Document Title	**Requirements on LIN**
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	042
Document Classification	Auxiliary

Document Version	1.3.0
Document Status	Final
Part of Release	4.0
Revision	3

Document Change History

Date	Version	Changed by	Change Description
01.12.2011	1.3.0	AUTOSAR Administration	• Delete [BSW01527] • Change [BSW01588] - Add requirement of wake pin
03.12.2009	1.2.0	AUTOSAR Administration	• Conformation the use of the terms cluster, network, bus and channel • Add LIN 2.1 support • Additional requirements for the LIN Tranceiver Driver • Legal disclaimer revised
23.06.2008	1.1.3	AUTOSAR Administration	Legal disclaimer revised
31.10.2007	1.1.2	AUTOSAR Administration	• Document meta information extended • Small layout adaptations made
24.01.2007	1.1.1	AUTOSAR Administration	• “Advice for users” revised • “Revision Information” added
04.10.2006	1.1.0	AUTOSAR Administration	• Legal disclaimer revised • Extend description of rationale of BSW01555
24.02.2006	1.0.0	AUTOSAR Administration	Initial Release

Disclaimer

This specification and the material contained in it, as released by AUTOSAR is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the specification.

The material contained in this specification is protected by copyright and other types of Intellectual Property Rights. The commercial exploitation of the material contained in this specification requires a license to such Intellectual Property Rights.

This specification may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only.
For any other purpose, no part of the specification may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The AUTOSAR specifications have been developed for automotive applications only. They have neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Advice for users

AUTOSAR Specification Documents may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the Specification Documents for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such Specification Documents, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

Table of Contents

1 Scope of this document

This document specifies the requirements for the following Basic Software Modules (module names in brackets):

- LIN Driver (Lin)
- LIN Interface (LinIf)
- LIN Transport Protocol (LinTp)

The intention is to reference as much as possible to the LIN 2.1 specification (see 4.3.2). The behaviour will be restricted to a master node. It is the goal to support LIN 2.1 slaves, LIN 2.0 slaves and LIN 1.3 slaves already existing on the market (i.e. that conforms to the respective specification).

The reader of this document should know the LIN specifications.

2 How to read this document

Each requirement has its unique identifier starting with the prefix “BSW” (for “Basic Software”). For any review annotations, remarks or questions please refer to this unique ID rather than chapter or page numbers!

2.1 Conventions used

In requirements, the following specific semantics are used (taken from Request for Comment RFC 2119 from the Internet Engineering Task Force IETF)

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. Note that the requirement level of the document in which they are used modifies the force of these words.

- MUST: This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
- MUST NOT: This phrase, or the phrase „SHALL NOT“, means that the definition is an absolute prohibition of the specification.
- SHOULD: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- MAY: This word, or the adjective „OPTIONAL“, means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation, which does not include a particular option, MUST be prepared to interoperate with another implementation, which does include the option, though perhaps with reduced functionality. In the same vein an implementation, which does include a particular option, MUST be prepared to interoperate with another implementation, which does not include the option (except, of course, for the feature the option provides.)

2.2 Requirement structure

Each module specific chapter contains a short functional description of the Basic Software Module. Requirements of the same kind within each chapter are grouped under the following headlines (where applicable):

“Definitions for LIN” are divided in four chapters.
LIN General requirements
LIN Interface
LIN Driver
LIN TP

The subchapters are only applied, if they are needed. See following structure.

- Non-functional requirements
- Functional requirements
- Configuration
- Initialization
- Normal Operation
- Fault Operation
- Shutdown Operations

3 Acronyms and abbreviations

The LIN 2.1 Glossary is kept as far as possible to make LIN knowledgably readers familiar with this document. Acronyms and abbreviations that are not found in the LIN 2.1 Glossary and therefore are not contained in the AUTOSAR glossary are described here.

Acronym:	***Description:***
LIN-PDU	LIN **P**rotocol **D**ata **U**nit is the LIN header and the LIN response, i.e. Break, synch, PID, Data (1-8) and checksum In LIN 2.0 specification this is called just frame. LIN_PDU is more precise and omits confusion.
LIN-SDU	LIN Service Data Unit. The data-part of the LIN response.
Schedule Table	A Schedule Table determines the traffic on a LIN bus (one channel). One LIN bus could have more than one Schedule Table.
Schedule Table Handler	The Schedule Table Handler is placed at the LIN Interface. It will initiate LIN-PDU's and confirm/indicate LIN-PDU's. It will be called by upper layers.
Schedule Table Manager	Keeps track of all available schedule and processes the active schedule table.
LIN Driver	Module name Lin. Describes the Software Driver.
LIN Interface	Module name LinIf. LIN Interface , describes the LIN 2.1 master communication stack (= LIN Master functionality)
Sleep-mode	In the LIN 2.1 specification the term stand-by and sleep-mode is used in similar manner. To be consequent here only sleep-mode is used

Abbreviation:	***Description:***
LIN	Local Interconnect Network
FF	First Frame
CF	Consecutive Frames
SF	Single Frames
N_PDU	Network Protocol Data Unit
PDUR	Protocol Data Unit Router
N_SDU	Network Service Data Unit
N_TA	Extended Addressing Mode Connection
UART	Universal Asynchronous Receiver Transmitter. Dear children have many names, it is also known as SCI and ESCI.
MRF	Master Request Frame
SRF	Slave Response Frame

4 Requirements Specification

4.1 Functional requirements

4.1.1 LIN General

4.1.1.1 General Requirements

4.1.1.1.1 [BSW01576] Usage of LIN specification

ID:	BSW01576
Initiator:	BMW
Date:	16.04.2008
Short Description:	The LIN 2.1 specification shall be reused as far as possible
Type:	New
Importance:	High
Description:	The following sections in the LIN 2.1 specification shall be reused - Chapter 2.3 Frame Transfer - Chapter 2.4 Schedules tables - Chapter 2.5 Task Behaviour Model - Chapter 2.6 Network Management - Chapter 2.7 Status Management - Chapter 3 Transport Layer - Chapter 4 Node Configuration and identification - Chapter 5 Diagnostics specification (partly) The diagnose classes II and III in AUTOSAR are optional and a precompiled option (see [BSW01579]). The Diagnostic Transport Protocol is also used by the node configuration and identification in the LIN 2.1 specification, so it is mandatory in AUTOSAR as a precompiled option. The remaining chapters in the LIN 2.1 specification will not be reused as is. Refer to the corresponding LIN Driver and Interface sections for the exact details. There are optional functionality in the LIN 2.1 specification (e.g. in the configuration): - All other optional functionality is decided by the design. The following item shall be used with AUTOSAR adaptations: - Application Program Interface Specification The [BSW01577] will take care that the Lin Interface is Compatible with the LIN 2.1 Specification, also [BSW01578] for the LIN Driver and [BSW01579] for the Lin TP. The usage of LIN 1.3 Nodes is already covered with the LIN 2.1 specification. If a cluster uses LIN2.0 nodes, the LIN Master has to support the LIN 2.0 configuration services also.
Rationale:	Reuse of existing standards. This ensures the reusability of LIN slave ECU's inside the vehicle architecture. Each 2.1 LIN Master will support both models to assign LIN IDs(2.0 and 2.1), the enhanced checksum for LIN 2.x and the classic checksum for LIN 1.x slaves and diagnostic messages (see LIN 2.1 protocol specification chapter 2.3.1.5 Checksum)

	The LIN Physical Layer Specification is not in the scope of AUTOSAR The checksum models (classic and enhanced) will be configurable for each LIN ID, except for the reserved LIN ID's (MRF and SRF)
Use Case:	--
Dependencies:	[BSW01577], [BSW01578] , [BSW01579]
Conflicts:	--
Supporting Material:	LIN 2.1 specification, LIN 2.0 specification
Contributes to:	--

4.1.1.1.2 [BSW01504] Usage of AUTOSAR architecture only in LIN master nodes mandatory

ID:	BSW01504
Initiator:	Vector
Date:	14.07.2004
Short Description:	For AUTOSAR it is only intended to specify and thus standardize the LIN master capabilities. LIN slave node implementations are not in the scope of these requirements.
Type:	New
Importance:	High
Description:	The AUTOSAR LIN should cover only LIN master nodes and skip LIN slave nodes. LIN slave nodes are aimed for maximal reuse and flexibility with low costs (i.e. RAM, ROM and runtime).
Rationale:	AUTOSAR architecture will need too many resources for a simple LIN slave. Thus not in the scope of AUTOSAR.
Use Case:	--
Dependencies:	--
Conflicts:	This requirement may seem to conflict with the requirement [BSW01576]. It is in fact a further specialization of that requirement.
Supporting Material:	--
Contributes to:	--

4.1.1.2 Initialization

4.1.1.2.1 [BSW01590] Usage of schedule tables for node configuration

ID:	BSW01590
Initiator:	BMW
Date:	23.04.2008
Short Description:	The node configuration of LIN slaves shall only be done via defined schedule table(s).
Type:	New
Importance:	High
Description:	The AUTOSAR LIN should cover the “normal” behavior to do the configuraton (assign Ids, NADs,…) of LIN slaves. This shall avoid non LIN-conform configuration methods.
Rationale:	This shall be part of the verification of the system design.
Use Case:	LIN node configuration at bus start or after a node fault
Dependencies:	--
Conflicts:	--

Supporting Material:	LIN 2.1 specification
Contributes to:	--

4.1.1.3 Normal Operation

4.1.1.3.1 [BSW01522] Consistent data transfer

ID:	BSW01522
Initiator:	CAS
Date:	03.04.2004
Short Description:	LIN-SDU shall be copied consistently for transfer
Type:	New
Importance:	High
Description:	The data from the upper layers needs to be copied consistently to the LIN Driver before transmission. The data from the LIN Driver shall be copied consistently to the upper layers after reception. The consistent coping includes the payload (data) and the flags.
Rationale:	Basic functionality. Guarantee 100% message LIN-SDU consistency for transmission and reception. Needed for every LIN-PDU transmission/reception on the LIN Bus.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.1.4 Shutdown Operation

4.1.1.4.1 [BSW01560] Support for wakeup during transition to sleep-mode

ID:	BSW01560
Initiator:	BMW
Date:	07.07.2005
Short Description:	If a wakeup occurs during transition to sleep-mode, this channel shall go back to the running mode
Type:	New
Importance:	High
Description:	If a wakeup occurs during transition to sleep-mode, the affected channels shall go back to the running mode. The upper layer shall be notified. In detail: If an upper layer wake-up is received during Stop: Stop process should be completed, and network affected LIN cluster started afterwards. If a slave wake-up is received during Stop: Stop operation should be cancelled, and upper layer notified, so that it decides which scheduler to start.
Rationale:	Safe wakeup and sleep handling.
Use Case:	The following use-cases shall be detected: If the master is processing the go-to-sleep command and the upper layer requests a wakeup.

	There is a time from the go-to-sleep command is transmitted on the bus until it is confirmed in the LIN Interface. During this time it is possible that a slave will transmit a wakeup-request
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.1.4.2 Fault Operation

None

4.1.2 LIN Interface

4.1.2.1 General requirements

4.1.2.1.1 [BSW01577] Compatibility to LIN protocol specification

ID:	BSW01577
Initiator:	BMW
Date:	23.04.2007
Short Description:	Compatibility to LIN 2.1 protocol specification
Type:	New
Importance:	High
Description:	The following sections of the LIN 2.1 specification shall be reused by the LIN Interface: LIN 2.1 Protocol Specification: - Chapter 2.3 Frame Transfer (handling of different types of LIN-PDUs, not the specific bytes of the LIN-PDU) - Chapter 2.4 Schedules tables - Chapter 2.5 Task Behaviour Model (handling of LIN-PDUs and errors. Not handling specific bytes in the LIN-PDU) - Chapter 2.6 Network Management - Chapter 2.7 Status Management The LIN 2.1 specification includes the behavior of previous versions of LIN specifications. If a cluster uses LIN2.0 nodes, the 2.1 LIN Master has to support the LIN 2.0 behavior.
Rationale:	Basic LIN functionality
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	LIN 2.1 protocol specification
Contributes to:	--

4.1.2.1.2 [BSW01551] Multiple LIN channel support for interface

ID:	BSW01551
Initiator:	BMW
Date:	16.12.2004
Short Description:	One LIN Interface shall support one or more LIN Drivers.
Type:	New
Importance:	Medium

Description:	There shall only be one instance of the LIN Interface in each ECU. One ECU might contain more than one LIN channel. Thus the LIN Interface shall support one or more LIN Drivers.
Rationale:	Devices, which use more than one LIN channels, exist on the market.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.2.1.3 [BSW01568] Hardware Independence

ID:	BSW01568
Initiator:	Mentor
Date:	08.12.2005
Short Description:	The LIN Interface implementation and interface shall be independent from underlying LIN hardware.
Type:	New
Importance:	High
Description:	The implementation may depend on the amount of available resources of the underlying hardware. The different mechanisms of hardware access are encapsulated by the LIN driver.
Rationale:	Portability and reusability.
Use Case:	If the underlying LIN device driver just handles one controller the implementation of the LIN interface may be more efficient.
Dependencies:	[BSW01552]
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.2.2 Initialization

4.1.2.2.1 [BSW01569] LIN Interface initialization

ID:	BSW01569
Initiator:	Mentor
Date:	12.12.2005
Short Description:	LIN Interface initialization
Type:	New
Importance:	Medium
Description:	The LIN Interface shall support initialization of each LIN channel separately. The selection of at least one static configuration set shall be done by a parameter.
Rationale:	--
Use Case:	If there are more than one LIN channels, than there exist more than one LDF-files.
Dependencies:	--
Conflicts:	--
Supporting Material:	Comparing with LIN 2.1 specification API's, the LIN Interface init will do both work of l_ifc_init and l_sys_init
Contributes to:	--

4.1.2.2.2 [BSW01570] Selection of static configuration sets

ID:	BSW01570
Initiator:	Mentor
Date:	09.12.2005
Short Description:	The LIN Interface shall support dynamic selection of configuration sets.
Type:	New
Importance:	Medium
Description:	The LIN Interface shall support the dynamic selection of at least one static configuration set by a parameter passed via the initialization interface. The selection of the appropriate configuration set itself as well as the way to incorporate the configuration sets into the ECU (Post-Build, Pre-Compile) is not affected by this requirement.
Rationale:	Support of different configurations during runtime
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.2.3 Normal Operation

4.1.2.3.1 [BSW01564] Schedule Table Manager

ID:	BSW01564
Initiator:	3Soft / BMW
Date:	18.10.2005
Short Description:	The schedule table manager is the responsible to select the active schedule table.
Type:	New
Importance:	High
Description:	The schedule table manager will keep schedule table to execute. The schedule table manager shall: • Be able to receive requests from an upper layer (e.g. LIN NM) which schedule table to execute • Keep a list of schedule table • Prioritize the requested schedule tables (immediately, or later) • Each one time schedule table shall have a unique priority per channel • Execute a schedule table once or continuously • Change to a specific requested schedule table depending on priority One or more modules from an upper layer will request the schedule table manager to execute specific schedule tables. The schedule table manager creates a sequence in runtime and instructs the schedule table handler which schedule table to execute. The schedule table manager will only coordinate the request from other modules requested Schedule Tables. The priority of the schedule tables is preconfigured. There exist one memory space for the "continuously execution schedule table", it will be overwritten by an newer request, the priority is always 1. The schedule table manager must be able to buffer the requests for "execution one time schedules", the size of the queue depends on the number of existing schedule tables (Number of existing Schedule Tables = Size of queue). The priority of the "execution one time schedules" must be unique.

	The priority of the "continuously execution schedule tables" is always the same and the lowest.
Rationale:	In LIN 2.1 the application interfaces directly to the LIN API. In AUTOSAR above modules shall be able to independently request a schedule table to be executed. Therefore the schedule table manager is a necessarily extension to the schedule table handler.
Use Case:	Example system start: a) "Run"- schedule table (execution continuously, low priority 1) b) "Wakeup"- schedule table (execution one time, high priority 10) c) "Node-01-init"- schedule table (execution one time, high priority 9) d) "Node-02-init"- schedule table (execution one time, high priority 8) → Sequence b) c) d) a) a) a)... Example re-init of a node after a node-reset e)"Run"- schedule table (execution continuously, low priority 1) f) "Node-02-init"- schedule table (execution one time, high priority 8) → Sequence ... a) a) f) a) a) ...
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.2.3.2 [BSW01546] Schedule Table Handler

ID:	BSW01546
Initiator:	BMW
Date:	20.09.2004
Short Description:	The LIN Interface shall contain a Schedule Table Handler.
Type:	New
Importance:	High
Description:	The schedule table handler will handle the transmission and reception of the LIN-PDUs on the LIN bus. It will query the Schedule table manager when active schedule table has reached the point to start transmission or reception of the LIN-PDU (i.e. when a schedule entry is due). The schedule table handler shall notify the upper layer about a successful / erroneous LIN-PDU transfer / reception through callbacks. The LIN 2.1 Spec. defines that the change of an schedule table occur at the end of a timeslot. The recommendation: A schedule table change from a “continuously execution schedule table" to a "execution one time schedule table” should occur at the next timeslot. A schedule table change from a "execution one time schedule table” to a “continuously execution schedule table" or other "execution one time schedule table” should occur at the end of the current schedule table.
Rationale:	--
Use Case:	--
Dependencies:	[BSW01564]
Conflicts:	--
Supporting Material:	LIN 2.1 specification, see LIN API l_sch_tick.
Contributes to:	--

4.1.2.3.3 [BSW01561] Main function

ID:	BSW01561
Initiator:	BMW
Date:	07.07.2005

Short Description:	The LIN Interface shall define a main function
Type:	New
Importance:	High
Description:	The main function is responsible for executing the schedule table handler Only one main function shall exist that executes the schedule table handler on all busses.
Rationale:	--
Use Case:	If an ECU is master on three LIN buses there is only one main function that executes all the schedule tables on the different busses.
Dependencies:	[BSW01546]
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.2.3.4 [BSW01549] Timer service for Scheduling

ID:	BSW01549
Initiator:	BMW
Date:	16.12.2004
Short Description:	The LIN Interface needs to use a timer service for scheduling
Type:	New
Importance:	Medium
Description:	The LIN Interface needs to use a timer service for scheduling. The LIN-PDU transmission and reception must be transported at the right time. The main function is taking care of the schedule handler, so it means that this function must be called with a given period.
Rationale:	To uphold normal communication.
Use Case:	--
Dependencies:	[BSW01561]
Conflicts:	--
Supporting Material:	This is adequate to the “time base” that is defined in the LIN 2.1 Configuration Language specification.
Contributes to:	--

4.1.2.3.5 [BSW01571] Transmission request service

ID:	BSW01571
Initiator:	Mentor
Date:	12.12.2005
Short Description:	Transmission request service
Type:	New
Importance:	Medium
Description:	The LIN Interface shall provide a transmission request service that allows an upper layer to request the LIN interface for a Sporadic LIN-PDU transmission. The LIN Interface transmits the Sporadic LIN-PDU according to the schedule-table rules.
Rationale:	To enable the Sporadic LIN-PDU behavior in AUTOSAR
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	See the LIN 2.1 Protocol specification chapter 2.3.3.3 Sporadic Frame.
Contributes to:	--

4.1.2.3.6 [BSW01514] Wake-up notification support

ID:	BSW01514
Initiator:	Vector
Date:	14.07.2004
Short Description:	The LIN Interface shall inform an upper layer about wake-up events
Type:	New
Importance:	Medium
Description:	The LIN Interface shall inform an upper layer if a wake-up request was notified by the underlying LIN Driver A wakeup notification shall only be raised in the case a go-to-sleep-command has been issued to the bus.
Rationale:	Basic functionality
Use Case:	Wakeup of ECU by LIN. Inform upper layer (ECU State Manager) about the wakeup reason
Dependencies:	--
Conflicts:	--
Supporting Material:	ECU state manager
Contributes to:	--

4.1.2.3.7 [BSW01515] API to wake-up by upper layer to LIN Interface

ID:	BSW01515
Initiator:	Vector
Date:	14.07.2004
Short Description:	The LIN Interface shall provide an API to wake-up a LIN channel cluster
Type:	New
Importance:	Medium
Description:	The LIN Interface shall provide an API to wake-up a LIN channel cluster. The LIN Interface shall support that each LIN channel cluster can be woken up separately.
Rationale:	Wake-up of LIN by upper layer.
Use Case:	--
Dependencies:	LIN NM specification.
Conflicts:	This is only allowed when ECU is capable to communicate via LIN. The LIN NM and/or ECU State Manager shall solve this dependency.
Supporting Material:	LIN 2.1 API specification, chapter 7.2.5.3 l_ifc_wake_up and LIN 2.1 Protocol Specification, chapter 2.6.2 Wake-up
Contributes to:	--

4.1.2.3.8 [BSW01502] RX indication and TX confirmation call-backs

ID:	BSW01502
Initiator:	Vector
Date:	14.07.2004
Short Description:	The LIN Interface shall support an API for RX/TX notifications.
Type:	New
Importance:	High
Description:	The PDU router provides APIs for RX notification and TX confirmation. The LIN Interface shall use this API.
Rationale:	This allows a clear interface to the upper layers (PDU router).
Use Case:	LIN master node implementation with e.g. gateway to other bus systems.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

Contributes to:	--

4.1.2.3.9 [BSW01558] Check successful communication

ID:	BSW01558
Initiator:	BMW
Date:	10.06.2005
Short Description:	Check the successful communication
Type:	New
Importance:	Medium
Description:	The LIN Interface shall query the LIN driver if the last message is successful transmitted or received on the LIN bus. This check shall be done by the schedule table handler. When the successful communication*) is detected the appropriate layer above shall be notified. The schedule table handler may also check if the LIN-PDU violates the maximum frame length. It is however not recommended since the overhead is too big and that all nodes in a LIN channel cluster shall conform to the LIN 2.1 DLL and NMNC test specification. *) see LIN 2.1 Spec., Status Management, Chapt.2.7, “Successful_transfer” shall be set when a frame has been successfully transferred by the node, i.e. a frame has either been received or transmitted.
Rationale:	--
Use Case:	The normal implementation would be to make the check in the main function called periodically after the LIN-PDU has been sent.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.2.4 Shutdown Operation

4.1.2.4.1 [BSW01523] API to send the LIN to sleep-mode

ID:	BSW01523
Initiator:	Vector
Date:	14.07.2004
Short Description:	There shall be a API call to send the LIN bus to sleep-mode.
Type:	New
Importance:	Medium
Description:	The LIN Interface shall provide an API to send the go-to-sleep-command on the LIN bus. It shall be possible to send the go-to-sleep-command on each LIN bus independently of each other
Rationale:	Basic functionality
Use Case:	--
Dependencies:	The LIN NM and/or ECU State Manager shall prevent any transition to an ECU state where no LIN communication is possible until the LIN go-to-sleep-command has been successfully executed.
Conflicts:	None
Supporting Material:	LIN 2.1 specification
Contributes to:	--

4.1.2.5 Fault Operation

none

4.1.3 LIN Driver

4.1.3.1 General requirements

4.1.3.1.1 [BSW01578] Compatibility to LIN Datalinklayer

ID:	BSW01578
Initiator:	BMW
Date:	23.04.2008
Short Description:	Compatibility to LIN 2.1 protocol specification (task behavior model)
Type:	New
Importance:	High
Description:	The frame processor has to be emulated by LIN Driver if not already supported by hardware e.g. LIN Controller The Task Behavior Model (chapter 2.5 in the LIN 2.1 Protocol specification) is part of the LIN Driver. The LIN 2.1 specification includes the behavior of previous versions of LIN specifications.
Rationale:	Basic LIN functionality
Use Case:	A device driver using an UART will implement the complete Task Behavior Model. If a LIN Hardware (e.g. LIN controller) is used, parts of the Task Behavior Model runs in hardware
Dependencies:	--
Conflicts:	--
Supporting Material:	LIN 2.1 protocol specification
Contributes to:	--

4.1.3.1.2 [BSW01553] Basic Software SPAL General requirements

ID:	BSW01553
Initiator:	BMW
Date:	16.12.2004
Short Description:	The LIN Driver shall fulfill the general SPAL requirements for Basic Software Modules.
Type:	New
Importance:	Medium
Description:	The LIN Driver shall fulfill the general SPAL requirements for Software Modules as specified in AUTOSAR_SRS_SPAL_General.SRS
Rationale:	Re-use of requirements valid for all low level Drivers
Use Case:	LIN Driver is in the same layer as the SPAL Drivers (e.g.: SPI, ADC). Therefore the general SPAL requirements shall be fulfilled by the LIN Driver also, if applicable.
Dependencies:	--
Conflicts:	General Requirements of SPAL doesn't have a final state
Supporting Material:	AUTOSAR General Requirements on SPAL [3]
Contributes to:	--

4.1.3.1.3 [BSW01552] Hardware abstraction LIN

ID:	BSW01552

Initiator:	BMW
Date:	16.12.2004
Short Description:	The LIN Driver shall offer a Hardware independent interface.
Type:	New
Importance:	High
Description:	The Interface from LIN Interface to LIN Driver shall be independent from underlying hardware.
Rationale:	Portability
Use Case:	Same LIN Interface can be used for different µCs.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.3.1.4 [BSW01503] Frame based API for send and received data

ID:	BSW01503
Initiator:	Vector
Date:	14.07.2004
Short Description:	An API shall exist that enables the LIN driver to directly copy up to 8 byte directly from/to the frame buffers.
Type:	New
Importance:	High
Description:	AUTOSAR COM creates the frames to be sent via CAN, LIN and other busses. The frames are transferred “as a block” to the lower layer. The CAN/LIN layers have therefore a need for an efficient read/write access of whole frame buffers (1 to 8 bytes).
Rationale:	Same behavior for AUTOSAR COM independently if reception/transmission is CAN or LIN based.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.3.1.5 [BSW01555] LIN Interface shall poll the LIN Driver for transmit/receive notifications

ID:	BSW01555
Initiator:	BMW
Date:	13.06.2005
Short Description:	The LIN driver shall have an API which the driver shall use to poll for transmit / receive notifications.
Type:	New
Importance:	High
Description:	The LIN Interface shall be able to poll the LIN Driver for transmit/receive notifications.
Rationale:	According to the Autosar Basic Software Architecture notifications by interrupt are not supported.
Use Case:	Basic functionality
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.3.1.6 [BSW01547] Support of standard UART and LIN optimized HW

ID:	BSW01547
Initiator:	Renesas / BMW
Date:	14.10.2004
Short Description:	The LIN Driver will support UART’s and LIN optimized Hardware
Type:	New
Importance:	High
Description:	The LIN Driver is responsible to handle the frame according to the hardware. It should be possible to support the complete the range of hardware from implementation using an UART to a complex LIN hardware controller. Using SW UART’s is out of the scope.
Rationale:	Implement a common driver interface. The LIN Driver will process the complete frame by it self.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.3.2 Initialization

4.1.3.2.1 [BSW01572] LIN driver initialization

ID:	BSW01572
Initiator:	Mentor
Date:	12.12.2005
Short Description:	LIN driver initialization
Type:	New
Importance:	Medium
Description:	The LIN Driver shall support the initialization of each LIN channel separately. The selection of at least one static configuration set shall be done by a parameter.
Rationale:	Hardware specific initialization of the UART, LIN controller. Initiation of variables used in the LIN driver.
Use Case:	If there are e.g. 2 LIN channels than there are also 2 different configuration files.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.3.2.2 [BSW01573] Selection of static configuration sets

ID:	BSW01573
Initiator:	Mentor
Date:	09.12.2005
Short Description:	The LIN Driver shall support dynamic selection of configuration sets.
Type:	New
Importance:	Medium
Description:	The LIN Driver shall support the dynamic selection of at least one static configuration set by a parameter passed via the initialization interface. The selection of the appropriate configuration set itself as well as the way to incorporate the configuration sets into the ECU (Post-Build, Pre-Compile) is not affected by this requirement.

Rationale:	Support of different configurations during runtime
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.3.3 Normal Operation

4.1.3.3.1 [BSW01563] Wake-up Notification

ID:	BSW01563
Initiator:	VW/IAV
Date:	13.09.2005
Short Description:	The LIN Driver shall provide a notification for wake-up events
Type:	New
Importance:	High
Description:	The LIN Driver shall notify the LIN Interface in case of a wake-up interrupt. The corresponding callback function itself is implemented inside the LIN Interface. This functionality shall only be implemented, if the LIN Hardware unit has a wake-up interrupt capability. The wake-up interrupt shall only be enabled when the channel is in sleep-mode mode. Otherwise a SynchBreak will be considered as a wake-up request.
Rationale:	Inform upper layer about the occurrence of a wake-up event
Use Case:	--
Dependencies:	[BSW01514]
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.3.3.2 [BSW01556] Multiple LIN channel support for driver

ID:	BSW01556
Initiator:	BMW
Date:	13.06.2005
Short Description:	Multiple LIN channel support for driver
Type:	New
Importance:	Medium
Description:	One LIN driver shall able to handle more than one LIN channel if the underlying hardware is equipped with more than one identical LIN controllers.
Rationale:	Portability
Use Case:	If an ECU is a master on two channels and it contains two identical UART hardware modules there is only one LIN driver interfacing both UARTs
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.3.4 Shutdown Operations

4.1.3.4.1 [BSW01566] Transition to sleep-mode

ID:	BSW01566

Initiator:	WP CAN, LIN
Date:	23.11.2005
Short Description:	Transition to sleep-mode
Type:	New
Importance:	High
Description:	After the LIN Driver is requested to be set to the sleep-mode by the appropriate function call it has to do as follows: - The LIN Driver shall activate sleep-mode as soon as possible after bus is idle. - After successful transmission of the go-to-sleep-command the wakeup monitoring shall be activated. - After wakeup monitoring is active the sleep mode shall be set and can be read out by LINif afterwards. Each LIN channel shall be handled independently.
Rationale:	Basic functionality
Use Case:	--
Dependencies:	[BSW01524]
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.3.4.2 [BSW01524] Support of reduced power operation mode

ID:	BSW01524
Initiator:	CAS
Date:	29.07.2004
Short Description:	Support of reduced power operating mode
Type:	New
Importance:	Medium
Description:	When going to sleep-mode mode, the LIN Driver shall put the corresponding LIN hardware to a reduced power operation mode if supported by hardware. This command shall be possible to be activated for each channel independently. This requirement does not conflict to [BSW01566]. This requirement [BSW01524] enables the Power Mode in the LIN hardware the other requirement [BSW01566] sets the LIN driver in sleep-mode mode.
Rationale:	Power saving
Use Case:	--
Dependencies:	[BSW01566]
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.3.5 Fault Operation

4.1.3.5.1 [BSW01526] Error notification

ID:	BSW01526
Initiator:	Vector
Date:	14.07.2004
Short Description:	The LIN Driver shall provide a status for error events on the bus.
Type:	New
Importance:	Medium

Description:	The LIN driver shall provide an API that returns the errors detected in the LIN communication. When the call is made the error-flags shall be reset. Each LIN channel shall be capable to notify its errors separately to the LIN interface
Rationale:	Bus error handling
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	Similar to the l_read_status function in the LIN 2.1 API specification.
Contributes to:	--

4.1.4 Lin Transceiver Driver

4.1.4.1 General Requirements

4.1.4.2 Configuration

4.1.4.2.1 [BSW01580] Configuration Data for LIN Transceiver Driver.

ID:	BSW01580
Initiator:	Bosch
Date:	04.08.2008
Short Description:	The bus transceiver driver package shall offer configuration parameters that are needed to configure the driver for a given bus and the supported notifications.
Type:	New
Importance:	Medium
Description:	Typical parameters are: - Wakeup by bus - Transceiver control via SPI or port pin - Call context of the notification functions (ISR, polling) to enable detection of necessary data consistency mechanisms during configuration time Please refer to the corresponding software specification for a more detailed view
Rationale:	Basic functionality for transceiver configuration.
Use Case:	--
Dependencies:	ECU configuration description
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.4.2.2 [BSW01581] Support for more than one LIN transceiver. Only pre-compile time configuration allowed.

ID:	BSW01581
Initiator:	Bosch
Date:	04.08.2008
Short Description:	The LIN transceiver driver shall support the configuration for more than one channel
Type:	New
Importance:	High
Description:	The driver shall be able to support multiple LIN busses on the ECU. It must be possible to configure the used transceiver type independently for each bus. Only Pre-Compile-Time configuration shall be possible

	Transceiver handling depends strongly on the used device. Therefore each transceiver may need its own implementation within the driver and only known and supported devices could be selected. A general solution for the transceiver driver for all use cases might not be possible. By default each LIN controller is attached to an own bus and needs therefore an own bus transceiver.
Rationale:	Basic functionality for transceiver configuration
Use Case:	Multi bus systems, e.g. LIN-LIN gateways
Dependencies:	ECU resource template
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.4.3 Initialization

4.1.4.3.1 [BSW01583] API to initialize the LIN Transceiver Driver.

ID:	BSW01583
Initiator:	Bosch
Date:	04.08.2008
Short Description:	The bus transceiver driver shall provide an API to initialize the driver internally and set then all attached transceivers in their pre-selected operation modes.
Type:	New
Importance:	High
Description:	The driver must be initialized during the power-up/reset sequence of the ECU. Depending on the used drivers to control the transceivers (e.g. DIO, SPI), they must be already available and working when the transceiver driver is initialized. The wakeup reason has to be detected and stored during the execution of the driver initialization, too
Rationale:	Set bus transceivers and driver in a pre-defined and known state
Use Case:	Basic functionality for transceiver control.
Dependencies:	SPI and DIO driver initialization. [BSW01588] API to read out wakeup reason. The bus transceiver driver setup information must provide the necessary configuration data to enable the generation tool to select the appropriate control mechanism (e.g. SPI, I/O ports) and to guarantee the correct allocation of the necessary communication resources and initialization sequences.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.4.4 Normal Operation

4.1.4.4.1 [BSW01582] LIN Transceiver Driver API shall be synchroneous.

ID:	BSW01582
Initiator:	Bosch
Date:	04.08.2008
Short Description:	The bus transceiver driver API shall be synchronous.
Type:	New

Importance:	High
Description:	The bus transceiver driver API shall execute the requested action immediately and shall deliver the result state immediately to the caller. This will ease up the implementation of wakeup and sleep concepts within the AUTOSAR BSW stack
Rationale:	Better usage of transceiver functionality in the complex AUTOSAR BSW environment.
Use Case:	Atomic transition to other operation mode; easier and better abstraction for upper layers like the ECU state manager or ComManager. Improved testability compared to asynchronous handling.
Dependencies:	ECU state manager, NM. SPAL in case a transceiver is connected via SPI
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.4.4.2 [BSW01584] API to request operation mode “standby”.

ID:	BSW01584
Initiator:	Bosch
Date:	04.08.2008
Short Description:	The bus transceiver driver shall support an API to send the addressed transceiver into its Standby mode.
Type:	New
Importance:	High
Description:	Many transceivers support the transition to the Sleep mode only via the transition to Standby mode. In addition, some power concepts have the need to set the transceiver to Standby only instead of Sleep mode. Not all transceivers will support such a state. If this is true for a given device, the driver shall confirm the state transition with success
Rationale:	Implementation of ECU low power modes with wakeup via bus and internal.
Use Case:	The upper service layers agreed together with other nodes to set the bus into the sleep mode. The transceiver shall be switched now to a state where the wakeup via bus is supported and the power consumption is as low as possible for the current state of the ECU.
Dependencies:	[BSW01585] API to request operation mode “sleep”.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.4.4.3 [BSW01585] API to request operation mode “sleep”.

ID:	BSW01585
Initiator:	Bosch
Date:	04.08.2008
Short Description:	The bus transceiver driver shall support an API to send the addressed transceiver into its Sleep mode.
Type:	New
Importance:	High
Description:	The transition to sleep mode will be requested with this API. Not all transceivers will support such a state. If this is true for a given device, the drive shall confirm the state transition with success
Rationale:	Implementation of ECU low power modes with wakeup via bus and internal.
Use Case:	The upper service layers agreed together with other nodes to set the bus into the sleep mode. The transceiver is already in StandBy and shall be

	switched to Sleep with lowest power consumption. Please note that the state sleep of the transceiver is often similar to the state “unpowered” of the ECU.
Dependencies:	[BSW01584] API to request operation mode “standby”.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.4.4.4 [BSW01586] API to request operation mode “normal”.

ID:	BSW01586
Initiator:	Bosch
Date:	04.08.2008
Short Description:	The bus transceiver driver shall support an API to send the addressed transceiver into its Normal mode.
Type:	New
Importance:	High
Description:	All transceivers support this state due to it’s the “working state”
Rationale:	Communication!
Use Case:	All communication must be enable to communicate.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.4.4.5 [BSW01587] API to read out current operation mode.

ID:	BSW01587
Initiator:	Bosch
Date:	04.08.2008
Short Description:	The bus transceiver driver shall support an API to read out the current operation mode of the transceiver of a specified bus within the ECU.
Type:	New
Importance:	High
Description:	The current operation mode of the transceiver will be necessary for upper layers (e.g. diagnostics). The API shall always return the current state seen by the transceiver driver (this may be a locally stored state, too)
Rationale:	State access to transceiver driver
Use Case:	Check for current operational mode during development and via diagnostic command.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.4.4.6 [BSW01588] API to read out wakeup reason.

ID:	BSW01588
Initiator:	Bosch
Date:	04.08.2008
Short Description:	The bus transceiver driver shall support an API to read out the reason of the last wakeup of a specified bus within the ECU.
Type:	New
Importance:	High
Description:	The transceiver driver shall be able to store the local view “who has requested the wakeup: bus or internally”.

	- Bus: The bus has caused the wakeup. - Internally: The wakeup has been caused by software - Sleep: The transceiver is in operation mode sleep and no wakeup has been occurred. - Wake pin: An edge on the wake pin of the transceiver (if present) has caused the wakeup. The wakeup reason should be “sleep” when the operation mode is not Normal and no wakeup has been occurred. When a wakeup has occurred, the API shall always return the first detected wakeup reason (e.g. if a wakeup by bus occurs and than nearly at the same time an internal wakeup, the wakeup reason is “bus”.). After leaving the operation mode Normal, the wakeup reason shall be set to “sleep” again
Rationale:	Detection of wakeup reason during development and via diagnostic command. May also be used by the NM or ECU state manager.
Use Case:	--
Dependencies:	(Bus specific) NM, diagnostics, ECU state manager
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.1.4.4.7 [BSW01589] API to enable/disable/clear wakeup Event.

ID:	BSW01589
Initiator:	Bosch
Date:	04.08.2008
Short Description:	The bus transceiver driver shall support an API to enable and disable the wakeup notification for each bus separately.
Type:	New
Importance:	High
Description:	To enable upper layers to command the bus transceiver safe into its standby and/or sleep state, an additional API to disable and enable the wakeup notification is necessary. If the notification is disabled, driver shall not perform the notification but store the event internally until the notification is enabled again. The notification shall then be processed immediately. It shall be possible to clear a pending wakeup event. If no further wakeup event occurs, no notification shall be performed after enabling the notification again. If a further wakeup event occurs it shall be notified
Rationale:	Safe wakeup and sleep handling.
Use Case:	All busses with a wakeup by bus are affected.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.2 Non-functional requirements

4.2.1 LIN General

None

4.2.2 LIN Interface

None

4.2.3 Driver

None

4.2.4 LIN Transport Layer

4.2.4.1 General requirements

4.2.4.1.1 [BSW01579] Compatibility to TP of LIN specification

ID:	BSW01579
Initiator:	BMW
Date:	23.04.2008
Short Description:	The AUTOSAR LIN Transport Layer shall be based on the Diagnostic Transport Layer for LIN 2.1.
Type:	New
Importance:	High
Description:	If no requirement is explicitly added or excluded, the implementation of the AUTOSAR LIN Transport Layer shall follow the LIN 2.1 specification (chapter 3.2 in the LIN 2.1 specification). The implementation of the LIN TP is a precompiled option. LIN TP is not scalable. The LIN 2.1 specification covers the behavior of previous versions of LIN specifications.
Rationale:	Reuse of existing standards for AUTOSAR BSW. The LIN 2.1 TP specification is based on the ISO 15765-2:2003 specifications, the Diagnostic Services for CAN.
Use Case:	The typical use-case is where a Diagnostic message is handoff from CAN to LIN through the CAN/LIN master gateway ECU.
Dependencies:	--
Conflicts:	--
Supporting Material:	ISO 15765-2:2003
Contributes to:	--

4.2.4.2 Initialization

4.2.4.2.1 [BSW01540] LIN Transport Layer Initialization

ID:	BSW01540
Initiator:	BMW
Date:	30.09.2004
Short Description:	LIN Transport Layer initialization.
Type:	New
Importance:	Medium
Description:	The LIN Transport Layer shall support an API for initialization. This service shall initialize all global variables of the module and set all transport protocol connections in a default state. If there is an ongoing TP session it shall be immediately stopped.
Rationale:	--
Use Case:	--

Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.2.4.2.2 [BSW01545] LIN Transport Layer Availability

ID:	BSW01545
Initiator:	BMW
Date:	30.09.2004
Short Description:	The LIN Transport Layer services shall not be operational before initializing the module.
Type:	New
Importance:	Medium
Description:	Before using the transmission capabilities of the LIN Transport Layer, It shall be initialized. If it is not the case, the services have to return an error.
Rationale:	To avoid usage of the module without a complete initialization this could cause the transmission of corrupted frames.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.2.4.3 Normal Operation

4.2.4.3.1 [BSW01534] Concurrent connection configuration

ID:	BSW01534
Initiator:	BMW
Date:	30.09.2004
Short Description:	The AUTOSAR LIN Transport Layer shall support only single connections that are half-duplex.
Type:	New
Importance:	Medium
Description:	The LIN TP shall support only single connection (point-to-point). Single connection means the MRF/SRF-communication could not be done simultaneous between the master and several slaves, only between the master and one slave. If the TP sequence is finished, then a new TP message can be setup to another slave. The single connection is always half-duplex, meaning that a TP message is only transported in one direction at a time.
Rationale:	To guarantee a correct MRF/SRF communication.
Use Case:	A tester-tool connected to an ECU on the CAN bus sends a diagnostic request through a CAN/LIN-master ECU to a LIN slave ECU. When the request is finished the LIN slave ECU sets up a TP message with the response to the diagnostic request.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.2.4.3.2 [BSW01574] Multiple Transport Layer instances

ID:	BSW01574
Initiator:	Mentor

Date:	12.12.2005
Short Description:	Multiple Transport Layer instances.
Type:	New
Importance:	High
Description:	It shall be possible to have one instance of the TP for each channel.
Rationale:	Since the only frames that can be used for TP on LIN are the MRF and SRF it is not possible to have more than one instance of a TP message on each LIN channel.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

4.2.4.3.3 [BSW01539] Transport connection properties

ID:	BSW01539
Initiator:	BMW
Date:	30.09.2004
Short Description:	The Transport connection properties shall be statically configured.
Type:	New
Importance:	High
Description:	The LIN Transport connection configuration shall statically assign properties of each N-SDU: - Its unique handle (N_SDU_Handle) - Minimum length of the N_SDU - Associated N_PDU handle (N_PDU_Handle) - Physical (1 to 1 communication) addressing - Direction type: full-duplex or half-duplex communication - Addressing mode: standard
Rationale:	At runtime the LIN Transport module shall have all the needed information to manage a transport connection.
Use Case:	This information can be used at generation time to check the network LIN cluster configuration with a TP point of view.
Dependencies:	--
Conflicts:	--
Supporting Material:	Similar to [BSW01074] for CAN TP
Contributes to:	--

4.2.4.3.4 [BSW01591] Diagnostic transmission handler

ID:	BSW01591
Initiator:	BMW
Date:	24.04.2008
Short Description:	The LINif shall take care of the behavior of the masternode transmission handler.
Type:	New
Importance:	High
Description:	A transmission handler shall be implemented by the master node according to the chapter 5.4.4.1 in the LIN specification. This describes the following modes: - Idle state - Tx functional active - Tx physical active - Rx physical active - Interleaved functional during Tx - Interleaved functional during Rx

Rationale:	The LIN Transport layer module shall have the specified needed information to manage a transport connection.
Use Case:	This information can be used at generation time to check the network configuration with a TP point of view.
Dependencies:	--
Conflicts:	--
Supporting Material:	LIN 2.1 Specification , Chap 5.4
Contributes to:	--

4.2.4.4 Fault Operation

4.2.4.4.1 [BSW01544] Error handling

ID:	BSW01544
Initiator:	BMW
Date:	02.09.2004
Short Description:	Error handling if an unexpected N_PDU is received by a node.
Type:	New
Importance:	High
Description:	In case of reception of unexpected N_PDU it shall respect the behavior defined in chapter “unexpected arrival of network protocol data unit” of the ISO-15765-2 specification. For others errors, just aborts the segmentation session.
Rationale:	This is an extension to the LIN specification since it does not describe how to handle error situations occurred during transportation of a TP message.
Dependencies:	--
Conflicts:	--
Supporting Material:	Section “unexpected arrival of network protocol data unit“ in the ISO-15765-2 specification.
Contributes to:	--

4.3 References

4.3.1 Deliverables of AUTOSAR

[1] Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf

[2] General Requirements on Basic Software Modules
AUTOSAR_SRS_BSWGeneral.pdf

[3] General Requirements on SPAL
AUTOSAR_SRS_SPALGeneral.pdf

4.3.2 Related standards and norms

[STD_LIN_SPEC] LIN Specification Rev. 2.1
www.lin-subbus.org