

|                                   |                                    |
|-----------------------------------|------------------------------------|
| <b>Document Title</b>             | Specification of LIN State Manager |
| <b>Document Owner</b>             | AUTOSAR                            |
| <b>Document Responsibility</b>    | AUTOSAR                            |
| <b>Document Identification No</b> | 255                                |

|                                 |                  |
|---------------------------------|------------------|
| <b>Document Status</b>          | published        |
| <b>Part of AUTOSAR Standard</b> | Classic Platform |
| <b>Part of Standard Release</b> | R24-11           |

| Document Change History |         |                            |   |
|-------------------------|---------|----------------------------|---|
| Date                    | Release | Changed by                 | Description   |
| 2024-11-27              | R24-11  | AUTOSAR Release Management | <ul style="list-style-type: none"> <li>Editorial Changes</li> <li>removed uptrace to [SRS_BSW_00334]</li> </ul>   |
| 2023-11-23              | R23-11  | AUTOSAR Release Management | <ul style="list-style-type: none"> <li>Updated Chapter 7.1.8</li> <li>Updated Chapter 8.3.2</li> <li>Removed Chapter 7.1.8.1 Wakeup repetitions for slave</li> <li>Editorial Changes in Chapter 8.5.1</li> <li>Editorial Changes in Chapter 8.5.2</li> <li>Corrected Figure in Chapter 9.1</li> <li>Added new parameter <a href="#">[ECUC_LinSM_00212]</a> in Chapter 10.3.2</li> </ul> |
| 2022-11-24              | R22-11  | AUTOSAR Release Management | <ul style="list-style-type: none"> <li>Changed ID from [SWS_LinSM_00211] to <a href="#">[SWS_LinSM_NA_00211]</a></li> <li>Renamed the arguments “Schedule” (<a href="#">[SWS_LinSM_00113]</a> LinSM_ScheduleRequest) and “schedule” (<a href="#">[SWS_LinSM_00129]</a> LinSM_ScheduleRequestConfirmation)</li> </ul>  |
| 2021-11-25              | R21-11  | AUTOSAR Release Management | <ul style="list-style-type: none"> <li>Corrected Figure 7 and <a href="#">[SWS_LinSM_00233]</a></li> </ul>  |





|            |        |                            |  |
|------------|--------|----------------------------|--|
| 2020-11-30 | R20-11 | AUTOSAR Release Management | <ul style="list-style-type: none"> <li>• Cleanup error sections in chapter 7</li> </ul>  |
| 2019-11-28 | R19-11 | AUTOSAR Release Management | <ul style="list-style-type: none"> <li>• Editorial changes</li> <li>• Changed Document Status from Final to published</li> </ul>   |
| 2018-10-31 | 4.4.0  | AUTOSAR Release Management | <ul style="list-style-type: none"> <li>• LIN Slave support (CONC 631)</li> <li>• Replaced references to Lin 2.1 by ISO 17987:2016</li> <li>• Editorial changes</li> </ul>                              |
| 2017-12-08 | 4.3.1  | AUTOSAR Release Management | <ul style="list-style-type: none"> <li>• LINSM E CONFIRMATION TIMEOUT changed to Runtime Error</li> </ul>  |
| 2016-11-30 | 4.3.0  | AUTOSAR Release Management | <ul style="list-style-type: none"> <li>• Editorial changes</li> </ul>  |
| 2015-07-31 | 4.2.2  | AUTOSAR Release Management | <ul style="list-style-type: none"> <li>• Modified header file structure</li> <li>• Debugging support marked as obsolete</li> <li>• Editorial changes</li> </ul>  |
| 2014-10-31 | 4.2.1  | AUTOSAR Release Management | <ul style="list-style-type: none"> <li>• Removed NULL pointer check requirement and moved to BSW General</li> <li>• Corrections in ECU parameter configuration</li> </ul>                              |
| 2014-03-31 | 4.1.3  | AUTOSAR Release Management | <ul style="list-style-type: none"> <li>• Editorial changes</li> </ul>  |
| 2013-10-31 | 4.1.2  | AUTOSAR Release Management | <ul style="list-style-type: none"> <li>• Minor bug fixes</li> <li>• Editorial changes</li> <li>• Removed chapter(s) on change documentation</li> </ul>   |
| 2013-03-15 | 4.1.1  | AUTOSAR Administration     | <ul style="list-style-type: none"> <li>• LIN wakeup and sleep mode handling corrected</li> </ul>   |
| 2011-12-22 | 4.0.3  | AUTOSAR Administration     | <ul style="list-style-type: none"> <li>• Added post-build configuration support</li> <li>• Added completion of Production error concept in Com Stack</li> <li>• Removed local network index</li> </ul> |



△

|            |       |                        |  |
|------------|-------|------------------------|--|
| 2010-09-30 | 3.1.5 | AUTOSAR Administration | <ul style="list-style-type: none"> <li>• Post-build configuration variant added</li> <li>• Module version check changed according SRS General SRS BSW 00004</li> <li>• TrcvModeType definition moved from LinIf to LinTrcv</li> </ul>  |
| 2010-02-02 | 3.1.4 | AUTOSAR Administration | <ul style="list-style-type: none"> <li>• Controlling of I-PDU groups has been moved to the BSW Mode Manager module</li> <li>• Interface to the LIN Transceiver module has been introduced since LIN Transceiver driver is a new module in release 4.0</li> <li>• Legal disclaimer revised</li> </ul> |
| 2008-08-13 | 3.1.1 | AUTOSAR Administration | <ul style="list-style-type: none"> <li>• Legal disclaimer revised</li> </ul>   |
| 2007-12-21 | 3.0.1 | AUTOSAR Administration | <ul style="list-style-type: none"> <li>• Initial Release</li> </ul>  |

## Disclaimer

This work (specification and/or software implementation) 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 work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work 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 work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

## Contents

|         |   |    |
|---------|---|----|
| 1       | Introduction and functional overview              | 8  |
| 1.1     | Architectural overview                            | 8  |
| 1.2     | Functional overview                               | 9  |
| 2       | Acronyms and Abbreviations                        | 10 |
| 3       | Related documentation                             | 11 |
| 3.1     | Input documents & related standards and norms     | 11 |
| 3.2     | Related specification                             | 11 |
| 4       | Constraints and assumptions                       | 12 |
| 4.1     | Limitations                                       | 12 |
| 4.2     | Applicability to car domains                      | 12 |
| 5       | Dependencies to other modules                     | 13 |
| 5.1     | Relation to Upper layers                          | 14 |
| 5.1.1   | Operating System                                  | 14 |
| 5.1.2   | Module DET (Default Error Tracer)                 | 15 |
| 5.1.3   | Module DEM (Diagnostic Event Manager)             | 15 |
| 5.1.4   | ComM  | 15 |
| 5.1.5   | BSW Mode Manager                                  | 15 |
| 5.2     | Relation to Lower layers                          | 15 |
| 5.2.1   | LinIf   | 15 |
| 5.3     | File structure                                    | 16 |
| 5.3.1   | Code file structure                               | 16 |
| 5.3.2   | Header File structure                             | 16 |
| 5.3.2.1 | LinSM header files                                | 16 |
| 5.3.2.2 | Included header files                             | 16 |
| 6       | Requirements Tracing                              | 18 |
| 7       | Functional specification                          | 19 |
| 7.1     | States and transitions of the LinSM state machine | 19 |
| 7.1.1   | LINSM_UNINIT                                      | 22 |
| 7.1.2   | LINSM_INIT  | 22 |
| 7.1.3   | LINSM_NO_COM                                      | 23 |
| 7.1.4   | LINSM_FULL_COM                                    | 24 |
| 7.1.5   | Goto sleep  | 24 |
| 7.1.5.1 | Goto sleep specific for master node               | 25 |
| 7.1.5.2 | Goto sleep specific for slave node                | 26 |
| 7.1.6   | Changing schedule table (Master only)             | 26 |
| 7.1.7   | Wake up process                                   | 27 |
| 7.1.8   | Timeout of requests                               | 28 |
| 7.2     | Handling multiple networks and drivers            | 29 |
| 7.2.1   | Multiple networks                                 | 30 |

|        |   |    |
|--------|---|----|
| 7.3    | Error Classification                    | 30 |
| 7.3.1  | Development Errors                      | 30 |
| 7.3.2  | Runtime Errors                          | 31 |
| 7.3.3  | Production Errors                       | 31 |
| 7.3.4  | Extended Production Errors              | 31 |
| 8      | API specification                       | 32 |
| 8.1    | Imported types                          | 32 |
| 8.1.1  | Standard types                          | 32 |
| 8.2    | Type definitions                        | 32 |
| 8.2.1  | LinSM_ModeType                          | 32 |
| 8.2.2  | LinSM_ConfigType                        | 33 |
| 8.3    | LinSM API                               | 33 |
| 8.3.1  | LinSM_Init                              | 33 |
| 8.3.2  | LinSM_ScheduleRequest                   | 34 |
| 8.3.3  | LinSM_GetVersionInfo                    | 35 |
| 8.3.4  | LinSM_GetCurrentComMode                 | 36 |
| 8.3.5  | LinSM_RequestComMode                    | 37 |
| 8.4    | Scheduled functions                     | 38 |
| 8.4.1  | LinSM_MainFunction                      | 38 |
| 8.5    | LinSM callbacks                         | 39 |
| 8.5.1  | LinSM_ScheduleRequestConfirmation       | 39 |
| 8.5.2  | LinSM_GotoSleepIndication               | 40 |
| 8.5.3  | LinSM_GotoSleepConfirmation             | 41 |
| 8.5.4  | LinSM_WakeupConfirmation                | 42 |
| 8.6    | Mandatory Interfaces                    | 42 |
| 8.7    | Optional Interfaces                     | 43 |
| 8.8    | Configurable Interfaces                 | 44 |
| 9      | Sequence diagrams                       | 45 |
| 9.1    | Goto-sleep process                      | 45 |
| 9.1.1  | Master                                  | 45 |
| 9.1.2  | Slave                                   | 46 |
| 9.2    | Internal wake up                        | 46 |
| 9.3    | Schedule switch (Master only)           | 47 |
| 10     | Configuration specification             | 49 |
| 10.1   | How to read this chapter                | 49 |
| 10.2   | Containers and configuration parameters | 49 |
| 10.2.1 | Configuration Tool                      | 49 |
| 10.3   | LinSM_Configuration                     | 49 |
| 10.3.1 | LinSM                                   | 50 |
| 10.3.2 | LinSMConfigSet                          | 50 |
| 10.3.3 | LinSMChannel                            | 52 |
| 10.3.4 | LinSMGeneral                            | 54 |
| 10.3.5 | LinSMSchedule                           | 56 |
| 10.4   | Published Information                   | 57 |

|       |   |    |
|-------|---|----|
| A     | Not applicable requirements   | 58 |
| A.1   | Change history of AUTOSAR traceable items                                   | 58 |
| A.2   | Traceable item history of this document according to AUTOSAR Release R22-11 | 58 |
| A.2.1 | Added Specification Items in R22-11   | 58 |
| A.2.2 | Changed Specification Items in R22-11                                       | 59 |
| A.2.3 | Deleted Specification Items in R22-11                                       | 59 |
| A.3   | Traceable item history of this document according to AUTOSAR Release R23-11 | 59 |
| A.3.1 | Added Specification Items in R23-11   | 59 |
| A.3.2 | Changed Specification Items in R23-11                                       | 59 |
| A.3.3 | Deleted Specification Items in R23-11                                       | 60 |
| A.4   | Traceable item history of this document according to AUTOSAR Release R24-11 | 60 |
| A.4.1 | Added Specification Items in R24-11   | 60 |
| A.4.2 | Changed Specification Items in R24-11                                       | 60 |
| A.4.3 | Deleted Specification Items in R24-11                                       | 60 |

# 1 Introduction and functional overview

This specification specifies the functionality, API and the configuration of the AUTOSAR Basic Software module LIN State Manager (LinSM). The LinSM together with the LIN Interface, LIN driver, LIN Transceiver driver forms the complete LIN protocol.

The LIN State Manager is designed to be hardware independent.

The LinSM is dependent on upper module Communication Manager [1](ComM) and lower module LIN Interface [2] (LinIf).

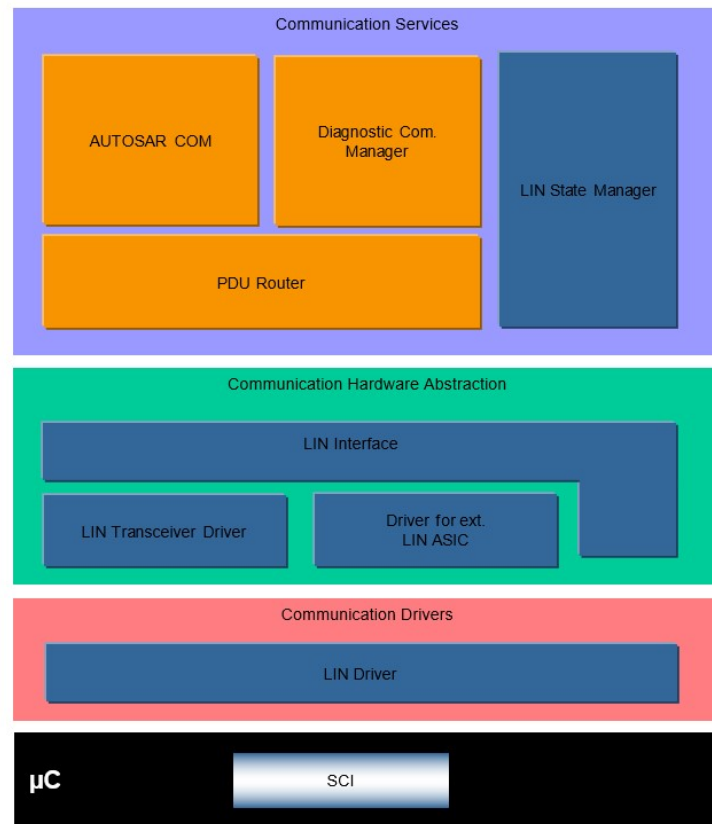
It is assumed that the reader is familiar with the ISO 17987 specification. This document will not describe functionality already described in the ISO 17987 specification.

Note that figures in this document are not regarded as requirements. All requirements are described in text prefixed with a requirement tag (e.g. LINSM042). Text not prefixed with a requirement shall be seen as informative text.

## 1.1 Architectural overview

The Layered Software Architecture [3, Layered Software Architecture] positions the Lin SM within the BSW architecture as shown below.





**Figure 1.1: AUTOSAR BSW software architecture - LIN stack scope**

## 1.2 Functional overview

The LinSM is responsible for the control flow of the LIN Bus.

This means:

- Switching schedule tables when requested by the upper layer(s) (for LIN master node only).
- Go to sleep and wake up handling, when requested by the upper layer(s) or indicated by the lower layer(s)
- Notification to upper layers when new state is entered.

## 2 Acronyms and Abbreviations

Acronyms and abbreviations used in this document. Additional abbreviations can be found in the ISO 17987 specification.

| Abbreviation / Acronym: | Description:  |
|-------------------------|---|
| API                     | Application Program Interface   |
| AUTOSAR                 | Automotive Open System Architecture   |
| BSW                     | Basic Software  |
| BswM                    | BSW Mode Manager  |
| ComM                    | Communication Manager   |
| DCM                     | Diagnostic Communication Manager  |
| DEM                     | Diagnostic Event Manager  |
| DET                     | Default Error Tracer  |
| ECU                     | Electric Control Unit   |
| ID                      | Identifier  |
| ISR                     | Interrupt Service Routine   |
| Jitter                  | Difference between longest delay and shortest delay (e.g. Worst case execution time - Best case execution time) |
| LIN                     | Local Interconnect Network  |
| LinIf                   | LIN Interface   |
| LinSM                   | LIN State Manager (the subject of this document)  |
| MCAL                    | Microcontroller Abstraction Layer   |
| PDU                     | Protocol Data Unit  |
| RAM                     | Random Access memory  |
| RTE                     | Run Time Environment  |
| RX                      | Receive   |
| SPAL                    | Standard Peripheral Abstraction Layer   |
| SRS                     | Software Requirement Specification  |
| SW                      | Software  |
| SWS                     | Software Design Specification   |
| TP                      | Transport Protocol  |
| TX                      | Transmit  |
| UART                    | Universal Asynchronous Receiver Transmitter   |
| UML                     | Universal Modelling Language  |
| URL                     | Uniform Resource Locator  |
| WP11                    | Work Package in AUTOSAR phase 2   |
| XML                     | Extensible Markup Language  |

## 3 Related documentation

### 3.1 Input documents & related standards and norms

- [1] Specification of Communication Manager  
AUTOSAR\_CP\_SWS\_COMManager
- [2] Specification of LIN Interface  
AUTOSAR\_CP\_SWS\_LINInterface
- [3] General Requirements on Basic Software Modules  
AUTOSAR\_CP\_RS\_BSWGeneral
- [4] General Specification of Basic Software Modules  
AUTOSAR\_CP\_SWS\_BSWGeneral
- [5] Specification of LIN Driver  
AUTOSAR\_CP\_SWS\_LINDriver
- [6] Specification of Default Error Tracer  
AUTOSAR\_CP\_SWS\_DefaultErrorTracer
- [7] Specification of Diagnostic Event Manager  
AUTOSAR\_CP\_SWS\_DiagnosticEventManager
- [8] Specification of Basic Software Mode Manager  
AUTOSAR\_CP\_SWS\_BSWModeManager
- [9] Requirements on LIN  
AUTOSAR\_CP\_RS\_LIN
- [10] Layered Software Architecture  
AUTOSAR\_CP\_EXP\_LayeredSoftwareArchitecture
- [11] Specification of Standard Types  
AUTOSAR\_CP\_SWS\_StandardTypes

### 3.2 Related specification

AUTOSAR provides a General Specification on Basic Software modules [4, SWS BSW General], which is also valid for Lin State Manager.

Thus, the specification SWS BSW General shall be considered as additional and required specification for Lin State Manager.

## 4 Constraints and assumptions

### 4.1 Limitations

There is at most one instance of the LinSM in each ECU. If the underlying LIN Driver [5] supports multiple networks, the LinSM may be LIN master or LIN slave on more than one cluster.

All references to (switching of) schedule tables do only apply to LIN master node; there are no schedule tables for LIN slave node.

### 4.2 Applicability to car domains

This specification is applicable to all car domains, where LIN is used.

## 5 Dependencies to other modules

This section describes the relations to other modules within the basic software. It describes the services that are used from these modules. Figure 5.1 shows the modules that are required or optional for the realization of the LinSM module. The figure is complete but is not regarded as requirement.

To be able for the LinSM module to operate the following modules will be interfaced:

### [SWS\_LinSM\_00001]

*Upstream requirements:* [SRS\\_BSW\\_00384](#)

[LIN Interface - LinIf]

### [SWS\_LinSM\_00085]

*Upstream requirements:* [SRS\\_BSW\\_00384](#)

[Diagnostic Event Manager - DEM]

### [SWS\_LinSM\_00086]

*Upstream requirements:* [SRS\\_BSW\\_00384](#)

[Default Error Tracer - DET]

### [SWS\_LinSM\_00105]

*Upstream requirements:* [SRS\\_BSW\\_00384](#)

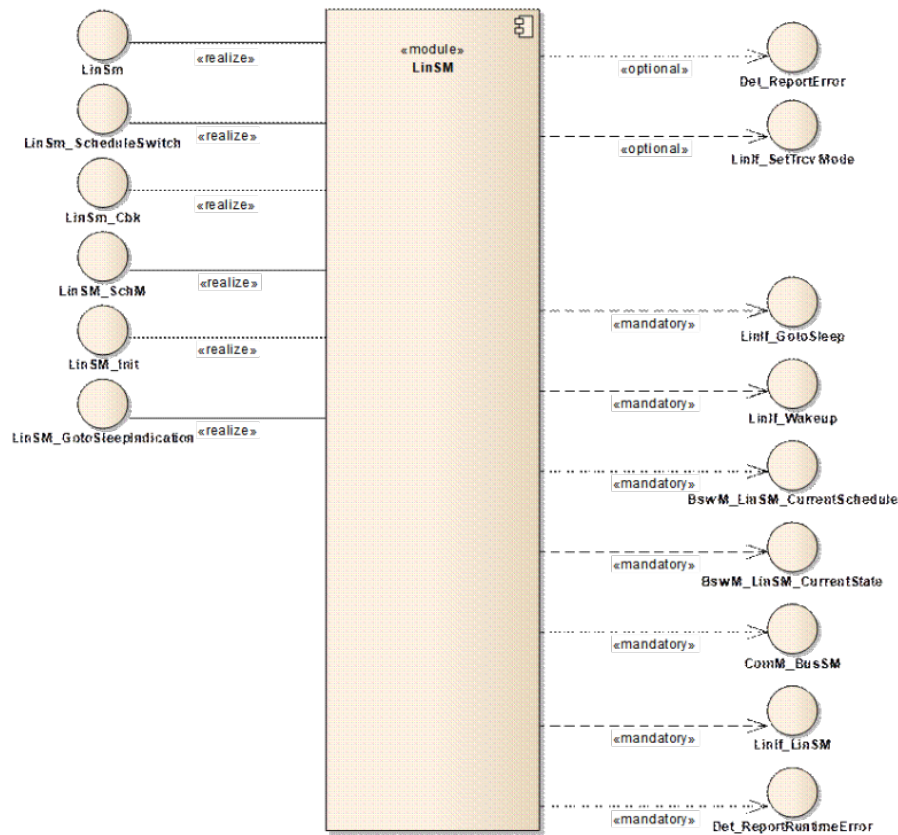
[Communication Manager - ComM]

### [SWS\_LinSM\_00196]

*Upstream requirements:* [SRS\\_BSW\\_00384](#)

[BSW Mode Manager - BswM]

Note that modules that are using the interface (except callbacks) from this module are not listed.



**Figure 5.1: Dependencies to other modules**

## 5.1 Relation to Upper layers

In principle, there is no requirement that specific modules shall call the interfaces of the LinSM module. Below, the normal users of LinSM module are listed.

### 5.1.1 Operating System

The LinSM module does contain access of shared data with above or below modules (using the API). The data that is shared will not need a help of the OS to protect the data for consistency (there are no array accesses, only simple type accesses). However, there may be reentrant functions that access the same data in the LinSM module. It is up to the implementer to solve these accesses.

### 5.1.2 Module DET (Default Error Tracer)

The Det\_ReportError - function of module DET [6] will be called for development and runtime errors.

### 5.1.3 Module DEM (Diagnostic Event Manager)

Production errors will be reported to the Diagnostic Event Manager [7] module.

### 5.1.4 ComM

The Com manager module requests the communication via the LIN stack and queries the state of the LinSM module.

### 5.1.5 BSW Mode Manager

The LinSM module will notify the BSW Mode Manager module [8] when a state is changed. The BSW Mode Manager module will interface the LinSM module when requesting a new schedule table (LIN master node only).

## 5.2 Relation to Lower layers

Below are the BSW modules that will be interfaced by LinSM module.

### 5.2.1 LinIf

The LinSM module assumes the following primitives to be provided by the LinIf [2] module:

- Transmission of the goto-sleep command (LIN master node only) and setting the lower layers to sleep mode (LinIf\_GotoSleep)
- The wakeup of the Lin bus (LinIf\_Wakeup)
- Request to change schedule tables (LinIf\_ScheduleRequest). Only applicable to LIN master node.

It is assumed that the LinIf module will call the following callbacks:

- Confirming that the operational mode has been entered, with or without transmission of a wakeup frame (LinSM\_WakeupConfirmation)

- Confirming that the sleep mode has been entered, after transmission of a goto-sleep command (LIN master node) or after reception of a goto-sleep command or bus idle detection (LIN slave node) (LinSM\_GotoSleepConfirmation)
- Confirming a schedule change (LinSM\_ScheduleRequestConfirmation). Only applicable to LIN master node.

**[SWS\_LinSM\_00002]** [The LinSM module shall not use or access the LIN driver or assume information about it any way other than what the LinIf module provides through the function calls to the LinIf module listed above.]

## 5.3 File structure

### 5.3.1 Code file structure

This chapter describes the c-files that implement the LinSM module Configuration. The code file structure is not defined completely within this specification. It is up to each implementer to design the missing structure details.

The pre compile and link time configuration parameters has to be kept in separate files:

### 5.3.2 Header File structure

This chapter describes the header files that will be included by the LinSM module and possible other modules.

#### 5.3.2.1 LinSM header files

Following header files will exist in a LinSM implementation:

**[SWS\_LinSM\_00005]** [A LinSM implementation shall provide a header file LinSM.h that contains all data exported from the LinSM - API declarations (except callbacks), extern types, and global data.]

#### 5.3.2.2 Included header files

Following external header files shall be included:

**[SWS\_LinSM\_00013]** [The LinSM module shall include the ComM.h file]



**[SWS\_LinSM\_00201]** [The LinSM module shall include the BswM\_LinSM.h]

**[SWS\_LinSM\_00305]** [The LinSM module shall include the ComM\_BusSM.h]

**[SWS\_LinSM\_00208]**

*Upstream requirements:* [SRS\\_BSW\\_00004](#)

[The LinSM module shall perform a consistency check between code files and header files based on pre-process-checking the version numbers of related code files and header files.]

## 6 Requirements Tracing

The following tables reference the requirements specified in [3] and [9] and links to the fulfillment of these. Please note that if column “Satisfied by” is empty for a specific requirement this means that this requirement is not fulfilled by this document.

| Requirement     | Description   | Satisfied by  |
|-----------------|---|---|
| [SRS_BSW_00004] | All Basic SW Modules shall perform a pre-processor check of the versions of all imported include files              | [SWS_LinSM_00208]   |
| [SRS_BSW_00101] | The Basic Software Module shall be able to initialize variables and hardware in a separate initialization function  | [SWS_LinSM_00155]   |
| [SRS_BSW_00167] | All AUTOSAR Basic Software Modules shall provide configuration rules and constraints to enable plausibility checks  | [SWS_LinSM_00073]   |
| [SRS_BSW_00358] | The return type of init() functions implemented by AUTOSAR Basic Software Modules shall be void                     | [SWS_LinSM_00155]   |
| [SRS_BSW_00369] | All AUTOSAR Basic Software Modules shall not return specific development error codes via the API                    | [SWS_LinSM_00113] [SWS_LinSM_00122]<br>[SWS_LinSM_00126]  |
| [SRS_BSW_00373] | The main processing function of each AUTOSAR Basic Software Module shall be named according the defined convention  | [SWS_LinSM_00156]   |
| [SRS_BSW_00384] | The Basic Software Module specifications shall specify at least in the description which other modules they require | [SWS_LinSM_00001] [SWS_LinSM_00085]<br>[SWS_LinSM_00086] [SWS_LinSM_00105]<br>[SWS_LinSM_00196]                   |
| [SRS_BSW_00406] | API handling in uninitialized state   | [SWS_LinSM_00116] [SWS_LinSM_00125]<br>[SWS_LinSM_00128] [SWS_LinSM_00131]<br>[SWS_LinSM_00134] [SWS_LinSM_00137] |
| [SRS_BSW_00407] | Each BSW module shall provide a function to read out the version information of a dedicated module implementation   | [SWS_LinSM_00117]   |
| [SRS_BSW_00414] | Init functions shall have a pointer to a configuration structure as single parameter                                | [SWS_LinSM_00155]   |

**Table 6.1: Requirements Tracing**

## 7 Functional specification

This chapter specifies the requirements on the module LinSM module. See the Basic Software Modules document [10] for an overview of the responsibilities of the LinSM.

The main responsibilities for the LinSM are:

- Control the communication status (no communication or full communication) of all LIN networks
- Handle schedule change requests (Only applicable to LIN master node)
- Handle communication mode requests
- Notify of state changes to upper layers

The LinSM module will not directly implement functionality in the LIN specification. The LinSM module will support the behavior defined in the ISO 17987 specification. The LIN behavior provided by the LinSM module will allow the reuse of existing LIN nodes conforming to the LIN 1.3, 2.0, 2.1, 2.2 and ISO 17987 specifications.

**[SWS\_LinSM\_00019]** [The LinSM module shall be able to handle one or more LIN networks.]

Number of LIN networks are restricted by the LinIf specification. All networks are handled via the NetworkHandleType specified by the ComM module.

The identification of the LIN networks is made using reference in the configuration to the ComM network handles.

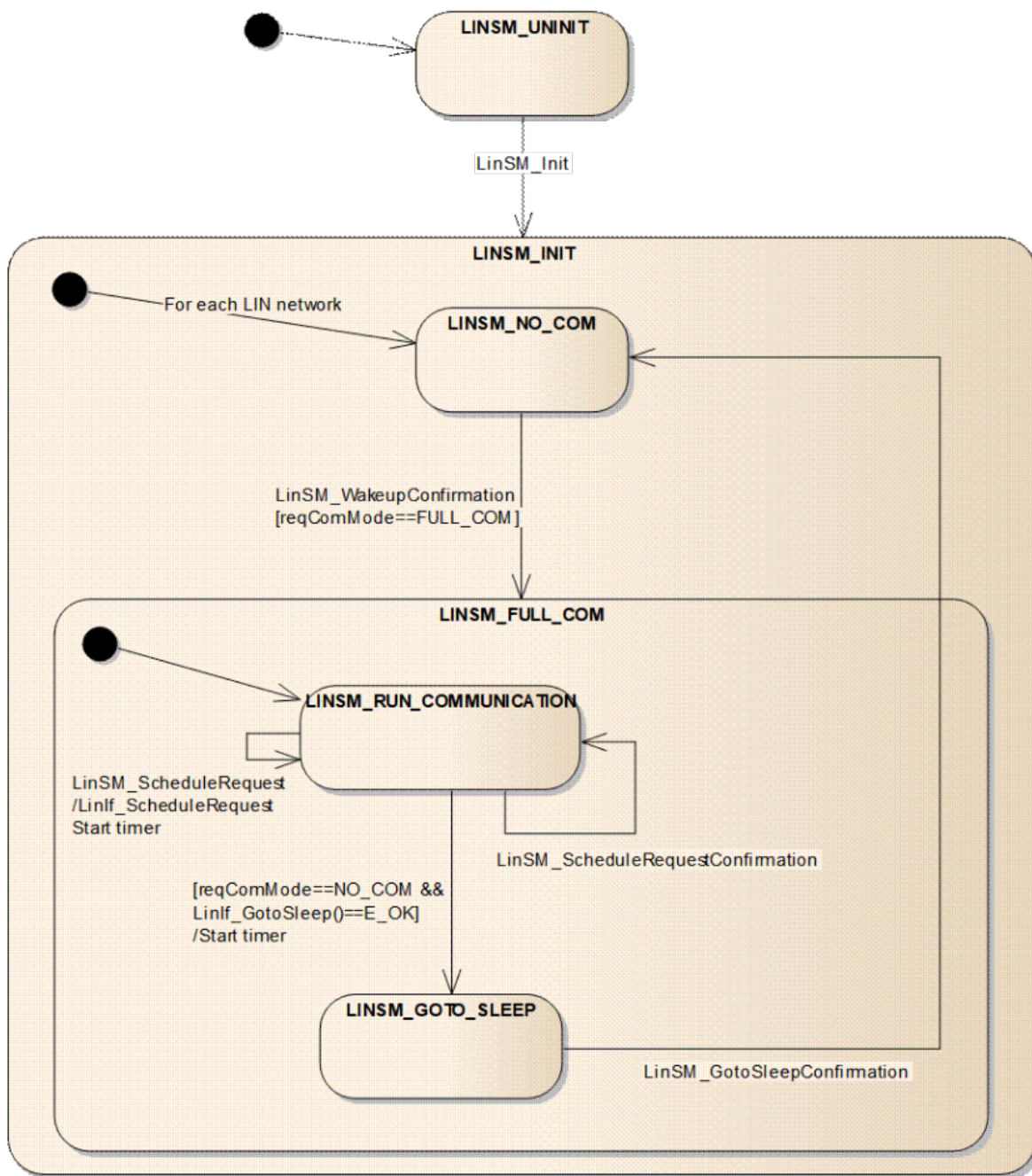
### 7.1 States and transitions of the LinSM state machine

The LinSM module will operate in a state-machine. Each network connected will operate in an independent sub-state-machine. Figure 7.1 and Figure 7.2 show a simplified version of the requirements below, the intention of the figures is not to be complete, rather give an overview.

**[SWS\_LinSM\_00020]** [The LinSM module shall have one state-machine containing the states LINSM\_UNINIT and LINSM\_INIT.]

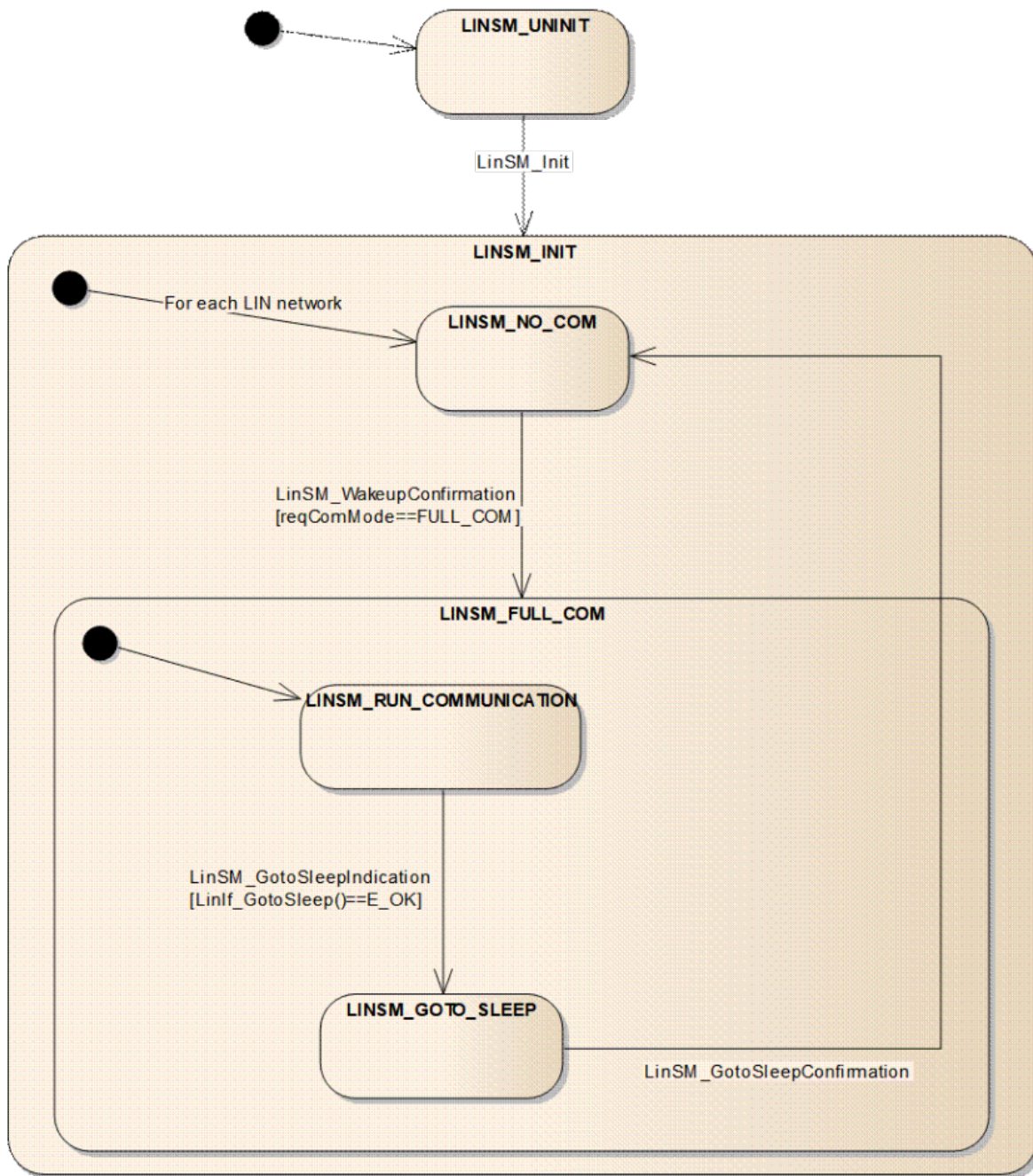
**[SWS\_LinSM\_00173]** [In the LINSM\_INIT there shall be a sub-state-machine for each network with the states LINSM\_NO\_COM and LINSM\_FULL\_COM.]

**[SWS\_LinSM\_00021]** [In LINSM\_INIT each network may be in the sub-states LINSM\_NO\_COM or LINSM\_FULL\_COM independently.]



**Figure 7.1: LinSM State Machine (master node)**





**Figure 7.2: LinSM State Machine (slave node)**

### 7.1.1 LINSM\_UNINIT

The uninit state is the first state that is active in the LinSM module.

**[SWS\_LinSM\_00022]** [There shall be a state called LINSM\_UNINIT]

**[SWS\_LinSM\_00161]** [The state LINSM\_UNINIT shall be active at start-up, before any API is called.]

### 7.1.2 LINSM\_INIT

After the initialization is made the LinSM module will activate the init state. There are two sub-states in this state. One is the no communication state where no communication is made on the LIN bus, the other is full communication where all communication is made and, for LIN master nodes, schedule tables are active. Each network may be in no communication or full communication independent of each other.

**[SWS\_LinSM\_00024]** [There shall be a state called LINSM\_INIT]

**[SWS\_LinSM\_00025]** [The LinSM state-machine shall transit from any state or sub-state to the state LINSM\_INIT when LinSM\_Init is called.]

**[SWS\_LinSM\_00152]** [The LinSM state-machine shall transit from any state or sub-state to sub-state LINSM\_NO\_COM for all networks when LinSM\_Init is called.]

**[SWS\_LinSM\_00043]** [When entering LINSM\_INIT the LinSM shall be put in an init state. Init state means that global variables, etc, shall be set to default value (reset value).]

**[SWS\_LinSM\_00160]** [The sub-state LINSM\_NO\_COM shall be active when entering the LINSM\_INIT state, for all networks when LinSM\_Init is called.]

**[SWS\_LinSM\_00216]** [The LinSM\_Init function shall set the schedule type NULL\_SCHEDULE for each configured channel. This requirement is only applicable for LIN master node.]

To make the LinSM independent from the LinIf module the LinSM module should not call the LinIf module in when the LinSM module is in the init function. The LinSM\_Init has therefore additional requirement, see [8.3.1](#)

### 7.1.3 LINSM\_NO\_COM

The no communication state is active after initialization and when the ComM module requests no communication (LIN master node) or when the LinIf indicates a bus sleep event (LIN slave node).

**[SWS\_LinSM\_00026]** [There shall be a sub-state called LINSM\_NO\_COM in the state LINSM\_INIT.]

**[SWS\_LinSM\_00027]** [When entering LINSM\_NO\_COM the LinSM module shall notify (with the exception [SWS\_LinSM\_00166]) ComM of the state change by calling the ComM\_BusSM\_ModelIndication with the parameter COMM\_NO\_COMMUNICATION for the specific network.]

**[SWS\_LinSM\_00193]** [When entering LINSM\_NO\_COM the LinSM module shall notify (with the exception [SWS\_LinSM\_00166]) BswM of the state change by calling the BswM\_LinSM\_CurrentState with the parameter LINSM\_NO\_COM for the specific network.]

There is one exception to the above two requirements. The rationale is that the ComM may not be initialized when executing the LinSM\_Init function.

**[SWS\_LinSM\_00166]** [The LinSM module shall not notify the state change to LINSM\_NO\_COM when the LinSM is executing the LinSM\_Init function, i.e. the LinSM\_Init function shall neither call ComM\_BusSM\_ModelIndication nor BswM\_LinSM\_CurrentState.]

**[SWS\_LinSM\_00028]** [When LINSM\_NO\_COM is active, the LinSM module shall not command the LinIf module to communicate for the selected network, i.e. bus shall be silent.]

Note: Upon entering or exiting the LINSM\_NO\_COM state the LinSM module will not set the hardware interface or  $\mu$ -controller into a new power mode. This is not in the scope of the LinSM]

**[SWS\_LinSM\_00203]** [When entering LINSM\_NO\_COM the transceiver shall be set to STANDBY if LinSMTransceiverPassiveMode is true and SLEEP otherwise by using the LinIf\_SetTrcvMode. This requirement is applicable only when LinSMTransceiverPassiveMode is configured for the channel.]

**[SWS\_LinSM\_00204]** [The LinIf\_SetTrcvMode shall not be called from the function LinSM\_Init.]

Note: There is no need to set the mode in the LinSM init function since the Transceiver will set the mode in its init function. The mode is selected in the Transceiver configuration.]

#### 7.1.4 LINSM\_FULL\_COM

The LINSM\_FULL\_COM is the only state where communication on the LIN bus is allowed. Each network can be in LINSM\_FULL\_COM independent of each other. All of the following requirements are applicable for each network.

**[SWS\_LinSM\_00032]** [There shall be a sub-state called LINSM\_FULL\_COM for each network in the state LINSM\_INIT.]

**[SWS\_LinSM\_00033]** [When entering LINSM\_FULL\_COM the ComM shall be notified of the state change by calling the ComM\_BusSM\_ModelIndication with the parameter COMM\_FULL\_COMMUNICATION for the specified network.]

**[SWS\_LinSM\_00192]** [When entering LINSM\_FULL\_COM the BswM shall be notified of the state change by calling the BswM\_LinSM\_CurrentState with the parameter LINSM\_FULL\_COM for the specified network.]

**[SWS\_LinSM\_00205]** [When entering LINSM\_FULL\_COM the transceiver shall be set to active by using the LinIf\_SetTrcvMode. This requirement is applicable only when LinSMTransceiverPassiveMode is configured for the channel.]

**[SWS\_LinSM\_00301]** [When entering LINSM\_FULL\_COM, the sub-state LINSM\_RUN\_COMMUNICATION will be entered.]

#### 7.1.5 Goto sleep

The goto-sleep sequence differs between master and slave nodes.

In a master node, when the ComM module requests the no communication mode, the LinSM will request the goto-sleep command to be sent on the LIN bus.

In a slave node, the LIN Interface indicates the bus sleep event to LinSM, either caused by reception of a goto-sleep command or by detection of a bus idle condition. If the ComM module has requested the no communication mode before, the bus sleep event is forwarded to ComM. Otherwise if the full communication mode requested by ComM module is active, the bus sleep event is not forwarded to ComM, but the wakeup process is restarted by LinSM after the goto-sleep sequence is completed.



In all cases, the entering of the no communication mode is notified to BswM and ComM. The callback will always be made, even if there was a problem.

**[SWS\_LinSM\_00035]** [The LinSM module may only call LinIf\_GotoSleep API in LinIf when the state LINSM\_FULL\_COM and the sub-state LINSM\_RUN\_COMMUNICATION is active.]

**[SWS\_LinSM\_00046]** [When LinSM\_GotoSleepConfirmation is called, and the current state/substate is LINSM\_FULL\_COM/LINSM\_GOTOSLEEP, the LinSM shall set the state to LINSM\_NO\_COM, regardless of the "success" parameter. In any other state, the LinSM\_GotoSleepConfirmation shall be ignored.]

**[SWS\_LinSM\_00302]** [If the LinIf\_GotoSleep returns E\_OK the LinSM sets the sub-state LINSM\_GOTOSLEEP.]

#### 7.1.5.1 Goto sleep specific for master node

This chapter is only applicable for LIN master nodes.

**[SWS\_LinSM\_10208]** [If the state is LINSM\_FULL\_COM, the ComM requests COMM\_NO\_COMMUNICATION; the LinSM shall call LinIf\_GotoSleep to transmit a goto sleep command on the requested network.]

**[SWS\_LinSM\_10209]** [In all other cases from [SWS\_LinSM\_10208] the LinIf\_GotoSleep shall not be called.]

**[SWS\_LinSM\_00036]** [If the ComM module calls LinSM\_RequestComMode requesting COMM\_NO\_COMMUNICATION the LinSM module shall directly call (and not wait for next main function call) the LinIf module function LinIf\_GotoSleep on the specified network.]

**[SWS\_LinSM\_00177]** [If the LinIf\_GotoSleep returns E\_NOT\_OK the LinSM\_RequestComMode shall return E\_NOT\_OK.

If the LinSM module returns LinSM\_RequestComMode with E\_NOT\_OK, the same state shall be set (so that a ComM\_BusSM\_ModeIndication and BswM\_LinSM\_CurrentState are called).]

### 7.1.5.2 Goto sleep specific for slave node

This chapter is only applicable for LIN slave nodes.

**[SWS\_LinSM\_00230]** [If the state is LINSM\_FULL\_COM, the ComM requests COMM\_NO\_COMMUNICATION; the LinSM shall store the requested communication mode and return E\_OK without further action.]

**[SWS\_LinSM\_00231]** [When LinSM\_GotoSleepIndication is called, and the current state is LINSM\_FULL\_COM, the LinSM shall directly call LinIf\_GotoSleep (and not wait for next main function call) to enter sleep mode on the requested network.]

**[SWS\_LinSM\_00232]** [In all other cases from SWS\_LinSM\_00231 the LinIf\_GotoSleep shall not be called.]

**[SWS\_LinSM\_00233]** [When the current state is LINSM\_FULL\_COM, and the requested communication mode by ComM module is COMM\_NO\_COMMUNICATION, LinIf shall call LinIf\_GotoSleep and afterwards notify ComM of the bus sleep event by calling ComM\_BusSM\_BusSleepMode for the specified network.]

**[SWS\_LinSM\_00234]** [In the case of [SWS\_LinSM\_00046] and the requested communication mode by ComM module is COMM\_FULL\_COMMUNICATION, the LinSM shall restart the wakeup up process.]

### 7.1.6 Changing schedule table (Master only)

This chapter is only applicable for LIN master nodes.

**[SWS\_LinSM\_00079]** [If the function LinSM\_ScheduleRequest is called, the LinSM module shall forward (and not wait for the next main function call) the request to the LinIf module using the function call LinIf\_ScheduleRequest.]

**[SWS\_LinSM\_00168]** [When the LinSM called LinIf\_ScheduleRequest from a call to LinSM\_ScheduleRequest, it shall forward the return value to its caller.]

**[SWS\_LinSM\_00213]** [If LinIf\_ScheduleRequest returns with E\_NOT\_OK the LinSM module shall call BswM\_LinSM\_CurrentSchedule with the old schedule table in the next main function call.]

**[SWS\_LinSM\_00206]** [When the LinSM module gets the confirmation of setting a schedule table from the LinIf module the BswM\_LinSM\_CurrentSchedule shall be called, if not timer has elapsed.]

**[SWS\_LinSM\_00214]** [If timer has elapsed, the LinSM module shall call BswM\_LinSM\_CurrentSchedule with unchanged schedule table.

Be aware of that the LinIf will switch to a NULL schedule when entering sleep, then it may make a schedule switch callback.]

**[SWS\_LinSM\_00207]** [If the LinIf confirms a schedule switch without a preceding call to request new schedule table the BswM\_LinSM\_CurrentSchedule shall be called]

### 7.1.7 Wake up process

A LIN network will be woken up if ComM module requests a wake up through the LinSM\_RequestComMode call or if a LIN node transmits the wakeup signal on the network. The wakeup by cluster is not handled by the LinSM module, it is handled by the EcuM module and will lead to that the ComM requests the network. In both cases the ComM will request full communication to the LinSM module for the specific network.

In case the LinIf is already awake (because of a LIN node waking up the bus) the LinIf will just ignore the wakeup call.

**[SWS\_LinSM\_00047]** [If the ComM requests COMM\_FULL\_COMMUNICATION the LinSM shall call LinIf\_Wakeup directly (and not wait for next main function call) to transmit a wake up signal on the requested network, except in the case of SWS\_LinSM\_00237. Additionally LinSM shall reset the counter for maximum number of retries.]

**[SWS\_LinSM\_00178]** [In all other cases from [SWS\_LinSM\_00047] the LinSM module shall not call LinIf\_Wakeup.]

**[SWS\_LinSM\_00049]** [When the LinIf notifies that the WakeUp is successfully sent (success = true), the state shall be set to LINSM\_FULL\_COM.]

**[SWS\_LinSM\_00202]** [In all other cases from [SWS\_LinSM\_00049] the state shall be set same state as previous to the request (so that a mode indication callback is made to BswM and ComM).]

**[SWS\_LinSM\_00176]** [If the LinIf\_Wakeup returns E\_NOT\_OK the LinSM\_RequestComMode shall return E\_NOT\_OK directly with no further action]

### 7.1.8 Timeout of requests

After calling `LinIf_Wakeup` the `LinSM` module is waiting for the `LinIf` module to confirm the wake up on the bus. There is a possibility that the confirmation is not received, and therefore the `LinSM` module will wait forever.

The cause for a missing wakeup confirmation could be a problem in software, but also a bus failure or a problem in the master node. A slave node confirms a bus wakeup not after wakeup transmission like a master node, but with reception of the first LIN header from the master node. A LIN slave node shall repeat the wakeup frame transmission up to three times if the communication does not start. After three (failing) wakeup requests the node shall wait a minimum time before restarting the wake up process.

**[SWS\_LinSM\_00175]** [There shall be independent wakeup confirmation timers for each network.]

**[SWS\_LinSM\_00162]** [The (countdown and expiration) handling of wakeup confirmation timers shall be done in the `LinSM_MainFunction`.]

**[SWS\_LinSM\_00159]** [The wakeup confirmation timeout parameter `LinSMConfirmationTimeout` shall have a time that is either 0 or divisible by the `LinSM_MainFunction` (i.e. `LinSM_MainFunction` period \* m; m integer >0).]

**[SWS\_LinSM\_00100]** [Before the `LinSM` calls `LinIf_Wakeup` on a channel it shall start a wakeup confirmation timer for the according channel with the configured timeout parameter `LinSMConfirmationTimeout` if `LinSMConfirmationTimeout` > 0.]

**[SWS\_LinSM\_00154]** [If `LinSM_WakeupConfirmation` is called on a channel before the wakeup confirmation timer has expired on the according channel `LinSM` shall stop the timer.]

**[SWS\_LinSM\_00102]** [If a wakeup confirmation timer elapses (i.e. no wakeup confirmation occurred) and the maximum number of retries (`LinSMModeRequestRepetitionMax`) has been reached and `LinSMDevErrorDetect` (`ECUC_LinSM_00206`) is set to TRUE, the `LinSM` module shall report `LINSM_E_CONFIRMATION_TIMEOUT` to DET.]

Making the timeout optional enhances implementation size, if the timeout is not required:

Note: If the configuration parameter `LinSMConfirmationTimeout` is set to zero the wakeup confirmation timer is not used, and hence a timeout cannot occur. This means that requirement **[SWS\_LinSM\_00102]** will not happen.

**[SWS\_LinSM\_00304]** [If a wakeup confirmation timer elapses (i.e. no wakeup confirmation occurred), LinSMNodeType is set to MASTER and ComM request is still COMM\_FULL\_COMMUNICATION then LinIf\_Wakeup shall be called.]

Note: LinSM stops to retry when confirmation callback LinSM\_WakeupConfirmation (SWS\_LinSM\_00132) occurs or timeout parameter LinSMConfirmationTimeout (ECUC\_LinSM\_00144) is expired and ComM requests COMM\_NO\_COMMUNICATION.

**[SWS\_LinSM\_00235]** [If a wakeup confirmation timer elapses (i.e. no wakeup confirmation occurred), the maximum number of retries (LinSMModeRequestRepetitionMax) has not been reached, LinSMNodeType is set to Slave and ComM request is still COMM\_FULL\_COMMUNICATION then LinIf\_Wakeup shall be called.]

**[SWS\_LinSM\_00236]** [If a wakeup confirmation timer elapses (i.e. no wakeup confirmation occurred), the maximum number of retries (LinSMModeRequestRepetitionMax) has been reached, LinSMNodeType is set to Slave then LinSM shall wait the silence-after-wakeup time (LinSMSilenceAfterWakeupTimeout) and afterwards restart the wakeup process by

- calling LinIf\_Wakeup if ComM request is still COMM\_FULL\_COMMUNICATION.
- resetting the counter for maximum number of retries.

]

**[SWS\_LinSM\_00237]** [If a wakeup confirmation timer elapses (i.e. no wakeup confirmation occurred), the maximum number of retries (LinSMModeRequestRepetitionMax) has been reached, LinSMNodeType is set to Slave, the silence-after-wakeup time (LinSMSilenceAfterWakeupTimeout) has not expired and ComM requests COMM\_FULL\_COMMUNICATION then the call of LinIf\_Wakeup shall be delayed to the expiration of the silence-after-wakeup time.]

## 7.2 Handling multiple networks and drivers

Usually only one LIN driver module (supporting multiple networks) is needed in an ECU to handle all LIN networks. However, rarely, some hardware configurations the ECU contain different LIN hardware (e.g. an advanced LIN controller and a UART). In such case, more than one different LIN drivers are required. This will not affect the LinSM module since the LIN driver only interfaces the LinIf module and not the LIN driver module directly.

The LinSM will only handle networks, and is not concerned to which driver the network maps to, this will be handled by the LinIf.

### 7.2.1 Multiple networks

Each network has a unique network index (LinSMComMNetworkHandleRef) in the LinSM configuration.

The configuration parameter LinSMComMNetworkHandleRef is referencing the ComM module configuration directly. This means that no mapping between networks has to be made in the LinSM module when interfacing to the LinIf module. The network index may be used directly to the LinIf module APIs.

**[SWS\_LinSM\_00164]** [The LinSM module shall use the same NetworkHandle value, received through an API, when interfacing to the LinIf module (when LIN network is required as a parameter).]

## 7.3 Error Classification

Section "Error Handling" of the document "General Specification of Basic Software Modules" describes the error handling of the Basic Software in detail. Above all, it constitutes a classification scheme consisting of five error types which may occur in BSW modules.

Based on this foundation, the following section specifies particular errors arranged in the respective subsections below.

### 7.3.1 Development Errors

**[SWS\_LinSM\_00053]** Definiton of development errors in module LinSM [

| Type of error  | Related error code          | Error value |
|--|-----------------------------|-------------|
| API called without initialization of LinSM                         | LINSM_E_UNINIT              | 0x00        |
| Referenced network does not exist (identification is out of range) | LINSM_E_NONEXISTENT_NETWORK | 0x20        |
| API service called with wrong parameter                            | LINSM_E_PARAMETER           | 0x30        |
| API service called with invalid pointer                            | LINSM_E_PARAM_POINTER       | 0x40        |
| Init function failed   | LINSM_E_INIT_FAILED         | 0x50        |

]

### 7.3.2 Runtime Errors

#### [SWS\_LinSM\_00224] Definition of runtime errors in module LinSM [

| <i>Type of error</i>                | <i>Related error code</i>    | <i>Error value</i> |
|-------------------------------------|------------------------------|--------------------|
| Timeout of the callbacks from LinIf | LINSM_E_CONFIRMATION_TIMEOUT | 0x00               |

]

### 7.3.3 Production Errors

There are no production errors.

### 7.3.4 Extended Production Errors

There are no extended production errors.

## 8 API specification

### 8.1 Imported types

#### 8.1.1 Standard types

In this chapter all types included from the following modules are listed. The standard AUTOSAR types are defined in the AUTOSAR Specification of Standard Types document [11].

Following types are used by the LinSM module:

#### [SWS\_LinSM\_00219] Definition of imported datatypes of module LinSM [

| Module  | Header File        | Imported Type        |
|---------|--------------------|----------------------|
| ComM    | Rte_ComM_Type.h    | ComM_ModeType        |
| Comtype | ComStack_Types.h   | NetworkHandleType    |
| LinIf   | LinIf.h            | LinIf_SchHandleType  |
| LinTrcv | Lin_GeneralTypes.h | LinTrcv_TrvcModeType |
| Std     | Std_Types.h        | Std_ReturnType       |
|         | Std_Types.h        | Std_VersionInfoType  |

]

### 8.2 Type definitions

Following types are defined by the LinSM module:

#### 8.2.1 LinSM\_ModeType

#### [SWS\_LinSM\_00220] Definition of datatype LinSM\_ModeType [

|                      |  |      |                    |
|----------------------|--|------|--------------------|
| <b>Name</b>          | LinSM_ModeType                                   |      |                    |
| <b>Kind</b>          | Type   |      |                    |
| <b>Derived from</b>  | uint8  |      |                    |
| <b>Range</b>         | LINSM_FULL_COM                                   | 0x01 | Full communication |
|                      | LINSM_NO_COM                                     | 0x02 | No communication   |
| <b>Description</b>   | Type used to report the current mode to the BswM |      |                    |
| <b>Available via</b> | LinSM.h  |      |                    |

]



## 8.2.2 LinSM\_ConfigType

### [SWS\_LinSM\_00221] Definition of datatype LinSM\_ConfigType [

|                      |  |   |
|----------------------|--|---|
| <b>Name</b>          | LinSM_ConfigType   |   |
| <b>Kind</b>          | Structure  |   |
| <b>Elements</b>      | implementation specific  |   |
|                      | <b>Type</b>  | – |
|                      | <b>Comment</b>   | – |
| <b>Description</b>   | Data structure type for the post-build configuration parameters. |   |
| <b>Available via</b> | LinSM.h  |   |

]

## 8.3 LinSM API

This is a list of API calls provided for upper layer modules.

### 8.3.1 LinSM\_Init

#### [SWS\_LinSM\_00155] Definition of API function LinSM\_Init

Upstream requirements: [SRS\\_BSW\\_00101](#), [SRS\\_BSW\\_00358](#), [SRS\\_BSW\\_00414](#)

[

|                           |  |   |
|---------------------------|--|---|
| <b>Service Name</b>       | LinSM_Init   |   |
| <b>Syntax</b>             | <pre>void LinSM_Init (     const LinSM_ConfigType* ConfigPtr )</pre> |   |
| <b>Service ID [hex]</b>   | 0x01   |   |
| <b>Sync/Async</b>         | Synchronous  |   |
| <b>Reentrancy</b>         | Non reentrant  |   |
| <b>Parameters (in)</b>    | ConfigPtr  | Pointer to the LinSM post-build configuration data. |
| <b>Parameters (inout)</b> | None   |   |
| <b>Parameters (out)</b>   | None   |   |
| <b>Return value</b>       | None   |   |
| <b>Description</b>        | This function initializes the LinSM.                                 |   |
| <b>Available via</b>      | LinSM.h  |   |

]

**[SWS\_LinSM\_00151]** [No other LinSM API or other module's (e.g. LinIf) API shall be called from the LinSM\_Init function. Other modules may not be initialized.]

### 8.3.2 LinSM\_ScheduleRequest

The service LinSM\_ScheduleRequest is only applicable for LIN master node.

#### [SWS\_LinSM\_00113] Definition of API function LinSM\_ScheduleRequest

Upstream requirements: [SRS\\_BSW\\_00369](#)

[

|                           |   |   |
|---------------------------|---|---|
| <b>Service Name</b>       | LinSM_ScheduleRequest   |   |
| <b>Syntax</b>             | <pre>Std_ReturnType LinSM_ScheduleRequest (     NetworkHandleType network,     LinIf_SchHandleType ScheduleTableIdx )</pre> |   |
| <b>Service ID [hex]</b>   | 0x10  |   |
| <b>Sync/Async</b>         | Asynchronous  |   |
| <b>Reentrancy</b>         | Reentrant   |   |
| <b>Parameters (in)</b>    | network   | Identification of the LIN channel   |
|                           | ScheduleTableIdx  | Index of the scheduled table  |
| <b>Parameters (inout)</b> | None  |   |
| <b>Parameters (out)</b>   | None  |   |
| <b>Return value</b>       | Std_ReturnType  | E_OK - Schedule table request has been accepted.<br>E_NOT_OK - Not possible to perform the request, e.g. not initialized. |
| <b>Description</b>        | The upper layer requests a schedule table to be changed on one LIN network.   |   |
| <b>Available via</b>      | LinSM.h   |   |

]

**[SWS\_LinSM\_00114]** [If LinSMDevErrorDetect is enabled: If the network parameter has an invalid value then the error-code LINSM\_E\_NONEXISTENT\_NETWORK shall be reported to the DET module and E\_NOT\_OK shall be returned.]

**[SWS\_LinSM\_00115]** [If LinSMDevErrorDetect is enabled: If the schedule parameter has an invalid value, then the error-code LINSM\_E\_PARAMETER shall be reported to the DET module and E\_NOT\_OK shall be returned.]

#### [SWS\_LinSM\_00116]

Upstream requirements: [SRS\\_BSW\\_00406](#)

[If LinSMDevErrorDetect is enabled: If the state LINSM\_UNINIT is active then the error-code LINSM\_E\_UNINIT shall be reported to the DET module and E\_NOT\_OK shall be returned.]

**[SWS\_LinSM\_10211]** [If the function LinSM\_ScheduleRequest is called and the state is not LINSM\_FULL\_COM, the LinSM\_ScheduleRequest shall return directly with E\_NOT\_OK.]

**[SWS\_LinSM\_00163]** [If the function `LinSM_ScheduleRequest` is called, `LinSMOverwritePendingScheduleRequestProcessing` is set to `FALSE` and another schedule request on the same network is pending, the `LinSM_ScheduleRequest` shall return directly with `E_NOT_OK`.]

**[SWS\_LinSM\_00165]** [If the function `LinSM_ScheduleRequest` is called, `LinSMOverwritePendingScheduleRequestProcessing` is set to `TRUE` and another schedule request on the same network is pending, the pending schedule request shall be overwritten with the new schedule request and return with `E_OK`.]

**[SWS\_LinSM\_00241]** [The function `LinSM_ScheduleRequest` shall be available if the `LinSM` module is configured as LIN master node on at least one channel.]

Note: In a pure LIN slave configuration, this function is not available. This depends on the configuration parameters `LinSMNodeType`.

### 8.3.3 LinSM\_GetVersionInfo

#### **[SWS\_LinSM\_00117]** Definition of API function `LinSM_GetVersionInfo`

Upstream requirements: [SRS\\_BSW\\_00407](#)

[

|                           |   |   |
|---------------------------|---|---|
| <b>Service Name</b>       | LinSM_GetVersionInfo  |   |
| <b>Syntax</b>             | <pre>void LinSM_GetVersionInfo (     Std_VersionInfoType* versioninfo )</pre> |   |
| <b>Service ID [hex]</b>   | 0x02  |   |
| <b>Sync/Async</b>         | Synchronous   |   |
| <b>Reentrancy</b>         | Reentrant   |   |
| <b>Parameters (in)</b>    | None  |   |
| <b>Parameters (inout)</b> | None  |   |
| <b>Parameters (out)</b>   | versioninfo   | Pointer to where to store the version information of this module. |
| <b>Return value</b>       | None  |   |
| <b>Description</b>        | –   |   |
| <b>Available via</b>      | LinSM.h   |   |

]

**[SWS\_LinSM\_00119]** [If `LinSMDevErrorDetect` is enabled: If the `versioninfo` pointer parameter is invalid (e.g. `NULL`), the error-code `LINSM_E_PARAM_POINTER` shall be reported to the DET module and `E_NOT_OK` shall be returned.]

### 8.3.4 LinSM\_GetCurrentComMode

#### [SWS\_LinSM\_00122] Definition of API function LinSM\_GetCurrentComMode

Upstream requirements: [SRS\\_BSW\\_00369](#)

|                           |  |  |
|---------------------------|--|--|
| <b>Service Name</b>       | LinSM_GetCurrentComMode  |  |
| <b>Syntax</b>             | Std_ReturnType LinSM_GetCurrentComMode (<br>NetworkHandleType network,<br>ComM_ModeType* mode<br>) |  |
| <b>Service ID [hex]</b>   | 0x11   |  |
| <b>Sync/Async</b>         | Synchronous  |  |
| <b>Reentrancy</b>         | Reentrant  |  |
| <b>Parameters (in)</b>    | network  | Identification of the LIN channel  |
| <b>Parameters (inout)</b> | None   |  |
| <b>Parameters (out)</b>   | mode   | Returns the active mode, see ComM_ModeType for descriptions of the modes           |
| <b>Return value</b>       | Std_ReturnType   | E_OK - Ok<br>E_NOT_OK - Not possible to perform the request, e.g. not initialized. |
| <b>Description</b>        | Function to query the current communication mode.  |  |
| <b>Available via</b>      | LinSM.h  |  |

**[SWS\_LinSM\_00123]** [If LinSMDevErrorDetect is enabled: If the network parameter has an invalid value, then the error-code LINSM\_E\_NONEXISTENT\_NETWORK shall be reported to the DET module and E\_NOT\_OK shall be returned.]

**[SWS\_LinSM\_00124]** [If LinSMDevErrorDetect is enabled: If the mode pointer parameter is invalid (e.g. NULL), then the error-code LINSM\_E\_PARAM\_POINTER shall be reported to the DET module and E\_NOT\_OK shall be returned.]

#### [SWS\_LinSM\_00125]

Upstream requirements: [SRS\\_BSW\\_00406](#)

[If LinSMDevErrorDetect is enabled: If the state LINSM\_UNINIT is active, then the error-code LINSM\_E\_UNINIT shall be reported to the DET module and E\_NOT\_OK shall be returned]

**[SWS\_LinSM\_00180]** [If active state is LINSM\_NO\_COM the state COMM\_NO\_COMMUNICATION shall be returned.]

**[SWS\_LinSM\_00181]** [If active state is LINSM\_FULL\_COM the state COMM\_FULL\_COMMUNICATION shall be returned.]

**[SWS\_LinSM\_00182]** [If active state is LINSM\_UNINIT the state COMM\_NO\_COMMUNICATION shall be returned. This is also captured above when the DET is enabled. This is to be defensive.]

Note that COMM\_SILENT\_COMMUNICATION is not used by the LinSM module.

### 8.3.5 LinSM\_RequestComMode

#### [SWS\_LinSM\_00126] Definition of API function LinSM\_RequestComMode

Upstream requirements: [SRS\\_BSW\\_00369](#)

[

|                           |   |  |
|---------------------------|---|--|
| <b>Service Name</b>       | LinSM_RequestComMode  |  |
| <b>Syntax</b>             | Std_ReturnType LinSM_RequestComMode (<br>NetworkHandleType network,<br>ComM_ModeType mode<br>)  |  |
| <b>Service ID [hex]</b>   | 0x12  |  |
| <b>Sync/Async</b>         | Asynchronous  |  |
| <b>Reentrancy</b>         | Reentrant   |  |
| <b>Parameters (in)</b>    | network   | Identification of the LIN channel  |
|                           | mode  | Request mode   |
| <b>Parameters (inout)</b> | None  |  |
| <b>Parameters (out)</b>   | None  |  |
| <b>Return value</b>       | Std_ReturnType  | E_OK - Request accepted<br>E_NOT_OK - Not possible to perform the request, e.g. not initialized. |
|                           |   |  |
| <b>Description</b>        | Requesting of a communication mode.<br><br>The mode switch will not be made instant. The LinSM will notify the caller when mode transition is made. |  |
| <b>Available via</b>      | LinSM.h   |  |

]

**[SWS\_LinSM\_00127]** [If LinSMDevErrorDetect is enabled: If the network parameter has an invalid value, then the error-code LINSM\_E\_NONEXISTENT\_NETWORK shall be reported to the DET module and E\_NOT\_OK shall be returned.]

**[SWS\_LinSM\_00191]** [If LinSMDevErrorDetect is enabled: If the mode parameter has an invalid value, then the error-code LINSM\_E\_PARAMETER shall be reported to the DET module and E\_NOT\_OK shall be returned.]

#### [SWS\_LinSM\_00128]

*Upstream requirements:* [SRS\\_BSW\\_00406](#)

[If LinSMDevErrorDetect is enabled: If the state LINSM\_UNINIT is active, then the error-code LINSM\_E\_UNINIT shall be reported to the DET module and E\_NOT\_OK shall be returned.]

**[SWS\_LinSM\_00183]** [If COMM\_SILENT\_COMMUNICATION is requested the function shall return E\_NOT\_OK directly without action]

**[SWS\_LinSM\_00223]** [LinSM\_RequestComMode shall store the requested mode, if the return value is E\_OK.

The next activation of the LinSM\_MainFunction will then process this request when processing the state machine.

Note, that the state machine definition in section refers to this stored request as req ComMode.]

## 8.4 Scheduled functions

This chapter lists the functions that are called with a fixed period.

### 8.4.1 LinSM\_MainFunction

This function is directly called by the Basic Software Scheduler module. The following function has no return value, no parameter and is non-reentrant.

There is no dependency to other main functions. This main function may be executed without considering other main functions. But scheduling the different main functions intelligent will minimize execution time and jitter.

#### [SWS\_LinSM\_00156] Definition of scheduled function LinSM\_MainFunction

*Upstream requirements:* [SRS\\_BSW\\_00373](#)

[

|                         |  |
|-------------------------|--|
| <b>Service Name</b>     | LinSM_MainFunction   |
| <b>Syntax</b>           | void LinSM_MainFunction (<br>void<br>)                               |
| <b>Service ID [hex]</b> | 0x30   |
| <b>Description</b>      | Periodic function that runs the timers of different request timeouts |





|                      |              |
|----------------------|--------------|
| <b>Available via</b> | SchM_LinSm.h |
|----------------------|--------------|

└

Design hint: The function LinSM\_MainFunction may be interrupted by other functions. It should be assured that the timers operated by this function are protected so that they behave correctly (e.g. by using critical sections if necessary).

**[SWS\_LinSM\_00157]** [The LinSM\_MainFunction shall handle the timers that are attached to the functions LinIf\_GotoSleep, LinIf\_Wakeup or LinIf\_ScheduleRequest (see paragraph)]

## 8.5 LinSM callbacks

### 8.5.1 LinSM\_ScheduleRequestConfirmation

The callback LinSM\_ScheduleRequestConfirmation is only applicable for LIN master node.

**[SWS\_LinSM\_00129] Definition of callback function LinSM\_ScheduleRequest Confirmation** [

|                           |   |                                   |
|---------------------------|---|-----------------------------------|
| <b>Service Name</b>       | LinSM_ScheduleRequestConfirmation   |                                   |
| <b>Syntax</b>             | <pre>void LinSM_ScheduleRequestConfirmation (     NetworkHandleType network,     LinIf_SchHandleType ScheduleTableIdx )</pre> |                                   |
| <b>Service ID [hex]</b>   | 0x20  |                                   |
| <b>Sync/Async</b>         | Synchronous   |                                   |
| <b>Reentrancy</b>         | Reentrant   |                                   |
| <b>Parameters (in)</b>    | network   | Identification of the LIN channel |
|                           | ScheduleTableIdx  | Index of the scheduled table      |
| <b>Parameters (inout)</b> | None  |                                   |
| <b>Parameters (out)</b>   | None  |                                   |
| <b>Return value</b>       | None  |                                   |
| <b>Description</b>        | The LinIf module will call this callback when the new requested schedule table is active.                                     |                                   |
| <b>Available via</b>      | LinSM.h   |                                   |

└

**[SWS\_LinSM\_00130]** [If LinSMDevErrorDetect is enabled: If the network parameter has an invalid value, then the error-code LINSM\_E\_NONEXISTENT\_NETWORK shall be reported to the DET module.]

### [SWS\_LinSM\_00131]

Upstream requirements: [SRS\\_BSW\\_00406](#)

[If LinSMDevErrorDetect is enabled: If the state LINSM\_UNINIT is active, then the error-code LINSM\_E\_UNINIT shall be reported to DET module.]

**[SWS\_LinSM\_00242]** [The callback function LinSM\_ScheduleRequestConfirmation shall be available if the LinSM module is configured as LIN master node on at least one channel.]

Note: In a pure LIN slave configuration, this function is not available. This depends on the configuration parameters LinSMNodeType.

## 8.5.2 LinSM\_GotoSleepIndication

The callback LinSM\_GotoSleepIndication is only applicable for LIN slave node.

### [SWS\_LinSM\_91000] Definition of callback function LinSM\_GotoSleepIndication

[

|                           |  |                                   |
|---------------------------|--|-----------------------------------|
| <b>Service Name</b>       | LinSM_GotoSleepIndication  |                                   |
| <b>Syntax</b>             | <pre>void LinSM_GotoSleepIndication (     NetworkHandleType Channel )</pre>  |                                   |
| <b>Service ID [hex]</b>   | 0x03   |                                   |
| <b>Sync/Async</b>         | Synchronous  |                                   |
| <b>Reentrancy</b>         | Reentrant for different Channels   |                                   |
| <b>Parameters (in)</b>    | Channel  | Identification of the LIN channel |
| <b>Parameters (inout)</b> | None   |                                   |
| <b>Parameters (out)</b>   | None   |                                   |
| <b>Return value</b>       | None   |                                   |
| <b>Description</b>        | <p>The LinIf will call this callback when the go to sleep command is received on the network or a bus idle timeout occurs.</p> <p>Only applicable for LIN slave nodes.</p> |                                   |
| <b>Available via</b>      | LinSM.h  |                                   |

]

**[SWS\_LinSM\_00239]** [If LinSMDevErrorDetect is enabled: If the network parameter has an invalid value, then the error-code LINSM\_E\_NONEXISTENT\_NETWORK shall be reported to the DET module.]

**[SWS\_LinSM\_00240]** [If LinSMDevErrorDetect is enabled: If the state LINSM\_UNINIT is active, then the error-code LINSM\_E\_UNINIT shall be reported to DET module.]



**[SWS\_LinSM\_00243]** [The callback function `LinSM_GotoSleepIndication` shall be available if the LinSM module is configured as LIN slave node on at least one channel.]

Note: In a pure LIN master configuration, this function is not available. This depends on the configuration parameters `LinSMNodeType`.

### 8.5.3 LinSM\_GotoSleepConfirmation

**[SWS\_LinSM\_00135]** Definition of callback function `LinSM_GotoSleepConfirmation` [

|                           |  |   |
|---------------------------|--|---|
| <b>Service Name</b>       | LinSM_GotoSleepConfirmation  |   |
| <b>Syntax</b>             | <pre>void LinSM_GotoSleepConfirmation (     NetworkHandleType network,     boolean success )</pre>                           |   |
| <b>Service ID [hex]</b>   | 0x22   |   |
| <b>Sync/Async</b>         | Synchronous  |   |
| <b>Reentrancy</b>         | Reentrant  |   |
| <b>Parameters (in)</b>    | network  | Identification of the LIN channel                         |
|                           | success  | True if goto sleep was successfully sent, false otherwise |
| <b>Parameters (inout)</b> | None   |   |
| <b>Parameters (out)</b>   | None   |   |
| <b>Return value</b>       | None   |   |
| <b>Description</b>        | The LinIf will call this callback when the go to sleep command is sent successfully or not sent successfully on the network. |   |
| <b>Available via</b>      | LinSM.h  |   |

]

**[SWS\_LinSM\_00136]** [If `LinSMDevErrorDetect` is enabled: If the network parameter has an invalid value, then the error-code `LINSM_E_NONEXISTENT_NETWORK` shall be reported to the DET module.]

**[SWS\_LinSM\_00137]**

*Upstream requirements:* [SRS\\_BSW\\_00406](#)

[If `LinSMDevErrorDetect` is enabled: If the state `LINSM_UNINIT` is active, then the error-code `LINSM_E_UNINIT` shall be reported to the DET module.]

## 8.5.4 LinSM\_WakeupConfirmation

### [SWS\_LinSM\_00132] Definition of callback function LinSM\_WakeupConfirmation

[

|                           |  |   |
|---------------------------|--|---|
| <b>Service Name</b>       | LinSM_WakeupConfirmation   |   |
| <b>Syntax</b>             | <pre>void LinSM_WakeupConfirmation (     NetworkHandleType network,     boolean success )</pre>                          |   |
| <b>Service ID [hex]</b>   | 0x21   |   |
| <b>Sync/Async</b>         | Synchronous  |   |
| <b>Reentrancy</b>         | Reentrant  |   |
| <b>Parameters (in)</b>    | network  | Identification of the LIN channel                     |
|                           | success  | True if wakeup was successfully sent, false otherwise |
| <b>Parameters (inout)</b> | None   |   |
| <b>Parameters (out)</b>   | None   |   |
| <b>Return value</b>       | None   |   |
| <b>Description</b>        | The LinIf will call this callback when the wake up signal command is sent not successfully/ successfully on the network. |   |
| <b>Available via</b>      | LinSM.h  |   |

]

**[SWS\_LinSM\_00133]** [If LinSMDevErrorDetect is enabled: If the network parameter has an invalid value, then the error-code LINSM\_E\_NONEXISTENT\_NETWORK shall be reported to the DET module.]

### [SWS\_LinSM\_00134]

*Upstream requirements:* [SRS\\_BSW\\_00406](#)

[If LinSMDevErrorDetect is enabled: If the state LINSM\_UNINIT is active, then the error-code LINSM\_E\_UNINIT shall be reported to the DET module.]

## 8.6 Mandatory Interfaces

This chapter defines all interfaces that are required to fulfill the core functionality.

## [SWS\_LinSM\_00229] Definition of mandatory interfaces required by module LinSM

| API Function               | Header File  | Description  |
|----------------------------|--------------|--|
| BswM_LinSM_CurrentSchedule | BswM_LinSM.h | Function called by LinSM to indicate the currently active schedule table for a specific LIN channel.   |
| BswM_LinSM_CurrentState    | BswM_LinSM.h | Function called by LinSM to indicate its current state.  |
| ComM_BusSM_BusSleepMode    | ComM.h       | Notification of the corresponding Bus State Manager that the actual bus mode is Bus-Sleep.<br><br>Only applicable for ComM channels with ComMNM Variant set to SLAVE_ACTIVE or SLAVE_PASSIVE.<br><br>E.g. LIN slaves (ComMNMVariant = SLAVE_ACTIVE) or Ethernet channels with OA TC10 compliant Ethernet hardware which act as passive communication slave (ComMNMVariant = SLAVE_PASSIVE and EthTrcvActAsSlavePassiveEnabled set to TRUE) |
| ComM_BusSM_ModeIndication  | ComM.h       | Indication of the actual bus mode by the corresponding Bus State Manager. ComM shall propagate the indicated state to the users with means of the RTE and BswM.  |
| Det_ReportRuntimeError     | Det.h        | Service to report runtime errors. If a callout has been configured then this callout shall be called.  |
| LinIf_GotoSleep            | LinIf.h      | Initiates a transition into the Sleep Mode on the selected channel.  |
| LinIf_ScheduleRequest      | LinIf.h      | Requests a schedule table to be executed.<br><br>Only used for LIN master nodes.   |
| LinIf_Wakeup               | LinIf.h      | Initiates the wake up process.   |

## 8.7 Optional Interfaces

This chapter defines all interfaces, which are required to fulfill an optional functionality of the module.

## [SWS\_LinSM\_00138] Definition of optional interfaces requested by module LinSM

| API Function      | Header File | Description                                      |
|-------------------|-------------|--|
| Det_ReportError   | Det.h       | Service to report development errors.            |
| LinIf_SetTrcvMode | LinIf.h     | Set the given LIN transceiver to the given mode. |

## 8.8 Configurable Interfaces

No configurable interfaces.

## 9 Sequence diagrams

This chapter will show use-cases for LIN communication and API usage. As the communication is in real-time it is not easy to show the real-time behavior in the UML dynamic diagrams. It is advisable to read the corresponding descriptive text to each UML diagram.

To show the behavior of the modules in the different use-cases, there are local function calls made to show what is done and when to get information. It is not mandatory to use these local functions; they are here just to make the use-cases more understandable.

Note that all parameters and return types are left out to make the diagrams easier to read and understand. If needed for clarification the parameter value or return value are shown.

### 9.1 Goto-sleep process

#### 9.1.1 Master

This chapter is only applicable for LIN master nodes.

This use-case shows the transition into the LINSM\_NO\_COM state when the goto-sleep command has been sent successfully on the bus.

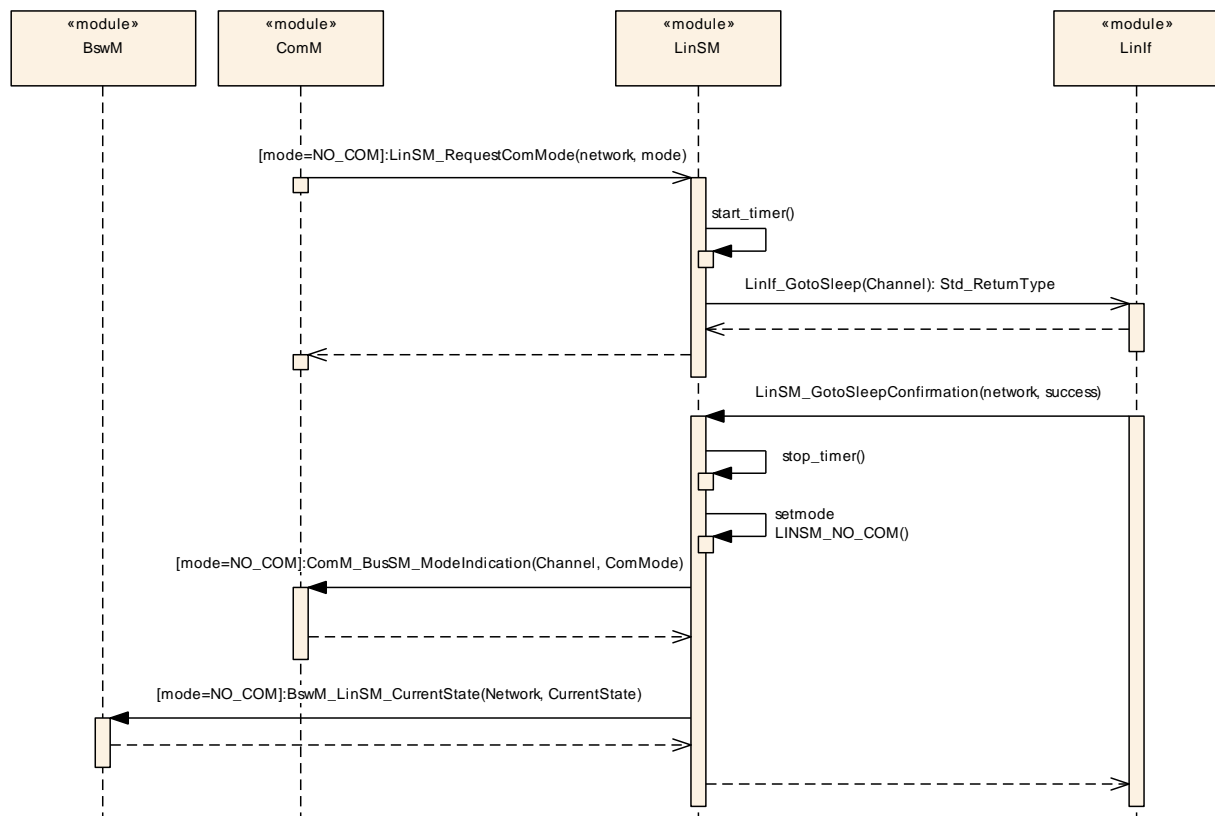
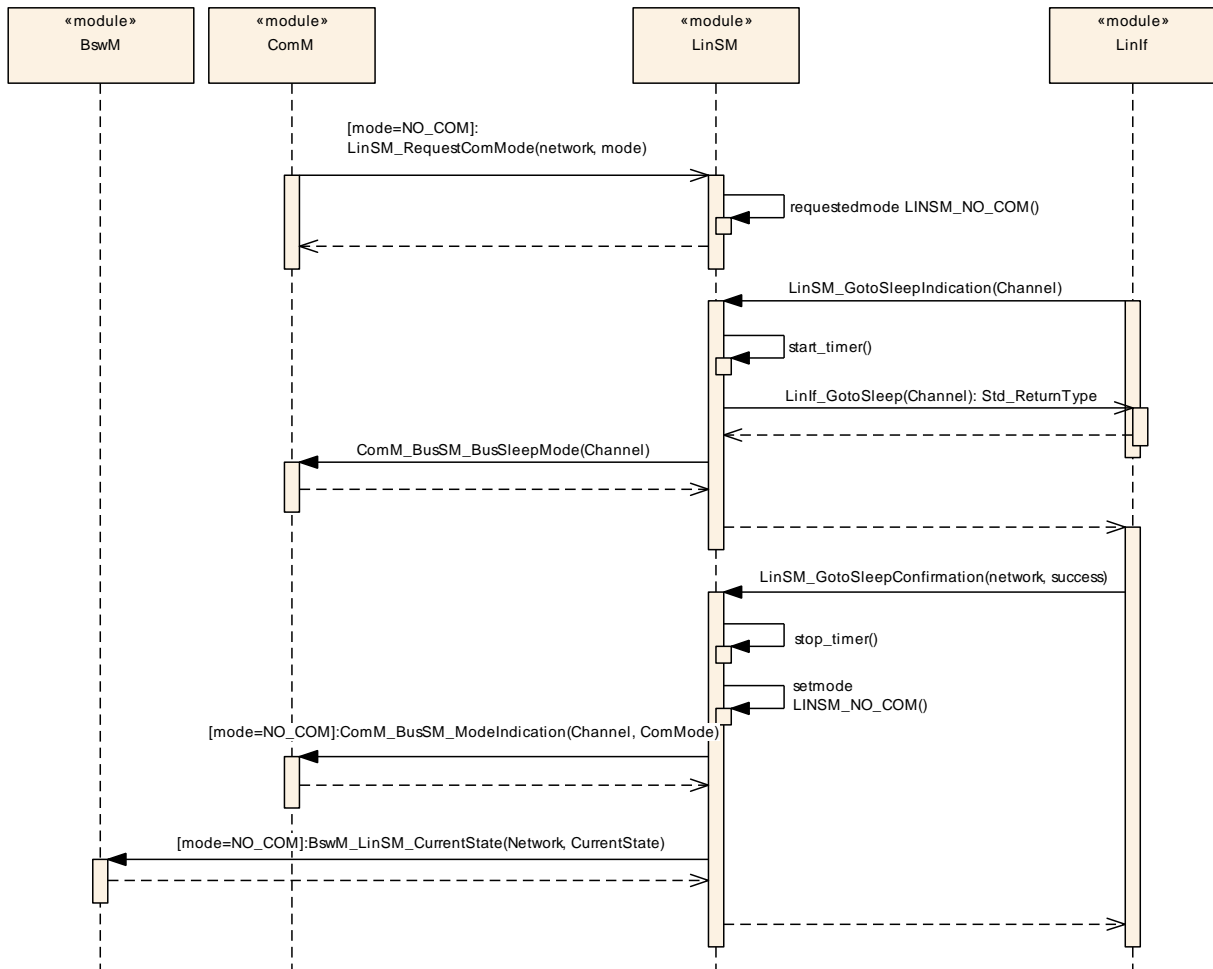


Figure 9.1: Goto-sleep command process (Master)

## 9.1.2 Slave

This chapter is only applicable for LIN slave nodes.

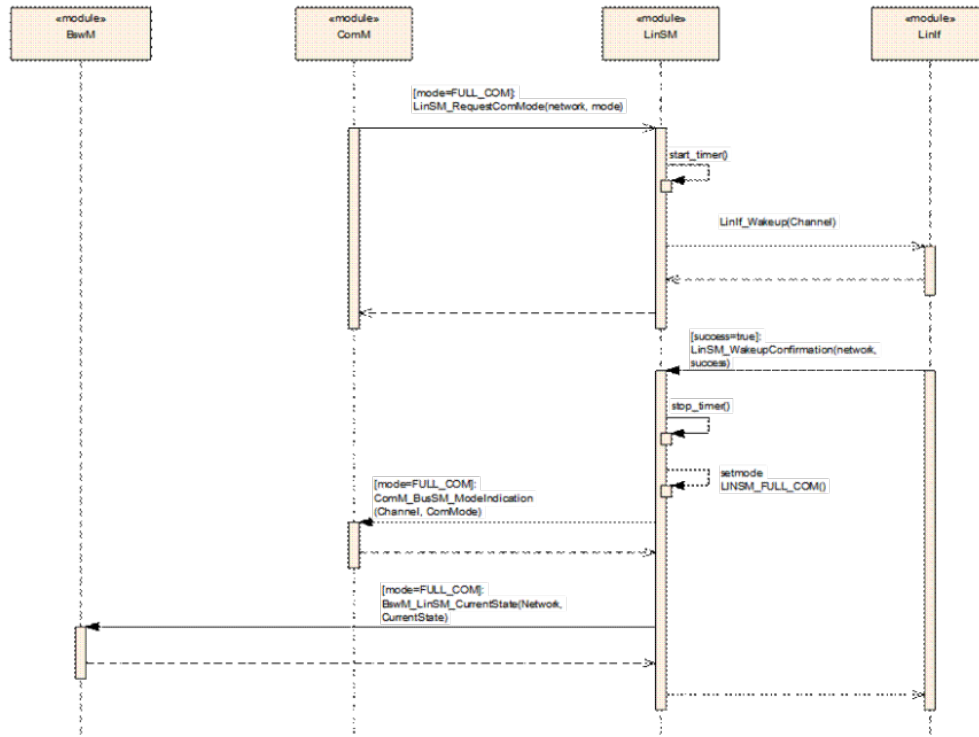
This use-case shows the transition into the LINSM\_NO\_COM state when the goto-sleep command has been received on the bus when no communication is requested.



**Figure 9.2: - Goto-sleep-command process (Slave)**

## 9.2 Internal wake up

The Figure 9.3 shows the internal wakeup. A wakeup is requested from module above the LinSM.

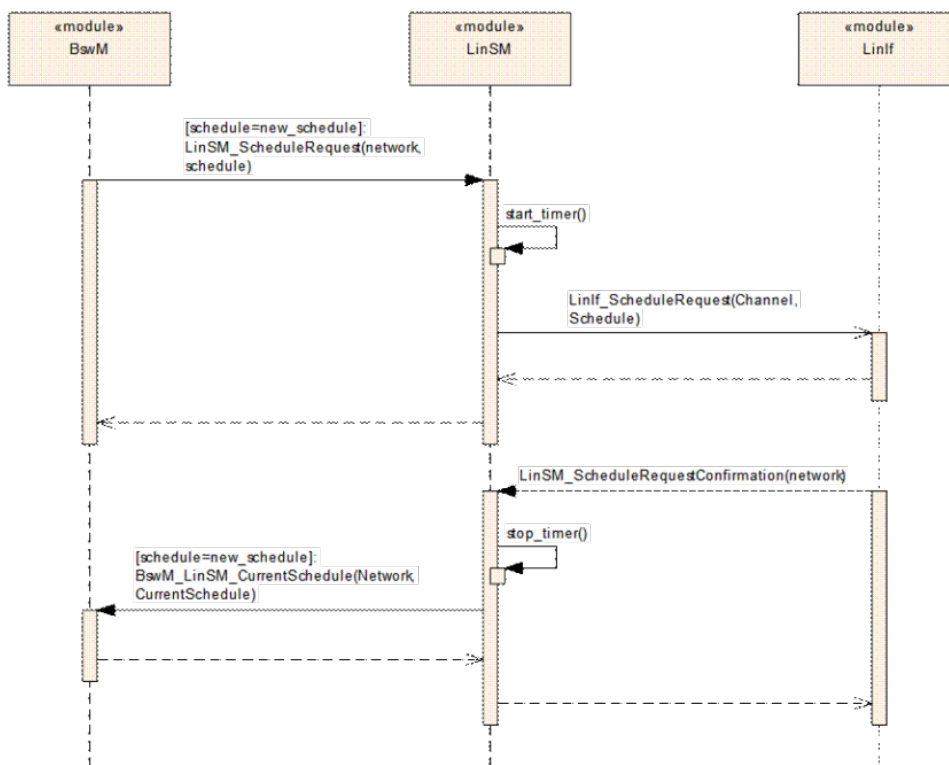


**Figure 9.3: - Internal wake up**

### 9.3 Schedule switch (Master only)

This chapter is only applicable for LIN master nodes.

Figure 9.4 shows the use-cases of switching the schedule table when the LinSM accepts the request.



**Figure 9.4: Schedule Table switch**



## 10 Configuration specification

In general, this chapter defines configuration parameters and their clustering into containers.

The chapter [10.3](#) specifies the structure (containers) and the parameters of the LinSM module.

### 10.1 How to read this chapter

For details refer to the chapter 10.1 "Introduction to configuration specification" in SWS\_BSWGeneral.

### 10.2 Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters describe [Chapter 7](#) and [Chapter 8](#).

Example: The LinSM\_Configuration is placed in a specific Flash sector. This flash sector may be reflashed after the ECU is placed in the vehicle.

#### 10.2.1 Configuration Tool

A configuration tool will create a configuration structure that is understood by the LinSM.

#### [SWS\_LinSM\_00073]

*Upstream requirements:* [SRS\\_BSW\\_00167](#)

[The LinSM module shall not make any consistency check of the configuration in run-time in production software. It may, however, be done if the Development Error Detection is enabled.]

### 10.3 LinSM\_Configuration

The paragraph defines the LinSM configuration.

### 10.3.1 LinSM

#### [ECUC\_LinSM\_00209] Definition of EcucModuleDef LinSM [

|                                   |  |
|-----------------------------------|--|
| <b>Module Name</b>                | LinSM  |
| <b>Description</b>                | Configuration of the Lin State Manager module.             |
| <b>Post-Build Variant Support</b> | true   |
| <b>Supported Config Variants</b>  | VARIANT-LINK-TIME, VARIANT-POST-BUILD, VARIANT-PRE-COMPILE |

| Included Containers            |              |  |
|--------------------------------|--------------|--|
| Container Name                 | Multiplicity | Scope / Dependency   |
| <a href="#">LinSMConfigSet</a> | 1            | This container contains the configuration parameters and sub containers of the AUTOSAR LinSm module. |
| <a href="#">LinSMGeneral</a>   | 1            | This container contains general parameters of LIN State Manager module.                              |

### 10.3.2 LinSMConfigSet

#### [ECUC\_LinSM\_00207] Definition of EcucParamConfContainerDef LinSMConfigSet [

|                                 |  |
|---------------------------------|--|
| <b>Container Name</b>           | LinSMConfigSet   |
| <b>Parent Container</b>         | <a href="#">LinSM</a>  |
| <b>Description</b>              | This container contains the configuration parameters and sub containers of the AUTOSAR LinSm module. |
| <b>Configuration Parameters</b> |  |

| Included Parameters                                  |              |                    |
|--|--------------|--------------------|
| Parameter Name                                       | Multiplicity | ECUC ID            |
| <a href="#">LinSMModeRequestRepetitionMax</a>        | 1            | [ECUC_LinSM_00208] |
| <a href="#">LinSMOverwritePendingScheduleRequest</a> | 1            | [ECUC_LinSM_00212] |

| Included Containers          |              |   |
|------------------------------|--------------|---|
| Container Name               | Multiplicity | Scope / Dependency                                    |
| <a href="#">LinSMChannel</a> | 1..*         | Describes each LIN channel the LinSM is connected to. |

## [ECUC\_LinSM\_00208] Definition of EcucIntegerParamDef LinSMModeRequestRepetitionMax

|                           |   |   |                     |
|---------------------------|---|---|---------------------|
| Parameter Name            | LinSMModeRequestRepetitionMax   |   |                     |
| Parent Container          | <a href="#">LinSMConfigSet</a>  |   |                     |
| Description               | Specifies the maximum amount of wakeup repetitions without a respective wakeup confirmation from the LinIf module until the LinSM module reports a runtime Error to DET. In case the channel is configured as slave it also specifies after how many retries the silence-after-wakeup is started. |   |                     |
| Multiplicity              | 1   |   |                     |
| Type                      | EcucIntegerParamDef   |   |                     |
| Range                     | 0 .. 255  |   |                     |
| Default value             | –   |   |                     |
| Post-Build Variant Value  | true  |   |                     |
| Value Configuration Class | Pre-compile time  | X | VARIANT-PRE-COMPILE |
|                           | Link time   | X | VARIANT-LINK-TIME   |
|                           | Post-build time   | X | VARIANT-POST-BUILD  |
| Scope / Dependency        | scope: local  |   |                     |

## [ECUC\_LinSM\_00212] Definition of EcucBooleanParamDef LinSMOverwritePendingScheduleRequest

|                                  |  |   |                                       |
|----------------------------------|--|---|---------------------------------------|
| Parameter Name                   | LinSMOverwritePendingScheduleRequest   |   |                                       |
| Parent Container                 | <a href="#">LinSMConfigSet</a>   |   |                                       |
| Description                      | Select the handling of a pending schedule requests processing, if a new schedule request is requested. <ul style="list-style-type: none"> <li>• FALSE: Reject a new schedule request, if a schedule request processing is pending.</li> <li>• TRUE: Overwrite a pending request processing with the new schedule request.</li> </ul> |   |                                       |
| Multiplicity                     | 1  |   |                                       |
| Type                             | EcucBooleanParamDef  |   |                                       |
| Default value                    | false  |   |                                       |
| Multiplicity Configuration Class | Pre-compile time   | X | VARIANT-PRE-COMPILE                   |
|                                  | Link time  | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
|                                  | Post-build time  | – |                                       |
| Value Configuration Class        | Pre-compile time   | X | VARIANT-PRE-COMPILE                   |
|                                  | Link time  | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
|                                  | Post-build time  | – |                                       |
| Scope / Dependency               | scope: local   |   |                                       |

### 10.3.3 LinSMChannel

#### [ECUC\_LinSM\_00142] Definition of EcucParamConfContainerDef LinSMChannel

|  |   |  |  |
|--|---|--|--|
| Container Name                                 | LinSMChannel  |  |  |
| Parent Container                               | <a href="#">LinSMConfigSet</a>                        |  |  |
| Description                                    | Describes each LIN channel the LinSM is connected to. |  |  |
| Configuration Parameters                       |   |  |  |
| Included Parameters                            |   |  |  |
| Parameter Name                                 | Multiplicity  | ECUC ID  |  |
| <a href="#">LinSMConfirmationTimeout</a>       | 1   | <a href="#">[ECUC_LinSM_00144]</a>   |  |
| <a href="#">LinSMNodeType</a>                  | 1   | <a href="#">[ECUC_LinSM_00211]</a>   |  |
| <a href="#">LinSMSilenceAfterWakeupTimeout</a> | 0..1  | <a href="#">[ECUC_LinSM_00210]</a>   |  |
| <a href="#">LinSMTransceiverPassiveMode</a>    | 0..1  | <a href="#">[ECUC_LinSM_00202]</a>   |  |
| <a href="#">LinSMComMNetworkHandleRef</a>      | 1   | <a href="#">[ECUC_LinSM_00145]</a>   |  |
| Included Containers                            |   |  |  |
| Container Name                                 | Multiplicity  | Scope / Dependency   |  |
| <a href="#">LinSMSchedule</a>                  | 0..*  | The schedule references to a schedule that is located in the LinIf configuration. Moreover, the PDU groups are located in the COM configuration. Note that there are two references to PDU groups. The simple reason is that a PDU group is only allowed to contain one direction (TX or RX). Only applicable to LIN master nodes. |  |

#### [ECUC\_LinSM\_00144] Definition of EcucFloatParamDef LinSMConfirmationTimeout

|                           |  |   |  |
|---------------------------|--|---|--|
| Parameter Name            | LinSMConfirmationTimeout   |   |  |
| Parent Container          | <a href="#">LinSMChannel</a>   |   |  |
| Description               | Wakeup Confirmation Timeout in seconds. The timeout must be longer than a wakeup command on the bus (i.e. it is bit rate dependent). |   |  |
| Multiplicity              | 1  |   |  |
| Type                      | EcucFloatParamDef  |   |  |
| Range                     | [0 .. INF]   |   |  |
| Default value             | –  |   |  |
| Post-Build Variant Value  | false  |   |  |
| Value Configuration Class | Pre-compile time   | X | VARIANT-PRE-COMPILE                      |
|                           | Link time  | X | VARIANT-LINK-TIME,<br>VARIANT-POST-BUILD |
|                           | Post-build time  | – |  |
| Scope / Dependency        | scope: local   |   |  |

## [ECUC\_LinSM\_00211] Definition of EcucEnumerationParamDef LinSMNodeType

[

|                           |  |             |              |
|---------------------------|--|-------------|--------------|
| Parameter Name            | LinSMNodeType                                |             |              |
| Parent Container          | <a href="#">LinSMChannel</a>                 |             |              |
| Description               | Specifies the LIN node type of this channel. |             |              |
| Multiplicity              | 1  |             |              |
| Type                      | EcucEnumerationParamDef                      |             |              |
| Range                     | MASTER                                       | Master node |              |
|                           | SLAVE  | Slave node  |              |
| Post-Build Variant Value  | false  |             |              |
| Value Configuration Class | Pre-compile time                             | X           | All Variants |
|                           | Link time                                    | –           |              |
|                           | Post-build time                              | –           |              |
| Scope / Dependency        | scope: local                                 |             |              |

]

## [ECUC\_LinSM\_00210] Definition of EcucFloatParamDef LinSMSilenceAfterWakeupTimeout

[

|                           |  |   |  |
|---------------------------|--|---|--|
| Parameter Name            | LinSMSilenceAfterWakeupTimeout   |   |  |
| Parent Container          | <a href="#">LinSMChannel</a>   |   |  |
| Description               | Timeout in seconds after a failed wakeup sequence until a new wakeup process is started.                 |   |  |
| Multiplicity              | 0..1   |   |  |
| Type                      | EcucFloatParamDef  |   |  |
| Range                     | [0 .. INF]   |   |  |
| Default value             | –  |   |  |
| Post-Build Variant Value  | false  |   |  |
| Value Configuration Class | Pre-compile time   | X | VARIANT-PRE-COMPILE                      |
|                           | Link time  | X | VARIANT-LINK-TIME,<br>VARIANT-POST-BUILD |
|                           | Post-build time  | – |  |
| Scope / Dependency        | scope: local   |   |  |
|                           | dependency: This parameter is only applicable for LIN slave nodes, depending on parameter LinSMNodeType. |   |  |

]

## [ECUC\_LinSM\_00202] Definition of EcucBooleanParamDef LinSMTransceiverPassiveMode

[

|                  |  |  |  |
|------------------|--|--|--|
| Parameter Name   | LinSMTransceiverPassiveMode  |  |  |
| Parent Container | <a href="#">LinSMChannel</a>   |  |  |
| Description      | Selects STANDBY (true) or SLEEP (false) transceiver mode when entering LINSM_NO_COM. |  |  |
| Multiplicity     | 0..1   |  |  |





|                                  |                     |   |  |
|----------------------------------|---------------------|---|--|
| Type                             | EcucBooleanParamDef |   |  |
| Default value                    | –                   |   |  |
| Post-Build Variant Multiplicity  | false               |   |  |
| Post-Build Variant Value         | false               |   |  |
| Multiplicity Configuration Class | Pre-compile time    | X | VARIANT-PRE-COMPILE                      |
|                                  | Link time           | X | VARIANT-LINK-TIME,<br>VARIANT-POST-BUILD |
|                                  | Post-build time     | – |  |
| Value Configuration Class        | Pre-compile time    | X | VARIANT-PRE-COMPILE                      |
|                                  | Link time           | X | VARIANT-LINK-TIME,<br>VARIANT-POST-BUILD |
|                                  | Post-build time     | – |  |
| Scope / Dependency               | scope: local        |   |  |

]

#### [ECUC\_LinSM\_00145] Definition of EcucReferenceDef LinSMComMNetworkHandleRef [

|                           |  |   |  |
|---------------------------|--|---|--|
| Parameter Name            | LinSMComMNetworkHandleRef  |   |  |
| Parent Container          | <a href="#">LinSMChannel</a>   |   |  |
| Description               | Unique handle to identify one certain LIN network. Reference to one of the network handles configured in the ComM. |   |  |
| Multiplicity              | 1  |   |  |
| Type                      | Symbolic name reference to ComMChannel   |   |  |
| Post-Build Variant Value  | false  |   |  |
| Value Configuration Class | Pre-compile time   | X | VARIANT-PRE-COMPILE                      |
|                           | Link time  | X | VARIANT-LINK-TIME,<br>VARIANT-POST-BUILD |
|                           | Post-build time  | – |  |
| Scope / Dependency        | scope: local   |   |  |

]

### 10.3.4 LinSMGeneral

#### [ECUC\_LinSM\_00139] Definition of EcucParamConfContainerDef LinSMGeneral [

|                          |   |  |  |
|--------------------------|---|--|--|
| Container Name           | LinSMGeneral  |  |  |
| Parent Container         | <a href="#">LinSM</a>   |  |  |
| Description              | This container contains general parameters of LIN State Manager module. |  |  |
| Configuration Parameters |   |  |  |

| Included Parameters                       |              |                    |
|---|--------------|--------------------|
| Parameter Name                            | Multiplicity | ECUC ID            |
| <a href="#">LinSMDevErrorDetect</a>       | 1            | [ECUC_LinSM_00206] |
| <a href="#">LinSMMainProcessingPeriod</a> | 1            | [ECUC_LinSM_00141] |
| <a href="#">LinSMVersionInfoApi</a>       | 1            | [ECUC_LinSM_00140] |

|                        |
|------------------------|
| No Included Containers |
|------------------------|

]

## [ECUC\_LinSM\_00206] Definition of EcucBooleanParamDef LinSMDevErrorDetect [

|                           |   |   |              |
|---------------------------|---|---|--------------|
| Parameter Name            | LinSMDevErrorDetect   |   |              |
| Parent Container          | <a href="#">LinSMGeneral</a>  |   |              |
| Description               | Switches the development error detection and notification on or off. <ul style="list-style-type: none"> <li>• true: detection and notification is enabled.</li> <li>• false: detection and notification is disabled.</li> </ul> |   |              |
| Multiplicity              | 1   |   |              |
| Type                      | EcucBooleanParamDef   |   |              |
| Default value             | false   |   |              |
| Post-Build Variant Value  | false   |   |              |
| Value Configuration Class | Pre-compile time  | X | All Variants |
|                           | Link time   | – |              |
|                           | Post-build time   | – |              |
| Scope / Dependency        | scope: local  |   |              |

]

## [ECUC\_LinSM\_00141] Definition of EcucFloatParamDef LinSMMainProcessingPeriod [

|                           |   |   |              |
|---------------------------|---|---|--------------|
| Parameter Name            | LinSMMainProcessingPeriod                           |   |              |
| Parent Container          | <a href="#">LinSMGeneral</a>                        |   |              |
| Description               | Fixed period that the MainFunction shall be called. |   |              |
| Multiplicity              | 1   |   |              |
| Type                      | EcucFloatParamDef                                   |   |              |
| Range                     | ]0 .. INF[  |   |              |
| Default value             | –   |   |              |
| Post-Build Variant Value  | false   |   |              |
| Value Configuration Class | Pre-compile time                                    | X | All Variants |
|                           | Link time   | – |              |
|                           | Post-build time                                     | – |              |
| Scope / Dependency        | scope: local  |   |              |

]

## [ECUC\_LinSM\_00140] Definition of EcucBooleanParamDef LinSMVersionInfoApi

[

|                           |   |   |              |
|---------------------------|---|---|--------------|
| Parameter Name            | LinSMVersionInfoApi                                   |   |              |
| Parent Container          | <a href="#">LinSMGeneral</a>                          |   |              |
| Description               | Switches the LinSM_GetVersionInfo function ON or OFF. |   |              |
| Multiplicity              | 1   |   |              |
| Type                      | EcucBooleanParamDef                                   |   |              |
| Default value             | false   |   |              |
| Post-Build Variant Value  | false   |   |              |
| Value Configuration Class | Pre-compile time                                      | X | All Variants |
|                           | Link time   | – |              |
|                           | Post-build time                                       | – |              |
| Scope / Dependency        | scope: local  |   |              |

]

### 10.3.5 LinSMSchedule

## [ECUC\_LinSM\_00146] Definition of EcucParamConfContainerDef LinSMSchedule

[

|                          |  |
|--------------------------|--|
| Container Name           | LinSMSchedule  |
| Parent Container         | <a href="#">LinSMChannel</a>   |
| Description              | The schedule references to a schedule that is located in the LinIf configuration. Moreover, the PDU groups are located in the COM configuration. Note that there are two references to PDU groups. The simple reason is that a PDU group is only allowed to contain one direction (TX or RX). Only applicable to LIN master nodes. |
| Configuration Parameters |  |

| Included Parameters                   |              |                    |
|---------------------------------------|--------------|--------------------|
| Parameter Name                        | Multiplicity | ECUC ID            |
| <a href="#">LinSMScheduleIndex</a>    | 1            | [ECUC_LinSM_00001] |
| <a href="#">LinSMScheduleIndexRef</a> | 1            | [ECUC_LinSM_00149] |

|                        |
|------------------------|
| No Included Containers |
|------------------------|

]



## [ECUC\_LinSM\_00001] Definition of EcucIntegerParamDef LinSMScheduleIndex

[

|                           |  |   |              |
|---------------------------|--|---|--------------|
| Parameter Name            | LinSMScheduleIndex   |   |              |
| Parent Container          | <a href="#">LinSMSchedule</a>  |   |              |
| Description               | This index parameter can be used by the BswM as a SymbolicNameReference target. The LinSM just forwards the request from the BswM to Linlf. Note that the value of the LinSMScheduleIndex shall be the same as the value from the Linlf. |   |              |
| Multiplicity              | 1  |   |              |
| Type                      | EcucIntegerParamDef (Symbolic Name generated for this parameter)   |   |              |
| Range                     | 0 .. 255   |   |              |
| Default value             | –  |   |              |
| Post-Build Variant Value  | false  |   |              |
| Value Configuration Class | Pre-compile time   | X | All Variants |
|                           | Link time  | – |              |
|                           | Post-build time  | – |              |
| Scope / Dependency        | scope: local<br>withAuto = true  |   |              |

]

## [ECUC\_LinSM\_00149] Definition of EcucReferenceDef LinSMScheduleIndexRef

[

|                                  |  |   |                     |
|----------------------------------|--|---|---------------------|
| <b>Parameter Name</b>            | LinSMScheduleIndexRef                                    |   |                     |
| <b>Parent Container</b>          | <a href="#">LinSMSchedule</a>                            |   |                     |
| <b>Description</b>               | Reference to a schedule table in the LinIf configuration |   |                     |
| <b>Multiplicity</b>              | 1  |   |                     |
| <b>Type</b>                      | Symbolic name reference to LinIfScheduleTable            |   |                     |
| <b>Post-Build Variant Value</b>  | true   |   |                     |
| <b>Value Configuration Class</b> | <b>Pre-compile time</b>                                  | X | VARIANT-PRE-COMPILE |
|                                  | <b>Link time</b>   | X | VARIANT-LINK-TIME   |
|                                  | <b>Post-build time</b>                                   | X | VARIANT-POST-BUILD  |
| <b>Scope / Dependency</b>        | scope: local   |   |                     |

]

## 10.4 Published Information

For details refer to the chapter 10.3 “Published Information” in [3, General Requirements on Basic Software Modules].

## A Not applicable requirements

### [SWS\_LinSM\_NA\_00211]

*Upstream requirements:* SRS\_BSW\_00404, SRS\_BSW\_00405, SRS\_BSW\_00170, SRS\_BSW\_00399, SRS\_BSW\_00400, SRS\_BSW\_00375, SRS\_BSW\_00416, SRS\_BSW\_00437, SRS\_BSW\_00168, SRS\_BSW\_00425, SRS\_BSW\_00432, SRS\_BSW\_00433, SRS\_BSW\_00422, SRS\_BSW\_00417, SRS\_BSW\_00161, SRS\_BSW\_00162, SRS\_BSW\_00005, SRS\_BSW\_00415, SRS\_BSW\_00343, SRS\_BSW\_00439, SRS\_BSW\_00359, SRS\_BSW\_00360, SRS\_BSW\_00331, SRS\_BSW\_00010, SRS\_BSW\_00333, SRS\_BSW\_00321, SRS\_BSW\_00341, SRS\_Lin\_01590, SRS\_Lin\_01560, SRS\_Lin\_01577, SRS\_BSW\_00438

[These requirements are not applicable to this specification.]

### A.1 Change history of AUTOSAR traceable items

Please note that the lists in this chapter also include traceable items that have been removed from the specification in a later version. These items do not appear as hyperlinks in the document.

### A.2 Traceable item history of this document according to AUTOSAR Release R22-11

#### A.2.1 Added Specification Items in R22-11

[SWS\_LinSM\_00001] [SWS\_LinSM\_00002] [SWS\_LinSM\_00005] [SWS\_LinSM\_00013] [SWS\_LinSM\_00019] [SWS\_LinSM\_00020] [SWS\_LinSM\_00021] [SWS\_LinSM\_00022] [SWS\_LinSM\_00024] [SWS\_LinSM\_00025] [SWS\_LinSM\_00026] [SWS\_LinSM\_00027] [SWS\_LinSM\_00028] [SWS\_LinSM\_00032] [SWS\_LinSM\_00033] [SWS\_LinSM\_00035] [SWS\_LinSM\_00036] [SWS\_LinSM\_00043] [SWS\_LinSM\_00046] [SWS\_LinSM\_00047] [SWS\_LinSM\_00049] [SWS\_LinSM\_00053] [SWS\_LinSM\_00073] [SWS\_LinSM\_00079] [SWS\_LinSM\_00085] [SWS\_LinSM\_00086] [SWS\_LinSM\_00100] [SWS\_LinSM\_00101] [SWS\_LinSM\_00102] [SWS\_LinSM\_00103] [SWS\_LinSM\_00105] [SWS\_LinSM\_00113] [SWS\_LinSM\_00114] [SWS\_LinSM\_00115] [SWS\_LinSM\_00116] [SWS\_LinSM\_00117] [SWS\_LinSM\_00119] [SWS\_LinSM\_00122] [SWS\_LinSM\_00123] [SWS\_LinSM\_00124] [SWS\_LinSM\_00125] [SWS\_LinSM\_00126] [SWS\_LinSM\_00127] [SWS\_LinSM\_00128] [SWS\_LinSM\_00129] [SWS\_LinSM\_00130] [SWS\_LinSM\_00131] [SWS\_LinSM\_00132] [SWS\_LinSM\_00133] [SWS\_LinSM\_00134] [SWS\_LinSM\_00135] [SWS\_LinSM\_00136] [SWS\_LinSM\_00137] [SWS\_LinSM\_00138] [SWS\_LinSM\_00151] [SWS\_LinSM\_00152] [SWS\_LinSM\_00154] [SWS\_LinSM\_00155] [SWS\_LinSM\_00156] [SWS\_LinSM\_00157] [SWS\_LinSM\_00159] [SWS\_LinSM\_00160] [SWS\_

[LinSM\\_00161](#)] [\[SWS\\_LinSM\\_00162](#)] [\[SWS\\_LinSM\\_00163](#)] [\[SWS\\_LinSM\\_00164](#)  
[\[SWS\\_LinSM\\_00166](#)] [\[SWS\\_LinSM\\_00168](#)] [\[SWS\\_LinSM\\_00170](#)] [\[SWS\\_LinSM\\_](#)  
[00172](#)] [\[SWS\\_LinSM\\_00173](#)] [\[SWS\\_LinSM\\_00175](#)] [\[SWS\\_LinSM\\_00176](#)] [\[SWS\\_](#)  
[LinSM\\_00177](#)] [\[SWS\\_LinSM\\_00178](#)] [\[SWS\\_LinSM\\_00180](#)] [\[SWS\\_LinSM\\_00181](#)  
[\[SWS\\_LinSM\\_00182](#)] [\[SWS\\_LinSM\\_00183](#)] [\[SWS\\_LinSM\\_00191](#)] [\[SWS\\_LinSM\\_](#)  
[00192](#)] [\[SWS\\_LinSM\\_00193](#)] [\[SWS\\_LinSM\\_00196](#)] [\[SWS\\_LinSM\\_00201](#)] [\[SWS\\_](#)  
[LinSM\\_00202](#)] [\[SWS\\_LinSM\\_00203](#)] [\[SWS\\_LinSM\\_00204](#)] [\[SWS\\_LinSM\\_00205](#)  
[\[SWS\\_LinSM\\_00206](#)] [\[SWS\\_LinSM\\_00207](#)] [\[SWS\\_LinSM\\_00208](#)] [\[SWS\\_LinSM\\_](#)  
[00213](#)] [\[SWS\\_LinSM\\_00214](#)] [\[SWS\\_LinSM\\_00215](#)] [\[SWS\\_LinSM\\_00216](#)] [\[SWS\\_](#)  
[LinSM\\_00219](#)] [\[SWS\\_LinSM\\_00220](#)] [\[SWS\\_LinSM\\_00221](#)] [\[SWS\\_LinSM\\_00223](#)  
[\[SWS\\_LinSM\\_00224](#)] [\[SWS\\_LinSM\\_00229](#)] [\[SWS\\_LinSM\\_00230](#)] [\[SWS\\_LinSM\\_](#)  
[00231](#)] [\[SWS\\_LinSM\\_00232](#)] [\[SWS\\_LinSM\\_00233](#)] [\[SWS\\_LinSM\\_00234](#)] [\[SWS\\_](#)  
[LinSM\\_00235](#)] [\[SWS\\_LinSM\\_00236](#)] [\[SWS\\_LinSM\\_00237](#)] [\[SWS\\_LinSM\\_00239](#)  
[\[SWS\\_LinSM\\_00240](#)] [\[SWS\\_LinSM\\_00241](#)] [\[SWS\\_LinSM\\_00242](#)] [\[SWS\\_LinSM\\_](#)  
[00243](#)] [\[SWS\\_LinSM\\_00301](#)] [\[SWS\\_LinSM\\_00302](#)] [\[SWS\\_LinSM\\_00304](#)] [\[SWS\\_](#)  
[LinSM\\_00305](#)] [\[SWS\\_LinSM\\_00307](#)] [\[SWS\\_LinSM\\_10208](#)] [\[SWS\\_LinSM\\_10209](#)  
[\[SWS\\_LinSM\\_10211](#)] [\[SWS\\_LinSM\\_91000](#)] [\[SWS\\_LinSM\\_NA\\_00211](#)]

### A.2.2 Changed Specification Items in R22-11

none

### A.2.3 Deleted Specification Items in R22-11

none

## A.3 Traceable item history of this document according to AUTOSAR Release R23-11

### A.3.1 Added Specification Items in R23-11

[\[SWS\\_LinSM\\_00165\]](#)

### A.3.2 Changed Specification Items in R23-11

[\[SWS\\_LinSM\\_00047](#)] [\[SWS\\_LinSM\\_00100](#)] [\[SWS\\_LinSM\\_00102](#)] [\[SWS\\_LinSM\\_](#)  
[00113](#)] [\[SWS\\_LinSM\\_00122](#)] [\[SWS\\_LinSM\\_00126](#)] [\[SWS\\_LinSM\\_00138](#)] [\[SWS\\_](#)  
[LinSM\\_00154](#)] [\[SWS\\_LinSM\\_00159](#)] [\[SWS\\_LinSM\\_00162](#)] [\[SWS\\_LinSM\\_00163](#)  
[\[SWS\\_LinSM\\_00175](#)] [\[SWS\\_LinSM\\_00219](#)] [\[SWS\\_LinSM\\_00220](#)] [\[SWS\\_LinSM\\_](#)  
[00221](#)] [\[SWS\\_LinSM\\_00229](#)] [\[SWS\\_LinSM\\_00235](#)] [\[SWS\\_LinSM\\_00236](#)] [\[SWS\\_](#)

[LinSM\\_00237](#)] [[SWS\\_LinSM\\_00241](#)] [[SWS\\_LinSM\\_00242](#)] [[SWS\\_LinSM\\_00243](#)]  
[[SWS\\_LinSM\\_00304](#)]

### **A.3.3 Deleted Specification Items in R23-11**

[SWS\_LinSM\_00101] [SWS\_LinSM\_00103] [SWS\_LinSM\_00170] [SWS\_LinSM\_-  
00172] [SWS\_LinSM\_00215] [SWS\_LinSM\_00307]

## **A.4 Traceable item history of this document according to AUTOSAR Release R24-11**

### **A.4.1 Added Specification Items in R24-11**

none

### **A.4.2 Changed Specification Items in R24-11**

none

### **A.4.3 Deleted Specification Items in R24-11**

none