by BSW.	
Available via	LinIf.h	

7.3.17 LinTp_GetVersionInfo

Function Name	LinTp_GetVersionInfo	
Syntax	void LinTp_GetVersionInfo(Std_VersionInfoType* versioninfo)	
Service ID [Hex]	0x42	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	None	
Parameters (Inout)	None	
Parameters (Out)	versioninfo	Pointer to where to store the version information of this module.
Return Value	None	
Description	Returns the version information of this module. This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Available via	LinIf.h	

7.3.18 LinTp_Shutdown

Function Name	LinTp_Shutdown	
Syntax	void LinTp_Shutdown(void)	

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
34 / 47

Service ID [Hex]	0x43
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	None
Return Value	None
Description	Shutowns the LIN TP. This function is used by BSW.
Available via	LinIf.h

7.3.19 LinTp_ChangeParameter

Function Name	LinTp_ChangeParameter	
Syntax	Std_ReturnType LinTp_ChangeParameter(PduldType id, TPPParameterType parameter, uint16 value)	
Service ID [Hex]	0x4b	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	id	Identification of the PDU which the parameter change shall affect.
	parameter	ID of the parameter that shall be changed.
	value	The new value of the parameter.
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	E_OK: The parameter was changed successfully. E_NOT_OK: The parameter change was rejected.
Description	Request to change a specific transport protocol parameter (e.g. block size).	

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
35 / 47

	This function is used by user. But it needs configuration. (It cannot be called directly by user)
Available via	LinIf.h

7.3.20 LinIf_CheckWakeup

Function Name	LinIf_CheckWakeup	
Syntax	Std_ReturnType LinIf_CheckWakeup(EcuM_WakeupSourceType WakeupSource)	
Service ID [Hex]	0x60	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	WakeupSource	Source device, which initiated the wakeup event: LIN controller or LIN transceiver.
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	E_OK: No error has occurred during execution of the API E_NOT_OK: An error has occurred during execution of the API or invalid WakeupSource
Description	<p>Will be called when the EcuM has been notified about a wakeup on a specific LIN channel.</p> <p>This function is used by user. But it needs configuration. (It cannot be called directly by user)</p>	
Available via	LinIf.h	

7.3.21 LinIf_EnableBusMirroring

Function Name	LinIf_EnableBusMirroring	
Syntax	Std_ReturnType LinIf_EnableBusMirroring(NetworkHandleType Channel, boolean MirroringActive)	
Service ID [Hex]	0x7f	
Sync/Async	Synchronous	

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
36 / 47

Reentrancy	Reentrant	
Parameters (In)	Channel	Identification of the LIN channel.
	MirroringActive	TRUE: Mirror_ReportLinFrame will be called for each frame received or transmitted on the given channel. FALSE: Mirror_ReportLinFrame will not be called for the given channel.
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	E_OK: Mirroring mode was changed. E_NOT_OK: Wrong Channel, or mirroring globally disabled (see LinIfBusMirroringSupport).
Description	Enables or disables mirroring for a LIN channel. This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Available via	LinIf.h	

7.3.22 LinIf_WakeupConfirmation

Function Name	LinIf_WakeupConfirmation	
Syntax	void LinIf_WakeupConfirmation(EcuM_WakeupSourceType WakeupSource)	
Service ID [Hex]	0x61	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	WakeupSource	Source device which initiated the wakeup event: LIN controller or LIN transceiver
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
37 / 47

Description	The LIN Driver or LIN Transceiver Driver will call this function to report the wake up source after the successful wakeup detection during CheckWakeup or after power on by bus. This function is used by BSW.
Available via	LinIf.h

7.3.23 LinIf_HeaderIndication

Function Name	LinIf_HeaderIndication	
Syntax	Std_ReturnType LinIf_HeaderIndication(NetworkHandleType Channel, Lin_PduType* PduPtr)	
Service ID [Hex]	0x78	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different Channels. Non reentrant for the same Channel.	
Parameters (In)	Channel	Identification of the LIN channel
Parameters (Inout)	PduPtr	Pointer to PDU providing the received PID and pointer to the SDU data buffer as in parameter. Upon return, the length, checksum type and frame response type are received as out parameter. If the frame response type is LIN_FRAMERESPONSE_TX, then the SDU data buffer contains the transmission data.
Parameters (Out)	None	
Return Value	Std_ReturnType	E_OK: Request has been accepted. E_NOT_OK: Request has not been accepted, development or production error occurred.
Description	The LIN Driver will call this function to report a received LIN header. This function is only applicable for LIN slave nodes (available only if the ECU has any LIN slave channel). This function is used by BSW.	
Available via	LinIf.h	

7.3.24 LinIf_RxIndication

Function Name	LinIf_RxIndication
----------------------	--------------------

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
38 / 47

Syntax	void LinIf_RxIndication(NetworkHandleType Channel, uint8* Lin_SduPtr)	
Service ID [Hex]	0x77	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different Channels. Non reentrant for the same Channel.	
Parameters (In)	Channel	Identification of the LIN channel
	Lin_SduPtr	Pointer to pointer to a shadow buffer or memory mapped LIN Hardware receive buffer where the current SDU is stored. This pointer is only valid if the response is received.
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	<p>The LIN Driver will call this function to report a successfully received response and provides the reception data to the LIN Interface. This function is only applicable for LIN slave nodes (available only if the ECU has any LIN slave channel).</p> <p>This function is used by BSW.</p>	
Available via	LinIf.h	

7.3.25 LinIf_TxConfirmation

Function Name	LinIf_TxConfirmation	
Syntax	void LinIf_TxConfirmation(NetworkHandleType Channel)	
Service ID [Hex]	0x7a	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different Channels. Non reentrant for the same Channel.	
Parameters (In)	Channel	Identification of the LIN channel
Parameters (Inout)	None	
Parameters (Out)	None	

User Manual

DOCUMENT NUMBER
(DOC NO)

SHT/SHTS
39 / 47

Return Value	None
Description	The LIN Driver will call this function to report a successfully transmitted response. This function is only applicable for LIN slave nodes (available only if the ECU has any LIN slave channel). This function is used by BSW.
Available via	LinIf.h

7.3.26 LinIf_LinErrorIndication

Function Name	LinIf_LinErrorIndication	
Syntax	void LinIf_LinErrorIndication(NetworkHandleType Channel, Lin_SlaveErrorType ErrorStatus)	
Service ID [Hex]	0x7b	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different Channels. Non reentrant for the same Channel.	
Parameters (In)	Channel	Identification of the LIN channel
	ErrorStatus	Type of detected error
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	The LIN Driver will call this function to report a successfully transmitted response. This function is only applicable for LIN slave nodes (available only if the ECU has any LIN slave channel). This function is used by BSW.	
Available via	LinIf.h	

7.3.27 LinIf_MainFunction_<LinIfChannel.ShortName>

Function Name	LinIf_MainFunction_<LinIfChannel.ShortName>
Syntax	void LinIf_MainFunction_<LinIfChannel.ShortName>(void)
Service ID [Hex]	0x80

Description	The main processing function of the LIN Interface. This function is used by BSW.
Available via	SchM_LinIf.h

7.3.28 Rte_Call_CallbackError_lin_cbk_err

Function Name	Rte_Call_CallbackError_lin_cbk_err
Syntax	FUNC(Std_ReturnType, RTE_CODE) Rte_Call_<>_LIN_CallbackError_lin_cbk_err(VAR(uint8, AUTOMATIC) channelId, VAR(uint8, AUTOMATIC) errorcode, VAR(uint8, AUTOMATIC) schEntryNum, VAR(uint8, AUTOMATIC) frameId)
Service ID [Hex]	-
Description	Notification function invoked when a LIN communication error occurs in LinIf. This function is used by user. But it needs configuration. (It cannot be called directly by user)
Available via	Rte_<>_LIN.h

7.3.29 Rte_Call_CallbackAfterSchedule_lin_after_schedule

Function Name	Rte_Call_CallbackAfterSchedule_lin_after_schedule
Syntax	FUNC(Std_ReturnType, RTE_CODE) Rte_Call_<>_LIN_CallbackAfterSchedule_lin_after_schedule(VAR(uint8, AUTOMATIC) channelId, VAR(uint8, AUTOMATIC) schId, VAR(uint8, AUTOMATIC) schEntryNum, VAR(uint8, AUTOMATIC) frameId)
Service ID [Hex]	-
Description	Notification function invoked just before transitioning to the next schedule in LinIf. This function is used by user. But it needs configuration. (It cannot be called directly by user)
Available via	Rte_<>_LIN.h

8 Generator

8.1 Generator Option

Options	Description
-G,--Generation	Symbolic parameters to be used for fore generation (skip validation).
-H,--Help	Display this help message.
-I,--Input <I>	ECU description file path of the module for which generation tool need to run.
-L,--Log	Symbolic parameters to be used for generation error log.
-M,--Module <M>	Specify module name and version to be generated code for.
-O,--Output <O>	Project-relative path to location where the generated code is to be placed.
-T,--Top_path <T>	Symbolic parameters to be used for set path of module.
-V,--Validate	Symbolic parameters to be used for invoking validation checks.

8.2 Generator Error Message

8.2.1 Error Messages

ERR_LinIf_0621001: Channel {0} has LinIfWakeupConfirmationUL, LinIfScheduleRequestConfirmationUL, LinIfGotoSleepConfirmationUL or LinIfGotoSleepIndicationUL, is configured as CDD while LinIfCddRef/LinIfPublicCddHeaderFile is empty

This error occurs:

LinIfCddRef or LinIfPublicCddHeaderFile is not set while one of the following parameters is set as CDD:

- LinIfWakeupConfirmationUL.
- LinIfScheduleRequestConfirmationUL.
- LinIfGotoSleepConfirmationUL.
- LinIfGotoSleepIndicationUL.

ERR_LinIf_0621002: There are more than one {0} in LinIf/LinIfGlobalConfig/LinIfChannel

This error occurs:

- any channel has duplicate properties, such as (LinIfChannelRef, LinIfComMNetworkHandleRef).

ERR_Linlf_0621003: In LinlfGeneral, LinlfTrcvDriverSupported is set (FALSE) but LinlfMultipleTrcvDriverSupported is set (TRUE)

This error occurs:

- parameter LinlfTrcvDriverSupported of the container LinlfGeneral is false but LinlfMultipleTrcvDriverSupported is set TRUE.

ERR_Linlf_0621004: In LinlfGeneral, LinlfTrcvDriverSupported is set (TRUE) but channel {0} is not set LinlfTransceiverDrvConfig

This error occurs:

- parameter LinlfTrcvDriverSupported of the container LinlfGeneral is true but channel is not configured LinlfTransceiverDrvConfig.

ERR_Linlf_0621005: In Channel {0} - Frame {1}, Linlf{2}PduRef ({3}) is configured LinlfUser{2}UL as CDD, so {4} is NOT empty

This error will occur:

- LinlfRxIndicationUL of the container LinlfRxPdu is not configured when LinlfUserRxIndicationUL of the container LinlfRxPdu is configured as CDD.
- LinlfTxConfirmationUL of the container LinlfTxPdu is not configured when LinlfUserTxUL of the container LinlfTxPdu is configured as CDD.
- LinlfTxTriggerTransmitUL of the container LinlfTxPdu is not configured when LinlfUserTxUL of the container LinlfTxPdu is configured as CDD.

ERR_Linlf_0621006: In Channel {0} - Frame {1} has type {2}, so {3} is not empty

This error will occur:

if LinlfSubstitutionFrames is not configured when LinlfFrameType of the container LinlfFrame is configured as Sporadic/Event Triggered Frame.

ERR_Linlf_0621007: In LinlfGeneral, the LinlfTpSupported is set FALSE, so channel ({0}) - frame ({1}) is NOT able to configured type as ({2})

This error will occur:

if LinIfFrameType of the container LinIfFrame is configured as MRF or SRF when LinIfTpSupported of the container LinIfGeneral is configured as false.

ERR_LinIf_0621008: In Channel ({0}) - Schedule ({1}), the Entry ({2}) has LinIfFrameRef ({3}) which refers to is EVENT_TRIGGERED frame, so LinIfCollisionResolvingRef is NOT empty

This error will occur:

if LinIfCollisionResolvingRef of the container LinIfEntry is not configured when LinIfFrameType of the referenced Frame of this Entry is EVENT_TRIGGERED.

ERR_LinIf_0621009: In Channel ({0}) - Frame ({1}) has LinIfFrameId ({2}) while LinIfFrameType {3} ({4})

This error will occur:

- if LinIfFrameId in the container LinIfFrame is other than 60 when LinIfFrameType is configured as MRF.
- if LinIfFrameId in the container LinIfFrame is other than 61 when LinIfFrameType is configured as SRF.
- if LinIfFrameId in the container LinIfFrame is 60 when LinIfFrameType is not configured as MRF.
- if LinIfFrameId in the container LinIfFrame is 60 when LinIfFrameType is not configured as SRF.

ERR_LinIf_0621010: In Channel ({0}) - LinIfScheduleTableIndex is not {1} in a Schedule table

This error will occur:

- if LinIfScheduleTableIndex is not unique in a Channel.
- if LinIfScheduleTableIndex is not sequential in a Channel.

ERR_LinIf_0621011: In Channel ({0}) - Frame ({1}) has type ({2}), is NOT in (Sporadic/MRF/SRF), but LinIfFrameId ({3}) is repeated in Channel table

This error will occur:

if LinIfFrameId in the container LinIfFrame is repeated in a channel when LinIfFrameType LinIfFrameType is not configured as Sporadic/MRF/SRF.

ERR_LinIf_0621012: In Channel ({0}) - Frame ({1}) is configured LinIfPduDirection (LinIfSlaveToSlavePdu), so LinIfFrameType is NOT in (ASSIGN, ASSIGN_NAD, CONDITIONAL, MRF, UNASSIGN and FREE)

This error will occur:

if LinIfFrameType in the container LinIfFrame is configured as <ASSIGN, ASSIGN_NAD, CONDITIONAL, MRF, UNASSIGN and FREE> when LinIfSlaveToSlavePdu in the container LinIfFrame is configured.

ERR_LinIf_0621013: In Channel ({0}) - Frame ({1}) has type ({2}) in (<ASSIGN|ASSIGN_FRAME_ID_RANGE|ASSIGN_NAD|CONDITIONAL|FREE|SAVE_CONFIGURATION|UNASSIGN>), so LinIfFixedFrameSdu and its LinIfFixedFrameSduByte must be configured

This error will occur, if container LinIfFixedFrameSduByte is empty when LinIfFrameType in the container LinIfFrame is configured as <ASSIGN|ASSIGN_FRAME_ID_RANGE|ASSIGN_NAD|CONDITIONAL|FREE|SAVE_CONFIGURATION|UNASSIGN>

ERR_LinIf_0621014: In Channel ({0}) - Frame ({1}) has the Substitution Frames table, has LinIfFramePriority is NOT UNIQUE/SEQUENTIAL

This error will occur:

if LinIfFramePriority of the associated Unconditional Frames is repeated.

ERR_LinIf_0621015: In Channel ({0}) - Frame ({1}) has type {2} which is NOT allowed in {3}

This error will occur:

- there is 'LinIfFrame is SPORADIC in channel Slave Node.
- there is 'LinIfFrame is EVENT TRIGGERED in channel Master Node.

ERR_LinIf_0621016: In Channel ({0}) - {1} refer to a {2}, is NOT itseft

This error will occur:

if LinIfFrameRef in the container LinIfEntry/LinIfSubstitutionFrameRef in the container LinIfSubstitutionFrames is referenced to a Frame of another channel.

ERR_LinTp_0622001: The LinTp is supported but list {0} is empty

This error occurs:

- LinIf/LinIfGeneral/LinIfTpSupported is set TRUE but list (/AUTOEVER/LinTp/LinTpGlobalConfig/LinTpChannelConfig) is empty.

ERR_LinTp_0622002: The channel_ref {0} is not configured in list {1}

This error occurs:

- any Tp channel is not configured in (LinTp/LinTpGlobalConfig/LinTpRxNSdu).
- any Tp channel is not configured in (LinTp/LinTpGlobalConfig/LinTpTxNSdu).

ERR_LinTp_0622003: There is duplicate {0} is configured in LinTp/LinTpGlobalConfig/LinTpChannelConfig

This error occurs:

- there is duplicate LinTpChannelRef.

ERR_LinTp_0622004: The TX {0} must configured in Com/ComConfig/ComIPdu (ComIPduType = 'TP' and ComIPduDirection = 'SEND')

This error occurs:

- any Pdu in TX is not configured in Com/ComConfig/ComIPdu (ComIPduType = 'TP' and ComIPduDirection = 'SEND').

ERR_LinTp_0622005: The RX {0} must configured in Com/ComConfig/ComIPdu (ComIPduType = 'TP' and ComIPduDirection = 'RECEIVE')

This error occurs:

- any Pdu in RX is not configured in Com/ComConfig/ComIPdu (ComIPduType = 'TP' and ComIPduDirection = 'RECEIVE').

ERR_LinTp_0622006: The LinIf channel {0} is not configured in (LinTp/LinTpGlobalConfig/LinTpChannelConfig)

This error occurs:

- any LinIf channel (LinIfComMNetworkHandleRef) is not configured in LinTp/LinTpGlobalConfig/LinTpChannelConfig.

ERR_LinTp_0622007: There is duplicate LinTp{0}NSduId ({1}) is configured in LinTp/LinTpGlobalConfig/LinTp{0}NSdu

This error will occur:

- if LinTpTxNSduId/LinTpRxNSduId is not unique in list Tx/Rx.

ERR_LinTp_0622008: There is duplicate LinTp{0}NSduChannelRef ({1}) and LinTp{0}NSduPduRef ({2}) is configured in LinTp/LinTpGlobalConfig/LinTp{0}NSdu

This error will occur:

- if LinTpTxNSduChannelRef and LinTpTxNSduPduRef is not unique in list Tx.
- if LinTpRxNSduChannelRef and LinTpRxNSduPduRef is not unique in list Rx"

8.2.2 Warning Messages

None

8.2.3 Information Messages

None

9 Appendix

None