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

Document Version	1.1.2
Document Status	Final
Part of Release	3.0
Revision	0001

Document Change History

Date	Version	Changed by	Change Description
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 BSW1555
24.02.2006	1.0.0	AUTOSAR Administration	Initial Release

Page left intentionally blank

Disclaimer

Any use of these specifications requires membership within the AUTOSAR Development Partnership or an agreement with the AUTOSAR Development Partnership. The AUTOSAR Development Partnership will not be liable for any use of these specifications.

Following the completion of the development of the AUTOSAR specifications commercial exploitation licenses will be made available to end users by way of written License Agreement only.

Advice to users of AUTOSAR Specification Documents:

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 2.0 specification (see 5.2). The behaviour will be restricted to a master node. It is the goal to support 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 2.0 specification.

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 2.0 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 2.0 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 2.0 master communication stack (= LIN Master functionality)
Sleep-mode	In the LIN 2.0 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 LIN General functional requirements

4.1.1 General functional requirements

4.1.1.1 [BSW01501] Usage of LIN 2.0 specification

Initiator:	Vector
Date:	14.07.2004
Short Description:	The LIN 2.0 specification shall be reused as far as possible
Type:	New
Importance:	High
Description:	The following sections in the LIN 2.0 specification shall be reused LIN 2.0 Protocol Specification: - Chapter 2. Frame Transfer - Chapter 3. Schedules - Chapter 4. Task Behaviour Model - Chapter 5. Network Management - Chapter 6. Status Management LIN 2.0 Diagnostic and Configuration Specification: - Chapter 2: Node Configuration - Chapter 3.3: Diagnostics Transport Layer The remaining chapters in the LIN 2.0 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 2.0 specification (e.g. in the configuration): - The Diagnostic Transport Protocol is optional in the LIN 2.0 specification, in AUTOSAR it is a precompiled option. (see [BSW1533]) - All other optional functionality is decided by the design. The following item shall be used with AUTOSAR adaptations: - Application Program Interface Specification The [BSW1567] will take care that the Lin Interface is Compatible with the LIN 2.0 Specification, also [BSW1565] for the LIN Driver and [BSW1533] for the Lin TP.
Rationale:	Reuse of existing standards. This ensures the reusability of LIN slave ECU's inside the vehicle architecture. Each LIN Master will support both checksum models, the enhanced checksum for LIN 2.0 and the classic checksum for LIN 1.3 slaves (see LIN 2.0 protocol specification chapter 2.1.5 Checksum) The LIN 2.0 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:	[BSW1567], [BSW1565], [BSW1533]
Conflicts:	--
Supporting Material:	LIN 2.0 specification

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

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 [BSW01501]. It is in fact a further specialization of that requirement.
Supporting Material:	--

4.1.2 Initialization

None

4.1.3 Normal Operation

4.1.3.1 [BSW01522] Consistent data transfer

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:	--

4.1.4 Shutdown Operation

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

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, this channel 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 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:	--

4.1.5 Fault Operation

None

4.2 LIN General Non-functional requirements

None

4.3 LIN Interface Functional requirements

4.3.1 General requirements

4.3.1.1 [BSW01567] Compatibility to LIN 2.0 protocol specification

Initiator:	Mentor
Date:	12.12.2005
Short Description:	Compatibility to LIN 2.0 protocol specification
Type:	New
Importance:	High
Description:	The following sections of the LIN 2.0 specification shall be reused by the LIN Interface: LIN 2.0 Protocol Specification: - Chapter 2. Frame Transfer (handling of different types of LIN-PDUs, not the specific bytes of the LIN-PDU) - Chapter 3. Schedules - Chapter 4. Task Behaviour Model (handling of LIN-PDUs and errors. Not handling specific bytes in the LIN-PDU) - Chapter 5. Network Management - Chapter 6. Status Management
Rationale:	Basic LIN functionality
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	LIN 2.0 protocol specification

4.3.1.2 [BSW01551] Multiple LIN channel support for interface

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:	--

4.3.1.3 [BSW01568] Hardware Independence

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:	--

4.3.2 Initialization

4.3.2.1 [BSW01569] LIN Interface initialization

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 2 LIN channels than there are also 2 different LDF-files.
Dependencies:	--
Conflicts:	--
Supporting Material:	Comparing with LIN 2.0 specification API's, the LIN Interface init will do both work of l_ifc_init and l_sys_init

4.3.2.2 [BSW01570] Selection of static configuration sets

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:	--

4.3.3 Normal Operation

4.3.3.1 [BSW01564] Schedule Table Manager

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 2.0 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:	--

4.3.3.2 [BSW01546] Schedule Table Handler

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 2.0 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 2.0 specification, see LIN API l_sch_tick.

4.3.3.3 [BSW01561] Main function

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:	--

4.3.3.4 [BSW01549] Timer service for Scheduling

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 2.0 Configuration Language specification.

4.3.3.5 [BSW01571] Transmission request service

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 2.0 Protocol specification chapter 2.3.3 Sporadic Frame.

4.3.3.6 [BSW01514] Wake-up notification support

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

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

Initiator:	Vector
Date:	14.07.2004
Short Description:	The LIN Interface shall provide an API to wake-up a LIN channel
Type:	New
Importance:	Medium
Description:	The LIN Interface shall provide an API to wake-up a LIN channel. The LIN Interface shall support that each LIN channel 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 2.0 API specification, chapter 2.5.5 l_ifc_wake_up and LIN 2.0 Protocol Specification, chapter 5.1 Wake-up

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

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:	--

4.3.3.9 [BSW01558] Check successful communication

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 shall conform to the LIN 2.0 test specification.

	*) see LIN 2.0 Spec., Status Management, Page 18, Chapt.6.4, “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:	--

4.3.4 Shutdown Operation

None

4.3.5 Fault Operation

4.3.5.1 [BSW01527] Notification for missing or faulty receive LIN-PDU

Initiator:	CAS
Date:	03.08.2004
Short Description:	Notify the PDU router of errors in received response field
Type:	New
Importance:	High
Description:	The LIN Interface shall provide an error notification for the upper layer if the response part of the LIN-PDU is not sent by the LIN slave or if the response part (from master or slave) contains errors: An error here is exactly the same as described in the LIN specification: • A mismatch between read-back and sent data • Checksum error There is an exception. If the LIN-PDU is an event-triggered frame an error (as described above) is not regarded as an error.
Rationale:	--
Use Case:	--
Dependencies:	--
Conflicts:	No
Supporting Material:	This corresponds to the response error in the LIN 2.0 specification (Protocol specification chapter 6, Status management) and the Task Behavior Model (LIN 2.0 Protocol Specification chapter 4).

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

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 2.0 specification

4.4 LIN Interface Non-functional requirements

None

4.5 LIN Driver Functional requirements

4.5.1 General requirements

4.5.1.1 [BSW01565] Compatibility to LIN 2.0 protocol specification

Initiator:	WP4.2.2.1.1
Date:	23.11.2005
Short Description:	Compatibility to LIN 2.0 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 4 in the LIN 2.0 Protocol specification) is part of the LIN Driver.
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 2.0 protocol specification

4.5.1.2 [BSW01553] Basic Software SPAL General requirements

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]

4.5.1.3 [BSW01552] Hardware abstraction LIN

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:	--

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

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:	--

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

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:	--

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

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:	--

4.5.2 Initialization

4.5.2.1 [BSW01572] LIN driver initialization

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:	--

4.5.2.2 [BSW01573] Selection of static configuration sets

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:	--

4.5.3 Normal Operation

4.5.3.1 [BSW01563] Wake-up Notification

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 Break byte 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:	--

4.5.3.2 [BSW01556] Multiple LIN channel support for driver

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:	--

4.5.4 Shutdown Operations

4.5.4.1 [BSW01566] Transition to sleep-mode

Initiator:	WP4.2.2.1.1
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:	--

4.5.4.2 [BSW01524] Support of reduced power operation mode

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:	--

4.5.5 Fault Operation

4.5.5.1 [BSW01526] Error notification

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 2.0 API specification.

4.6 Driver Non-functional requirements

None

4.7 LIN Transport Layer Functional requirements

4.7.1 General requirements

4.7.1.1 [BSW01533] Compatibility to TP of LIN 2.0 specification

Initiator:	BMW
Date:	30.09.2004
Short Description:	The AUTOSAR LIN Transport Layer shall be based on the Diagnostic Transport Layer for LIN 2.0.
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 2.0 specification (chapter 3.3 in the LIN 2.0 Protocol specification). The implementation of the LIN TP is a precompiled option. LIN TP is not scaleable.
Rationale:	Reuse of existing standards for AUTOSAR BSW. The LIN 2.0 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

4.7.2 Initialization

4.7.2.1 [BSW01540] LIN Transport Layer Initialization

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:	--

4.7.2.2 [BSW01545] LIN Transport Layer Availability

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:	--

4.7.3 Normal Operation

4.7.3.1 [BSW01534] Concurrent connection configuration

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:	--

4.7.3.2 [BSW01574] Multiple Transport Layer instances

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:	--

4.7.3.3 [BSW01539] Transport connection properties

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 configuration with a TP point of view.
Dependencies:	--
Conflicts:	--
Supporting Material:	Similar to [BSW01074] for CAN TP

4.7.4 Fault Operation

4.7.4.1 [BSW01544] Error handling

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 2.0 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.

5 References

5.1 Deliverables of AUTOSAR

[1] Layered Software Architecture
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_LayeredSoftwareArchitecture.pdf

[2] General Requirements on Basic Software Modules
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_SRS_General.pdf

[3] General Requirements on SPAL
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_SRS_SPAL_General.pdf

5.2 Related standards and norms

[STD_LIN_SPEC] LIN Specification Rev. 2.0
www.lin-subbus.org