Document Title	SRS_LIN: Complete Change Documentation 4.3.0 - 4.3.1
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	695

Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.3.1

Table of Contents

1 SRS_LIN

1.1 Specification Item SRS_Lin_01514

Trace References:

RS_BRF_01768, RS_BRF_01104, RS_BRF_01064

Content:

Type:	
Description:	The LIN Interface shall inform an upper layer if a wake-up request was notified by the underlying LIN Driver.
Rationale:	Basic functionality
Applies to:	
Use Case:	Wakeup of ECU by LIN. Inform upper layer (ECU State Manager) about the wakeup reason
Supporting Material:	ECU state manager

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73095: [LinSM][LinIf][Lin] Inconsistent LIN channel states upon initialization

  Problem description:

  As per SWS_Lin_00171, LIN channel state upon initialization shall be LIN_CH_OPERATIONAL

  As per SWS_LinIf_00507, LinIf channel state upon initialization shall be either LINIF_CHANNEL_OPERATIONAL or LINIF_CHANNEL_SLEEP depending on the configuration parameter LinIfStartupState

  As per SWS_LinSM_00152, LinSM state for all networks upon initialization shall be LINSM_NO_COM

  This is an inconsistent state within the LIN Stack since LIN driver channel is operational but LinSM is in NO_COM and LinIf state could be either!

  Tracking the history of LinIf (Bug 27547) it seems that when LinIfStartupState was introduced, then Bug 27547 comment 8 was not addressed at that time to avoid this inconsistency.

  Is there a use case to start the LIN channels in OPERATIONAL?
  Should the LinSM already transition the channels to SLEEP following initialization? (similar to CanSM)

Or should the Lin driver specifications be updated to allow for either state following initialization (similar to LinIf and LinTrcv specifications)

Agreed solution:

================
For CP R4.3.1
================

SRS LIN
==
1) [SRS_Lin_01514], Description: Change
from
<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.
to
>The LIN Interface shall inform an upper layer if a wake-up request was notified by the underlying LIN Driver.

SWS LinIf
==
1) Sec. 4.1 Limitations: Add following description
+It's assumed that all of connected LIN Slave ECUs can receive a wakeup frame when they are already operational (as LIN Master ECU starts with LINIF_CHANNEL_SLEEP state).

2) Chap. 7, [SWS_LinIf_00507]: Change
from
<[SWS_LinIf_00507] The LIN Interface shall transit from LINIF_UNINIT to
< LINIF_CHANNEL_OPERATIONAL when the function LinIf_Init is called and the configuration parameter LinIfStartupState is set to NORMAL.
< LINIF_CHANNEL_SLEEP when the function LinIf_Init is called and the configuration parameter LinIfStartupState is set to SLEEP. ()
to
>[SWS_LinIf_00507] The LIN Interface shall transit from LINIF_UNINIT to LINIF_CHANNEL_SLEEP without sending go-to-sleep command, when the function LinIf_Init is called. ()
>Note: it is assumed that automatically slave nodes will enter bus sleep mode earliest after 4s and latest 10s of bus inactivity (as specified in LIN 2.1).

3) Chap. 7, Figure 4 (stm LinIf Node Management): Update the figure as de-

scribed below
-Remove the transition to LINIF_CHANNEL_OPERATIONAL with LinIf_Init [LinIfStartupState=NORMAL]
-Remove the transition to LINIF_CHANNEL_SLEEP with LinIf_Init [LinIfStartupState=SLEEP]
+Add the initial transition to LINIF_CHANNEL_SLEEP inside the state LINIF_INIT

4) Chap. 10, ECUC_LinIf_00069: Set the parameter LinIfStartupState to obsolete

5) Chap. 10, Figure 20: Remove the parameter LinIfStartupState

SWS LinTrcv
===
Chapter 7.1
Update figure: remove transitions from NOT_ACTIVE to LINTRCV_TRCV_MODE_STANDBY and LINTRCV_TRCV_MODE_NORMAL.
Remove the following sentence:
The function LinTrcv_Init() causes a state change to either LINTRCV_TRCV_MODE_SLEEP, LINTRCV_TRCV_MODE_NORMAL or LINTRCV_TRCV_MODE_STANDBY (any of these 3 states belong to the upper state ACTIVE). This depends on the configuration and is independent configurable for each channel.

Change SWS_LinTrcv_00119 from
The function LinTrcv_Init shall set the LIN transceiver hardware to the state configured by the configuration parameter LinTrcvInitState.
This can be LINTRCV_TRCV_MODE_NORMAL, LINTRCV_TRCV_MODE_STANDBY or LINTRCV_TRCV_MODE_SLEEP.
to
The function LinTrcv_Init shall set the LIN transceiver hardware to the state LINTRCV_TRCV_MODE_SLEEP.

Remove SWS_LinTrcv_00146.
remove configurations note after SWS_LinTrcv_00146

set ECUC_LinTrcv_00005 to obsolete.

SWS Lin
===
- Change SWS_Lin_00171 to: "On entering the state LIN_INIT, the Lin module

shall set each channel into state LIN_CH_SLEEP, enable the wake-up detection (if enabled by LinChannelWakeupSupport), and optionally set the LIN hardware unit to reduced power operation mode (if supported by HW)."
- Change initial transition in figure 7-1 to LIN_CH_SLEEP.
- SWS_Lin_00168 Lin_GetStatus: Change LIN_OPERATIONAL description to "Normal operation; the related LIN channel is woken up from the LIN_CH_SLEEP and no data has been sent.

SWS LinSM
==
no changes required

TPS System Template
==
1) Appendix C: Remove the rule of the parameter LinIfStartupState
–Last change on issue 73095 comment 46–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.2 Specification Item SRS_Lin_01561

Trace References:

RS_BRF_01768, RS_BRF_01056

Content:

Type:	Valid
Description:	The main function is responsible for executing the schedule table handler
Rationale:	–
Applies to:	
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.
Supporting Material:	–

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #78796: CP_R4.3.1 Document finalization

Problem description:

1) The change history entry needs to be updated for CP Release 4.3.1.
2) For Word documents: A document shall meet formal requirements before releasing.
3) Hiding of internal document change history
4) For MOD files: Adaption of "_readme.txt"
5) For documents delivered as ZIP archive: Usage of current AUTOSAR XML schema in every arxml file

Agreed solution:

In case that your document has not been changed since last release, please leave a short comment in your implementation task and set it to closed/invalid.

Please note: the following changes shall be done when all the changes of your document are clear (for R4.3.1)!

ad 1)
Please insert a change history entry for document finalization at the beginning of the Document Change History table in your document and use the following content:

| Date |Release|Changed by |Change Description|
|2017-12-08| 4.3.1 |AUTOSAR Release Management| [cd] |

A) In case your document has been changed since last release, the Change Description [cd] comprises up to 4 bullet points describing the major changes from Release 4.3.0 to Release 4.3.1. The description shall contain no references to Bugzilla issue numbers. This text will make up the Change History of the to-be-released document. Please
state major changes with a high impact (e.g. high BWC or usage) or
put this standard entry: "minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation"

B) In case your document will be released for the first time, please use as Change Description [cd] the entry "Initial release".

C) In case your document has been already released (in another standard), please put as first line of Change Description [cd] the following:
–Migration of document to standard "Classic Platform"–

ad 2)

*** For Word documents ***
Please check your document once more against this checklist:
https://svn.autosar.org/repos/work/02_ProjectOrganization/06_Templates/AUTOSAR_Checklist_F
Please verify also the PDF file as this is the one which will be released (it is automatically generated right next to the Word source document within four hours).

ad 3)
If not already done, please hide the whole internal document change history.
Please check the PDF file and ensure that nothing of the internal document change history is shown. Especially in case of remaining bullet points, please follow this instruction to get rid of those in your Word document, as well:
https://www.autosar.org/wiki/doku.php?id=collab_guide:hide_internal_changehistory_man

ad 4)
*** For MOD documents only ***
Please create or adapt the folder "contents". That folder shall include all files which will be used for the release R4.3.1 (zip-archive). That folder shall also include the two files: "_disclaimer.txt" and "_readme.txt"

The file "_readme.txt" shall include (besides existing information):
- Document Title: DOC LONG NAME
- Document Owner: AUTOSAR
- Document Responsibility: AUTOSAR
- Document Identification Number: UID
- Document Status: Final
- Part of AUTOSAR Standard: Classic Platform
- Part of AUTOSAR Release: 4.3.1
- Date: 2017-12-08

ad 5)
*** For documents delivered as ZIP archive ***
In case the archive contains arxml files, please check whether every arxml file references the current AUTOSAR XML schema:
xsi:schemaLocation="http://autosar.org/schema/r4.0 AUTOSAR_00044.xsd"

If another schema location is given, please contact technical.office@autosar.org.

BW-C-Level:

Application	Specification	Bus
1	1	1

1.3 Specification Item SRS_Lin_01564

Trace References:

RS_BRF_01768, RS_BRF_01592

Content:

Type:	
Description:	The schedule table manager will keep schedule table to execute. The schedule table manager shall:
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.
Applies to:	
Use Case:	Example system start:
Supporting Material:	–

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76265: [LIN]Conflicting information in SRS_Lin_01564

  Problem description:

  Inside [SRS_Lin_01564], There are statements that are conflicting to LinIf specification.

  There is no parameter in LinIf/LinSM to assign priority for any schedule table, these requirements are coming from a version where ScheduleTablePriority was supported.
  From AR4.0 onwards this is handled inside BSWM.

  Remove the below statements from SRS_Lin_01564-

  "Each one time schedule table shall have a unique priority per channel"
  "Change to a specific requested schedule table depending on priority"

  "The priority of the schedule tables is preconfigured."

  "the priority is always 1.
  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."

"Prioritize the requested schedule tables (immediately, or later)"

Agreed solution:

1) Change the Description of [SRS_Lin_01564]:
from
< 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 a newer request, the priority is always 1.
< 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.
to
> 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
> * Execute a schedule table once or continuously
> One or more modules from an upper layer will create a sequence of schedule tables and request the schedule table manager to execute specific schedule tables. Priority handling is not handled inside LinStack.
> The schedule table manager will only coordinate the running schedule table and schedule table requests.
> There exist one memory space for the "continuously execution schedule table", it will be overwritten by a newer request.
–Last change on issue 76265 comment 19–

BW-C-Level:

Application	Specification	Bus
1	1	1