

MICROSAR LIN Transceiver Driver

Technical Reference

Tle7259

Version 3.01.00

Authors	Edgar Tongoona, Nessim Ben Ammar
Status	Released

Document Information

History

Author	Date	Version	Remarks
Edgar Tongoona	2013-01-02	2.00.00	Creation
Edgar Tongoona	2013-09-27	2.01.00	Removed DEM and corrected error codes
Edgar Tongoona	2014-02-07	2.02.00	Added ICU notification
Edgar Tongoona	2014-09-30	3.00.00	Added Post-build Selectable and wakeup handling compliance to AR4.1.x
Nessim Ben Ammar	2016-11-24	3.00.01	Added Appl_LinTrcv_30_Tle7259_Wait
Nessim Ben Ammar	2017-03-25	3.01.00	Added separate wakeup support by pin Added sleep init workaround

Reference Documents

No.	Source	Title	Version
[1]	AUTOSAR	AUTOSAR_SWS_LIN_TransceiverDriver.pdf	V1.2.0
[2]	AUTOSAR	AUTOSAR_SWS_DET.pdf	V3.2.0
[3]	AUTOSAR	AUTOSAR_SWS_DEM.pdf	V4.2.0
[4]	AUTOSAR	AUTOSAR_TR_BSWModuleList.pdf	V1.6.0
[5]	Infineon AG	TLE 7259 G, LIN Transceiver, Data Sheet	Rev 2.0
[6]	Philips	TJA1020 LIN Transceiver, Data Sheet	2004 Jan 13
[7]	Atmel	ATA6661 LIN Transceiver, Data Sheet	2009 Feb
[8]	Freescale	MC33661 LIN Transceiver, Data Sheet	Rev 7.0

Scope of the Document

This technical reference describes the general use of the MICROSAR LIN Transceiver Driver.



Caution

We have configured the programs in accordance with your specifications in the questionnaire. Whereas the programs do support other configurations than the one specified in your questionnaire, Vector's release of the programs delivered to your company is expressly restricted to the configuration you have specified in the questionnaire.



Caution

This symbol calls your attention to warnings.

Contents

1	Component History	7
2	Introduction.....	8
2.1	Architecture Overview	9
2.2	Supported Transceivers	10
3	Functional Description	12
3.1	Features	12
3.1.1	Deviations	12
3.1.2	Additions/ Extensions.....	12
3.1.2.1	Memory initialization	12
3.2	Initialization	12
3.3	Operation modes	13
3.4	Set operation mode.....	13
3.5	Get operation mode	14
3.6	Wakeup event detection.....	14
3.7	Error Handling.....	15
3.7.1	Development Error Reporting.....	15
3.7.1.1	Parameter Checking	16
4	Integration.....	17
4.1	Scope of Delivery.....	17
4.1.1	Static Files	17
4.1.2	Dynamic Files	17
4.2	Compiler Abstraction and Memory Mapping.....	18
4.3	Critical Sections	18
4.4	Dependency to DIO Driver	19
1.1	Dependency to ICU Driver	19
5	API Description.....	20
5.1	Type Definitions	20
5.2	Services provided by LINTRCV.....	21
5.2.1	LinTrcv_30_Tle7259_InitMemory	21
5.2.2	LinTrcv_30_Tle7259_Init.....	21
5.2.3	LinTrcv_30_Tle7259_SetOpMode.....	22
5.2.4	LinTrcv_30_Tle7259_GetOpMode	23
5.2.5	LinTrcv_30_Tle7259_GetBusWuReason	23
5.2.6	LinTrcv_30_Tle7259_SetWakeupMode.....	24
5.2.7	LinTrcv_30_Tle7259_CheckWakeup.....	25

5.2.8	LinTrcv_30_Tle7259_GetVersionInfo	25
5.3	Services used by LINTRCV	26
5.4	Callback Functions.....	26
5.4.1	Appl_LinTrcv_30_Tle7259_Wait.....	26
5.5	Configurable Interfaces	27
6	Configuration	28
6.1	Configuration Variants.....	28
6.2	Configuration with a GCE.....	28
6.2.1	LinTrcvGeneral.....	28
6.2.2	LinTrcvChannel	29
6.2.3	LinTrcvDioAccess	31
6.2.4	LinTrcvSpiSequence	31
7	Glossary and Abbreviations	33
7.1	Glossary	33
7.2	Abbreviations	33
8	Contact.....	34

Illustrations

Figure 2-1	AUTOSAR 4.x Architecture Overview	9
Figure 2-2	AUTOSAR 4.x architecture	9
Figure 2-3	Interfaces to adjacent modules of the LINTRCV	10

Tables

Table 1-1	Component history.....	7
Table 3-1	Supported AUTOSAR standard conform features	12
Table 3-2	Not supported AUTOSAR standard conform features	12
Table 3-3	Service IDs	15
Table 3-4	Errors reported to DET	15
Table 3-5	Development Error Reporting: Assignment of checks to services	16
Table 4-1	Static files	17
Table 4-2	Generated files	17
Table 4-3	Compiler abstraction and memory mapping.....	18
Table 5-1	Type definitions.....	21
Table 5-2	LinTrcv_30_Tle7259_InitMemory	21
Table 5-3	LinTrcv_30_Tle7259_Init	22
Table 5-4	LinTrcv_30_Tle7259_SetOpMode	22
Table 5-5	LinTrcv_30_Tle7259_GetOpMode	23
Table 5-6	LinTrcv_30_Tle7259_GetBusWuReason	24
Table 5-7	LinTrcv_30_Tle7259_SetWakeupMode	25
Table 5-8	LinTrcv_30_Tle7259_CheckWakeup	25
Table 5-9	LinTrcv_30_Tle7259_GetVersionInfo.....	26
Table 5-10	Services used by the LINTRCV	26
Table 6-1	Container LinTrcv	28
Table 6-2	Container LinTrcvGeneral	28
Table 6-3	Attributes of LinTrcvGeneral	29
Table 6-4	Container LinTrcvChannel	29
Table 6-5	Attributes of LinTrcvChannel.....	30
Table 6-6	Sub Containers of LinTrcvChannel	30
Table 6-7	Sub Containers of LinTrcvAccess	30
Table 6-8	Container LinTrcvDioAccess.....	31
Table 6-9	Attributes of LinTrcvDioAccess	31
Table 6-10	Container LinTrcvSpiSequence	31
Table 6-11	Attributes of LinTrcvSpiSequence	32
Table 7-1	Glossary	33
Table 7-2	Abbreviations.....	33

1 Component History

The component history gives an overview over the important milestones that are supported in the different versions of the component.

Component Version	New Features
2.00.00	first version of the module supporting AUTOSAR R4.0.3
3.00.00	Support of Post-Build-Selectable Support of external wakeup handling according to AUTOSAR R4.1.2

Table 1-1 Component history

2 Introduction

This document describes the functionality, API and configuration of the AUTOSAR BSW module LINTRCV as specified in [1].

Supported AUTOSAR Release*:	4	
Supported Configuration Variants:	pre-compile, post build selectable	
Vendor ID:	LINTRCV_VENDOR_ID	30 decimal (= Vector-Informatik, according to HIS)
Module ID:	LINTRCV_MODULE_ID	64 decimal (according to ref. [4])

* For the precise AUTOSAR Release 4.x please see the release specific documentation.

The MICROSAR LIN Transceiver Driver provides an abstraction layer for the used LIN transceiver hardware. It offers a hardware independent interface to the upper layer components.

The following figure shows where the LINTRCV is located in the AUTOSAR architecture.

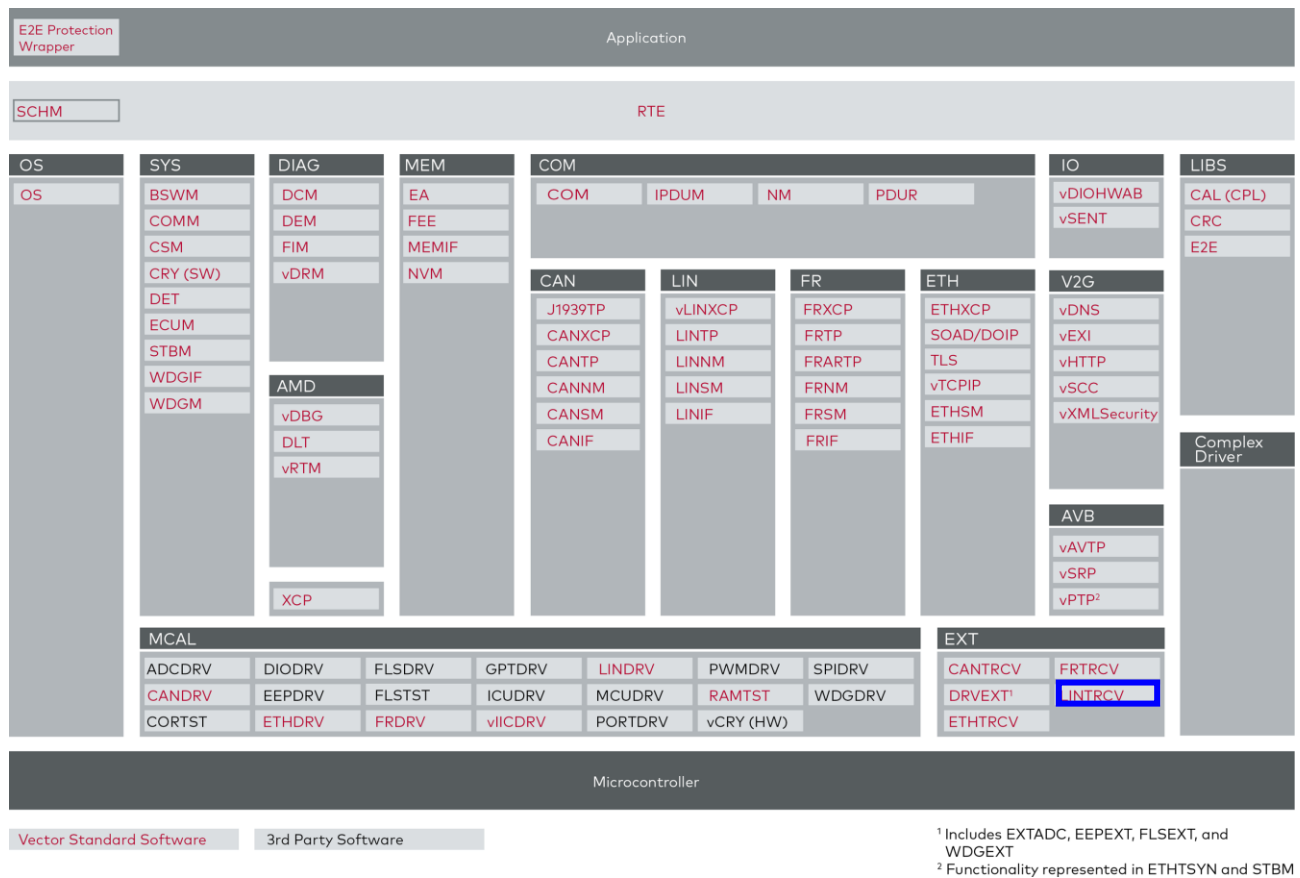


Figure 2-2 AUTOSAR 4.x architecture

The next figure shows the interfaces to adjacent modules of the LINTRCV. These interfaces are described in chapter 5.

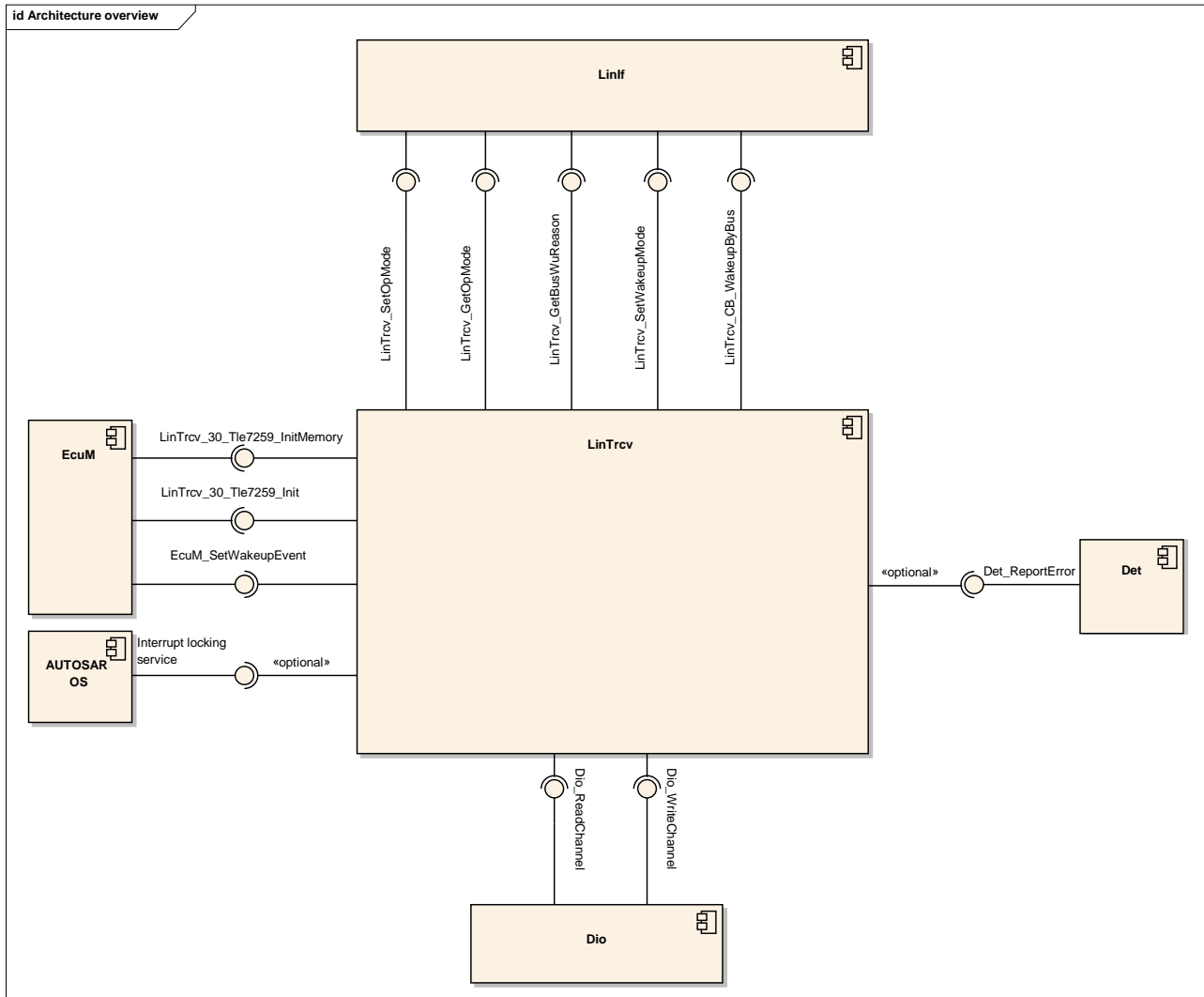


Figure 2-3 Interfaces to adjacent modules of the LINTRCV

2.2 Supported Transceivers

Due to their compatibility the Tle7259 transceiver driver can be used with other transceivers. The transceiver's operation mode must be changeable by changing the level on the main enable pin (pin that enables the normal mode also called EN, SLP_N...). The transceiver's behavior must correspond to the following table:

Operation mode / state	EN PIN	Rx PIN	Tx PIN
Normal Mode	High	-	-
Sleep Mode (No Wakeups)	Low	High	High
Wakeup By Bus Detected	Low	Low	High / Weak Pull-Down
Wakeup By Pin Detected	Low	Low	Low / Strong Pull-Down


Caution

The Tx and Rx pins must be configured as digital inputs with pullups. The pins can be configured as communication pins during the communication phases and as standard inputs with pullups during the sleep/standby phases.

3 Functional Description

3.1 Features

The features listed in the following tables cover the complete functionality specified for the LINTRCV.

The AUTOSAR standard functionality is specified in [1], the corresponding features are listed in the tables

- > Table 3-1 Supported AUTOSAR standard conform features
- > Table 3-2 Not supported AUTOSAR standard conform features

The following features specified in [1] are supported:

Supported AUTOSAR Standard Conform Features
LIN Transceiver initialization.
LIN Transceiver control via DIO.
Detection of wakeup.
Reading and setting operation mode of LIN Transceiver.

Table 3-1 Supported AUTOSAR standard conform features

3.1.1 Deviations

The following features specified in [1] are not supported:

Not Supported AUTOSAR Standard Conform Features
LIN Transceiver self diagnostics.

Table 3-2 Not supported AUTOSAR standard conform features

3.1.2 Additions/ Extensions

3.1.2.1 Memory initialization

To have an independent memory initialization for this BSW module the additional function `LinTrcv_30_Tle7259_InitMemory()` was added. This must be called before normal initialization if initialized variables are not initialized during startup phase.

3.2 Initialization

After power on the LIN transceiver hardware has to be initialized. There for the LIN Transceiver Driver provides two service functions.

The function `LinTrcv_30_Tle7259_InitMemory()` initializes all initialized variables of the LIN Transceiver Driver. This function has to be called after power on or reset before any other function in case initialized variables are not set after power on or reset (i.e. by the startup code).

The function `LinTrcv_30_Tle7259_Init()` initializes all LIN Transceiver Driver channels which are selected by the generation tool. Each LIN transceiver is switched into the configured channel specific init mode. Operation mode after power on or reset can be normal or sleep.

**Caution**

Some transceiver hardware cannot be switched directly to sleep mode after power-on. For these, the `LinTrcvSleepInitWorkaround` can be enabled: the transceiver hardware will then be switched into normal mode then to sleep mode. The user must ensure that the Tx level remains high during the initialization otherwise spikes will propagate to the bus.

3.3 Operation modes

The operation modes are described as follows:

Operation Mode	Communication	Wakeup Detection	MCU power
Normal	Enabled	Disabled	Enabled
Standby	Disabled	Enabled	Enabled
Sleep	Disabled	Enabled	Disabled

The LIN transceiver driver supports the normal and sleep operation modes only. The standby mode is merged with the sleep mode: the hardware layout will determine whether the MCU power supply is cut or not.

**Caution**

Once the transceiver hardware detects a wakeup, it switches to a “Standby” mode which is different from the standby mode definition in AUTOSAR. The driver then considers the hardware to be in sleep mode with a wakeup event.

3.4 Set operation mode

The operation mode of the LIN transceiver hardware is changed by service function `LinTrcv_30_Tle7259_SetOpMode()`. It can be switched from normal into sleep mode and vice-versa.



Caution

Each operation mode transition requires a certain time to be complete. Depending on how fast the MCU is running, the MCU might be trying to communicate although the transceiver hardware is not ready yet. This can be avoided by configuring a non-zero wait count value: the `Appl_LinTrcv_30_GenericLin_Wait` call-out will then be invoked after each state transition allowing the user to implement the required delay to ensure the proper functioning.

3.5 Get operation mode

To get the current operation mode of a specified LIN transceiver hardware, the service function `LinTrcv_30_Tle7259_GetOpMode()` has to be called.

3.6 Wakeup event detection

Call `LinTrcv_30_Tle7259_CheckWakeup()` to check if the transceiver hardware has detected a wakeup by bus or pin event. It always stores the wakeup event and returns `E_OK` only if:

- The LIN Transceiver channel is in sleep mode
- A wakeup by bus or by pin was detected
- Wakeup mode is enabled for the transceiver channel

Otherwise, it will return `E_NOT_OK`.

If the above conditions are met, the EcuM and the LIN Interface is informed by calling `EcuM_SetWakeupEvent()` and `LinIf_WakeupConfirmation()`.

The service function `LinTrcv_30_Tle7259_GetBusWakeupReason()` returns the reason which caused the wakeup.

The service function `LinTrcv_30_Tle7259_SetWakeupMode()` sets the wakeup mode requested by the LIN Interface.

3.7 Error Handling

3.7.1 Development Error Reporting

By default, development errors are reported to the DET using the service `Det_ReportError()` as specified in [2], if development error reporting is enabled (i.e. pre-compile parameter `LINTRCV_DEV_ERROR_DETECT==STD_ON`).

If another module is used for development error reporting, the function prototype for reporting the error can be configured by the integrator, but must have the same signature as the service `Det_ReportError()`.

The reported LINTRCV ID is 64.

The reported service IDs identify the services which are described in 5.2. The following table presents the service IDs and the related services:

Service ID	Service
0x00	LinTrcv_30_Tle7259_Init
0x01	LinTrcv_30_Tle7259_SetOpMode
0x02	LinTrcv_30_Tle7259_GetOpMode
0x03	LinTrcv_30_Tle7259_GetBusWuReason
0x04	LinTrcv_30_Tle7259_GetVersionInfo
0x05	LinTrcv_30_Tle7259_SetWakeupMode
0x07	LinTrcv_30_Tle7259_CheckWakeup

Table 3-3 Service IDs

The errors reported to DET are described in the following table:

Error Code	Description
0x01 LINTRCV_30_TLE7259_E_INVALID_LIN_NETWORK	API called with wrong parameter for LIN network
0x02 LINTRCV_30_TLE7259_E_PARAM_POINTER	API called with null pointer parameter
0x11 LINTRCV_30_TLE7259_E_UNINIT	API service used without initialization
0x21 LINTRCV_30_TLE7259_E_TRCV_NOT_SLEEP	API service called in wrong transceiver operation mode
0x22 LINTRCV_30_TLE7259_E_TRCV_NOT_NORMAL	API service called in wrong transceiver operation mode
0x23 LINTRCV_30_TLE7259_E_PARAM_TRCV_WAKEUP_MODE	API service called with invalid parameter for transceiver wakeup mode
0x24 LINTRCV_30_TLE7259_E_ALREADY_INITIALIZED	API service called with invalid parameter for operation mode
0x25 LINTRCV_30_TLE7259_E_CONFIG	API service called with invalid mode because optional transition is not enabled

Table 3-4 Errors reported to DET

3.7.1.1 Parameter Checking

AUTOSAR requires that API functions check the validity of their parameters. The checks in Table 3-5 are internal parameter checks of the API functions. These checks are for development error reporting and can be en-/disabled by means of en-/disabling the DET reporting via the parameter `LINTRCV_DEV_ERROR_DETECT`.

The following table shows which parameter checks are performed on which services:

Service	Check			
	LinTrcvIndex	OpMode	Reason	TrcvWakeupMode
<code>LinTrcv_30_Tle7259_SetOpMode()</code>	■	■		
<code>LinTrcv_30_Tle7259_GetOpMode()</code>	■	■		
<code>LinTrcv_30_Tle7259_GetBusWuReason()</code>	■		■	
<code>LinTrcv_30_Tle7259_SetWakeupMode()</code>	■			■
<code>LinTrcv_30_Tle7259_CheckWakeup()</code>	■			

Table 3-5 Development Error Reporting: Assignment of checks to services

4 Integration

This chapter gives necessary information for the integration of the MICROSAR LINTRCV into an application environment of an ECU. This component is only applicable on LIN transceiver hardware Infineon Tle7259.

4.1 Scope of Delivery

The delivery of the LINTRCV contains the files which are described in the chapters 4.1.1 and 4.1.2:

4.1.1 Static Files

File Name	Source Code Delivery	Object Code Delivery	Description
LinTrcv_30_Tle7259.h	■		Header file which has to be included by higher layers.
LinTrcv_30_Tle7259.c	■		Implementation

Table 4-1 Static files

4.1.2 Dynamic Files

The dynamic files are generated by the configuration tool [config tool].

File Name	Description
LinTrcv_30_Tle7259_Cfg.h	Header file contains define definitions and external data declarations. It is included by LinTrcv_30_Tle7259.h.
LinTrcv_30_Tle7259_Cfg.c	Contains the pre-compile and post build selectable configuration data.

Table 4-2 Generated files

4.2 Compiler Abstraction and Memory Mapping

The objects (e.g. variables, functions, constants) are declared by compiler independent definitions – the compiler abstraction definitions. Each compiler abstraction definition is assigned to a memory section.

The following table contains the memory section names and the compiler abstraction definitions of the LINTRCV and illustrates their assignment among each other.

Memory Mapping Sections	Compiler Abstraction Definitions	LINTRCV_30_TLE7259_VAR	LINTRCV_30_TLE7259_VAR_ZERO_INIT	LINTRCV_30_TLE7259_APPL_VAR	LINTRCV_30_TLE7259_CONST	LINTRCV_30_TLE7259_CODE	LINTRCV_30_TLE7259_APPL_CODE
LINTRCV_30_TLE7259_START_SEC_VAR_NOINIT_UNSPECIFIED LINTRCV_30_TLE7259_STOP_SEC_VAR_NOINIT_UNSPECIFIED		■					
LINTRCV_30_TLE7259_START_SEC_VAR_ZERO_INIT_8BIT LINTRCV_30_TLE7259_STOP_SEC_VAR_ZERO_INIT_8BIT			■				
LINTRCV_30_TLE7259_START_SEC_CONST_8BIT LINTRCV_30_TLE7259_STOP_SEC_CONST_8BIT					■		
LINTRCV_30_TLE7259_START_SEC_CONST_32BIT LINTRCV_30_TLE7259_STOP_SEC_CONST_32BIT					■		
LINTRCV_30_TLE7259_START_SEC_CONST_UNSPECIFIED LINTRCV_30_TLE7259_STOP_SEC_CONST_UNSPECIFIED					■		
LINTRCV_30_TLE7259_START_SEC_CODE LINTRCV_30_TLE7259_STOP_SEC_CODE						■	

Table 4-3 Compiler abstraction and memory mapping

4.3 Critical Sections

The MICROSAR LIN Transceiver Driver has no code sections which are executed on interrupt level. But if the functions of the LIN Transceiver Driver can be interrupted by each other due to different call context in a preemptive operating system, protection of certain areas must be granted.

LINTRCV_EXCLUSIVE_AREA_0

- > Must lock interrupts if API functions of the LINTRCV can interrupt each other (i.e. preemptive task system, where a call of `LinTrcv_30_Tle7259_CheckWakeup()` can interrupt call of `LinTrcv_30_Tle7259_SetWakeupMode()`).

To ensure data consistency and a correct function of the LINTRCV the code section must not be interrupted. Therefore the integrator must provide the following exclusive areas in the schedule manager.

It is recommended to use AUTOSAR OS 'Resources' for these exclusive areas to prevent priority inversion and dead-locks.

4.4 Dependency to DIO Driver

The MICROSAR LIN Transceiver Driver performs hardware access by calling service functions of the lower layer component DIO Driver:

- > Function `Dio_WriteChannel()` is used to set the logical level of the channel pins to which the LIN transceiver hardware is connected.
- > Function `Dio_ReadChannel()` is used to get the logical level of the channel pins to which the LIN transceiver hardware is connected.
- > The DIO driver has to provide the pin assignment for the LIN transceiver hardware pins, which depend on the used LIN transceiver. These pins are referred by the LIN Transceiver Driver by using symbolic names or values, which must be specified in the configuration tool.

1.1 Dependency to ICU Driver

The LIN transceiver driver performs hardware access by calling service functions of the lower layer component Icu driver:

- > Function `Icu_DisableNotification()` is used by the transceiver driver to disable the ICU notification after wakeup event notification.
- > Function `Icu_EnableNotification()` is used by the transceiver driver to enable the ICU notification during the transition into the sleep mode.

The proposal described in this chapter enables the user to support the use-case by using an additional µC I/O port to generate the wakeup information via the ICU driver. This I/O port is connected parallel to the LIN Rx port.

The user has to configure the ICU channel. Additionally the wakeup source for the LIN channel which shall be handled via this ICU channel have to be chosen.

Depending on the ICU configuration options provided it is possible to configure the "IcuDefaultStartEdge". This option should be configured to `ICU_FALLING_EDGE`, because the wakeup event is provided by the Rx pin via a level change from recessive to dominant.

If the transceiver shall also wake up the ECU and not only the LIN channel then it is necessary to activate the "Wakeup Capability" for this ICU channel.

5 API Description

For an interfaces overview please see Figure 2-3.

5.1 Type Definitions

The types defined by the LINTRCV are described in this chapter.

Type Name	C-Type	Description	Value Range
LinTrcv_TrcvModeType	enum	Operating modes of the LIN Transceiver Driver	LINTRCV_TRCV_MODE_NORMAL Transceiver mode NORMAL
			LINTRCV_TRCV_MODE_STANDBY Transceiver mode STANDBY
			LINTRCV_TRCV_MODE_SLEEP Transceiver mode SLEEP
LinTrcv_TrcvWakeupModeType	enum	Wake up operating modes of the LIN Transceiver Driver	LINTRCV_WUMODE_ENABLE The notification for wakeup events is enabled on the addressed network.
			LINTRCV_WUMODE_DISABLE The notification for wakeup events is disabled on the addressed network
			LINTRCV_WUMODE_CLEAR A stored wakeup event is cleared on the addressed network
LinTrcv_TrcvWakeupReasonType	enum	This type denotes the wake up reason detected by the LIN transceiver in detail	LINTRCV_WU_ERROR Due to an error wake up reason was not detected.
			LINTRCV_WU_NOT_SUPPORTED The transceiver does not support any information for the wake up reason
			LINTRCV_WU_BY_BUS The transceiver has detected, that the network has caused the wake up of the ECU
			LINTRCV_WU_BY_PIN The transceiver has detected a wake-up event at one of the transceiver's pins (not at the LIN bus)
			LINTRCV_WU_INTERNALLY The transceiver has detected, that the network has been woken up by the ECU via a request to NORMAL mode
			LINTRCV_WU_RESET The transceiver has detected, that

Type Name	C-Type	Description	Value Range
			the wake up is due to an ECU reset
			LINTRCV_WU_POWER_ON
			The transceiver has detected, that the wake up is due to an ECU reset after power on

Table 5-1 Type definitions

5.2 Services provided by LINTRCV

The LINTRCV API consists of services, which are realized by function calls.

5.2.1 LinTrcv_30_Tle7259_InitMemory

Prototype	
Void LinTrcv_30_Tle7259_InitMemory (void)	
Parameter	
-	-
Return code	
-	-
Functional Description	
<p>> This function initializes all initialized variables of the LIN Transceiver Driver. This function has to be called after power on or reset before any other function only in case initialized variables are not set after power on or reset (i.e. by the startup code).</p>	
Particularities and Limitations	
<p>> This function must be called before any other functionality of the LIN Transceiver Driver.</p>	
Expected Caller Context	
<p>> This function must be called from task level and is not reentrant.</p>	

Table 5-2 LinTrcv_30_Tle7259_InitMemory

5.2.2 LinTrcv_30_Tle7259_Init

Prototype	
void LinTrcv_30_Tle7259_Init (void)	
Parameter	
-	-

Return code	
-	-
Functional Description	
<p>> This function initializes all channels of the LIN Transceiver Driver which are configured in the configuration tool.</p>	
Particularities and Limitations	
<p>> This function must be called before any other service functionality of the LIN Transceiver Driver</p>	
Expected Caller Context	
<p>> This function must be called from task level and is not reentrant.</p>	

Table 5-3 LinTrcv_30_Tle7259_Init

5.2.3 LinTrcv_30_Tle7259_SetOpMode

Prototype	
<pre>Std_ReturnType LinTrcv_30_Tle7259_SetOpMode (uint8 LinTrcvIndex, LinTrcv_TrvcModeType OpMode)</pre>	
Parameter	
LinTrcvIndex	Index of the selected transceiver
OpMode	Pointer which contains the desired operation mode
Return code	
E_OK / E_NOT_OK	<p>E_OK: will be returned if the transceiver state has been changed to the requested mode.</p> <p>E_NOT_OK: will be returned if the transceiver state change has failed or the parameter is out of the allowed range. The previous state has not been changed.</p>
Functional Description	
<p>Sets the LIN transceiver to the requested operation mode. These operation modes are:</p> <p>> LINTRCV_30_TLE7259_OP_MODE_NORMAL</p> <p>> LINTRCV_30_TLE7259_OP_MODE_SLEEP</p>	
Particularities and Limitations	
<p>> The LIN Transceiver Driver must be initialized.</p> <p>> Not each transition from one mode in any other is allowed.</p>	
Expected Caller Context	
<p>> This function can be called from task or interrupt level and is not reentrant.</p>	

Table 5-4 LinTrcv_30_Tle7259_SetOpMode

5.2.4 LinTrcv_30_Tle7259_GetOpMode

Prototype	
Std_ReturnType LinTrcv_30_Tle7259_GetOpMode (uint8 LinTrcvIndex, LinTrcv_TrcvModeType *OpMode)	
Parameter	
LinTrcvIndex	Index of the selected transceiver
OpMode	Pointer to operation mode of the bus the API is applied to
Return code	
E_OK / E_NOT_OK	E_OK: will be returned if the operation mode was detected. E_NOT_OK: will be returned if the operation mode was not detected.
Functional Description	
Returns the operation mode of the selected LIN transceiver. These operation modes are:	
> LINTRCV_30_TLE7259_OP_MODE_NORMAL	
> LINTRCV_30_TLE7259_OP_MODE_SLEEP	
Particularities and Limitations	
> The LIN Transceiver Driver must be initialized.	
Expected Caller Context	
> This function can be called from task or interrupt level and is not reentrant.	

Table 5-5 LinTrcv_30_Tle7259_GetOpMode

5.2.5 LinTrcv_30_Tle7259_GetBusWuReason

Prototype	
Std_ReturnType LinTrcv_30_Tle7259_GetBusWuReason (uint8 LinTrcvIndex, LinTrcv_TrcvModeType *Reason)	
Parameter	
LinTrcvIndex	Index of the selected transceiver.
Reason	Pointer to wake up reason of the bus the API is applied to.
Return code	
E_OK / E_NOT_OK	E_OK: will be returned if the wake up reason was detected. E_NOT_OK: will be returned if the wake up reason was not detected.

Functional Description
<p>Returns the wakeup reason for the channel LinTrcvIndex. These wakeup reasons are:</p> <ul style="list-style-type: none"> > LINTRCV_WU_ERROR: Due to an error wake up reason was not detected. > LINTRCV_WU_NOT_SUPPORTED: The transceiver does not support any information for the wake up reason. > LINTRCV_WU_BY_BUS: The transceiver has detected that the network has caused the wake up of the ECU. > LINTRCV_WU_BY_PIN: The transceiver has detected a wake-up event at one of the transceiver's pins (not at the LIN bus). > LINTRCV_WU_INTERNALLY: The transceiver has detected that the network has been woken up by the ECU via a request to NORMAL mode. > LINTRCV_WU_RESET: The transceiver has detected, that the wake up is due to an ECU reset. > LINTRCV_WU_POWER_ON: The transceiver has detected, that the wake up is due to an ECU reset after power on.
Particularities and Limitations
<ul style="list-style-type: none"> > The LIN Transceiver Driver must be initialized.
Expected Caller Context
<ul style="list-style-type: none"> > This function can be called from task or interrupt level and is not reentrant.

Table 5-6 LinTrcv_30_Tle7259_GetBusWuReason

5.2.6 LinTrcv_30_Tle7259_SetWakeupMode

Prototype	
Ret_code LinTrcv_30_Tle7259_SetWakeupMode (uint8 LinTrcvIndex, LinTrcv_TrcvModeType TrcvWakeupMode)	
Parameter	
LinTrcvIndex	Index of the selected transceiver
TrcvWakeupMode	Requested transceiver wakeup reason
Return code	
E_OK / E_NOT_OK	E_OK: Will be returned, if the wakeup state has been changed to the requested mode. E_NOT_OK: Will be returned, if the wakeup state change has failed or the parameter is out of the allowed range. The previous state has not been changed.
Functional Description	
<p>Enables and disables the reporting from wakeup events on the channel LinTrcvIndex. To clear wake-up events on the channel LinTrcvIndex is also provided.</p> <ul style="list-style-type: none">> LINIF_TRCV_WU_ENABLE: Report wakeup events to upper layer> LINIF_TRCV_WU_DISABLE: Do not report wakeup events to upper layer> LINIF_TRCV_WU_CLEAR: Clear a pending wakeup event (Only when the wakeup mode is disabled)	

Particularities and Limitations
<ul style="list-style-type: none"> > The LIN Transceiver Driver must be initialized. > If the wakeup handling is not enabled in GENy the function always return E_NOT_OK.
Expected Caller Context
<ul style="list-style-type: none"> > This function can be called from task or interrupt level and is not reentrant.

Table 5-7 LinTrcv_30_Tle7259_SetWakeupMode

5.2.7 LinTrcv_30_Tle7259_CheckWakeup

Prototype	
Std_ReturnType LinTrcv_30_Tle7259_CheckWakeup (uint8 LinTrcvIndex)	
Parameter	
LinTrcvIndex	Index of the selected transceiver
Return code	
E_OK /E_NOT_OK	E_OK: when a wakeup is detected. E_NOT_OK: when no wakeup is detected or wakeup by bus is not supported.
Functional Description	
This function returns if a wakeup was detected on the bus. It can be used in interrupt and polling mode.	
Particularities and Limitations	
> The LIN Transceiver Driver must be initialized.	
Expected Caller Context	
> This function can be called from task or interrupt level and is not reentrant.	

Table 5-8 LinTrcv_30_Tle7259_CheckWakeup

5.2.8 LinTrcv_30_Tle7259_GetVersionInfo

Prototype	
void LinTrcv_30_Tle7259_GetVersionInfo (Std_VersionInfoType VersionInfo)	
Parameter	
VersionInfo	Pointer to version information of this module.
Return code	
-	-
Functional Description	
Gets the version of the module and returns it in VersionInfo.	
Particularities and Limitations	
> -	
Expected Caller Context	

> This function can always be called.

Table 5-9 LinTrcv_30_Tle7259_GetVersionInfo

5.3 Services used by LINTRCV

In the following table services provided by other components, which are used by the LINTRCV are listed. For details about prototype and functionality refer to the documentation of the providing component.

Component	API
DET	Det_ReportError()
Dio	Dio_WriteChannel() Dio_ReadChannel()
EcuM	EcuM_SetWakeupEvent() EcuM_GeneratorCompatibilityError()
LinIf	LinIf_WakeupConfirmation()
SchM	SchM_Enter_LinTrcv_30_Tle7259_LINTRCV_30_TLE7259_EXCLUSIVE_AREA_0() SchM_Exit_LinTrcv_30_Tle7259_LINTRCV_30_TLE7259_EXCLUSIVE_AREA_0()

Table 5-10 Services used by the LINTRCV

5.4 Callback Functions

5.4.1 Appl_LinTrcv_30_Tle7259_Wait

Prototype	
void Appl_LinTrcv_30_Tle7259_Wait (uint32 LinTrcvWaitCount)	
Parameter	
LinTrcvWaitCount	Pointer to operation mode of the bus the API is applied to
Return code	
-	-
Functional Description	
> Allows the user to implement a delay after each mode transition	
Particularities and Limitations	
> The LIN Transceiver Driver must be initialized.	
Expected Caller Context	
> This function is called from the LinTrcv_30_Tle7259_Init and LinTrcv_30_Tle7259_SetOpMode functions	

Table 5-11 Appl_LinTrcv_30_Tle7259_Wait

5.5 Configurable Interfaces

There are no configurable interfaces for LIN transceiver driver.

6 Configuration

In the LINTRCV the attributes can be configured according to/ with the following methods/ tools:

- > Configuration in DaVinci Configurator 5, for a detailed description see 6.2

6.1 Configuration Variants

The LINTRCV supports the configuration variants

- > VARIANT-PRE-COMPILE
- > VARIANT-POSTBUILD-SELECTABLE

The configuration classes of the LINTRCV parameters depend on the supported configuration variants.

For their definitions please see the LinTrcv_30_Tle7259_bswmd.arxml file.

6.2 Configuration with a GCE

Container Name	LinTrcv
Path	\MICROSAR\LinTrcv
Multiplicity	0...1
Description	Configuration of the LinTrcv (LIN Transeiver) module.

Table 6-1 Container LinTrcv

6.2.1 LinTrcvGeneral

Container Name	LinTrcvGeneral
Path	\MICROSAR\LinTrcv
Multiplicity	1...1
Description	

Table 6-2 Container LinTrcvGeneral

Attribute Name	Multi- plicity	Value Type	Values <small>Default value is typed bold</small>	Description
Dev Error Detect	1...1	Boolean	true false	Switches development error detection and notification on and off. If switched on, #define LINTRCV_DEV_ERROR_DETECT ON shall be generated. If switched off, #define LINTRCV_DEV_ERROR_DETECT OFF shall be generated. Define shall be part of file

Attribute Name	Multi- plicity	Value Type	Values <small>Default value is typed bold</small>	Description
				LinTrcv_Cfg.h.
Get Version Info	1...1	Boolean	true false	If enabled the service function LinTrcv_30_Tle7259_GetVersionInfo() can be used to get version information of the LIN Transceiver Driver module.
Index	1...1	Integer	0 - 255	Specifies the Instance Id of this module instance. If only one instance is present it shall have the Id 0.
SafeBSWChecks	1...1	Boolean	true false	Enabled/Disables the safe checks for the component.
User Config File	0...1	String	empty	The content of this specified user file is added to the file LinTrcv_30_Tle7259_Cfg.h.
Wait Count	1...1	Integer	0 - 4294967295	Wait count for transceiver state changes.
Wake Up Support	1...1	Boolean	true false	Informs whether wake up is supported or not. In case wake up is not supported by LIN transceiver hardware the setting shall be false. The wake up ability may be switched on or off for each channel of one LIN transceiver by LinTrcvWakeupSourceRef.

Table 6-3 Attributes of LinTrcvGeneral

6.2.2 LinTrcvChannel

Container Name	LinTrcvChannel
Path	\MICROSAR\LinTrcv
Multiplicity	1...*
Description	

Table 6-4 Container LinTrcvChannel

Attribute Name	Multi- plicity	Value Type	Values <small>Default value is typed bold</small>	Description
LinTrcvChannelId	1...1	Integer	0 - 255	Unique identifier of the LIN Transceiver Channel.
LinTrcvChannelUsed	1...1	Boolean	true false	Shall the related LIN transceiver channel be used?
LinTrcvInitState	1...1	String	LINTRCV_TRCV_ MODE_NORMAL LINTRCV_TRCV_ MODE_STANDBY LINTRCV_TRCV_	State of LIN transceiver after call to LinTrcv_Init.

Attribute Name	Multi- plicity	Value Type	Values <small>Default value is typed bold</small>	Description
			MODE_SLEEP	
LinTrcvWakeupByBus Used	1...1	Boolean	true false	Is wake up by bus supported? If LIN transceiver hardware does not support wake up by bus value is always FALSE. If LIN transceiver hardware supports wake up by bus value is TRUE or FALSE depending whether it is used or not.
LinTrcvWakeupByPin Used	1...1	Boolean	true false	Is wake up by pin supported? If LIN transceiver hardware does not support wake up by pin value is always FALSE. If LIN transceiver hardware supports wake up by pin value is TRUE or FALSE depending whether it is used or not.
LinTrcvSleepInitWork around	1...1	Boolean	true false	Some transceivers cannot be initialized in sleep mode after power on. When enabled and if the channel init state is set to sleep, the driver will set the transceiver to normal mode then to sleep mode. The TX pin level must remain high during the operation to avoid dominant spikes on the LIN bus.
LinTrcvIcuChannelRef	0...1	-	-	Reference to the IcuChannel to enable/disable the interrupts for wakeups.
LinTrcvWakeupSourceRef	0...1	-	-	Reference to a wakeup source in the EcuM configuration. This reference is only needed if LinTrcvWakeupByBusUsed is true. Implementation Type: reference to EcuM_WakeupSourceType.

Table 6-5 Attributes of LinTrcvChannel

Sub Container	Multiplicity
LinTrcvAccess	1...*

Table 6-6 Sub Containers of LinTrcvChannel

Sub Container	Multiplicity
LinTrcvDioAccess	0...1
LinTrcvSpiSequence	0...1

Table 6-7 Sub Containers of LinTrcvAccess

6.2.3 LinTrcvDioAccess

Container Name	LinTrcvDioAccess
Path	\\MICROSAR\\LinTrcv\\LinTrcvChannel\\LinTrcvAccess
Multiplicity	0...1
Description	Container gives LIN transceiver driver information about accessing ports and port pins. In addition relation between LIN transceiver hardware pin names and Dio port access information is given. If a LIN transceiver hardware has no Dio interface, there is no instance of this container

Table 6-8 Container LinTrcvDioAccess

Attribute Name	Multiplicity	Value Type	Values Default value is typed bold	Description
LinTrcvHardwareInterfaceName	1...1	String	-	LIN transceiver hardware interface name. It is typically the name of a pin. From a Dio point of view it is either a port, a single channel or a channel group. Depending on this fact either LINTRCV_DIO_PORT_SYMBOLIC_NAME or LINTRCV_DIO_CHANNEL_SYMBOLIC_NAME or LINTRCV_DIO_CHANNEL_GROUP_SYMBOLIC_NAME shall reference a Dio configuration. The LIN transceiver driver implementation description shall list up this name for the appropriate LIN transceiver hardware.
LinTrcvDioSymRefName	1...1	-	-	Choice Reference to a DIO Port, DIO Channel or DIO Channel Group. This reference replaces the LINTRCV_DIO_PORT_SYM_NAME, LINTRCV_DIO_CHANNEL_SYM_NAME and LINTRCV_DIO_GROUP_SYM_NAME references in the Lin Trcv SWS.

Table 6-9 Attributes of LinTrcvDioAccess

6.2.4 LinTrcvSpiSequence

Container Name	LinTrcvSpiSequence
Path	\\MICROSAR\\LinTrcv\\LinTrcvChannel\\LinTrcvAccess
Multiplicity	0...1
Description	Container gives LIN transceiver driver information about one SPI sequence

Table 6-10 Container LinTrcvSpiSequence

Attribute Name	Multi- plicity	Value Type	Values <small>Default value is typed bold</small>	Description
LinTrcvSpiSequence	1...1	-	-	Reference to a Spi sequence configuration container.

Table 6-11 Attributes of LinTrcvSpiSequence

7 Glossary and Abbreviations

7.1 Glossary

Term	Description
CFG5	DaVinci Configurator 5, generation tool for MICROSAR components

Table 7-1 Glossary

7.2 Abbreviations

Abbreviation	Description
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
BSW	Basis Software
DEM	Diagnostic Event Manager
DET	Development Error Tracer
EAD	Embedded Architecture Designer
ECU	Electronic Control Unit
HIS	Hersteller Initiative Software
ISR	Interrupt Service Routine
MICROSAR	Microcontroller Open System Architecture (the Vector AUTOSAR solution)
PPort	Provide Port
RPort	Require Port
RTE	Runtime Environment
SRS	Software Requirement Specification
SWC	Software Component
SWS	Software Specification

Table 7-2 Abbreviations

8 Contact

Visit our website for more information on

- > News
- > Products
- > Demo software
- > Support
- > Training data
- > Addresses

www.vector.com