

MICROSAR LIN Driver

Technical Reference

Tricore

Version 6.4.0

Authors	Lutz Pflüger, Andreas Pick
Status	Released

Document Information

History

Author	Date	Version	Remarks
Lutz Pflüger	2013-08-30	4.0.0	Initial release
Lutz Pflüger	2013-11-13	5.0.0	R8, change title, add application lock/unlock API
Lutz Pflüger	2014-12-02	6.0.0	MICROSAR AR4-R11, remove duplicated Chapter 4
Lutz Pflüger	2015-01-14	6.1.0	Rename Add hardware deviation on break field Change CLC register API Add Lin_GetRxPin() description
Lutz Pflüger	2016-01-28	6.2.0	Remove Lin_GetRxPin() description Update TechnicalReference_Lin_Core.pdf version Remove SWS reference document of Lin Remove Chapter 1, 3, 5.2, 5.3, 6.1 Move derivate/compiler info to Chapter 1 Change Category 1 interrupts deviations Cleanup Abbreviations Change Scope of the Document
Lutz Pflüger	2016-04-08	6.3.0	Remove Chapter 2.1 Update TechnicalReference_Lin_Core.pdf version
Andreas Pick	2018-04-03	6.04.00	Hardware Software Interface updated

Reference Documents

No.	Source	Title	Version
[1]	Vector	TechnicalReference_Lin_Core.pdf	see delivery

Scope of the Document

This technical reference describes the hardware specific use of the LIN driver software. It supplements the general LIN driver technical reference [1]

**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.

Contents

1	Hardware Overview	5
1.1	Deviations of Platform	6
2	API Description	7
2.1	Services used by LIN	7
2.1.1	Appl_UnlockEndinit	7
2.1.2	Appl_LockEndinit	7
2.1.3	osWritePeripheral32	8
3	Abbreviations.....	9
4	Contact.....	10

Tables

Table 1-1	Hardware interface structures for memory-mapped i/o	5
Table 2-1	Appl_UnlockEndinit.....	7
Table 2-2	Appl_LockEndinit.....	7
Table 2-3	osWritePeripheral32	8
Table 3-1	Abbreviations.....	9

1 Hardware Overview

The following table summarizes information about the LIN Driver. It gives you detailed information about the derivatives and compilers. As very important information the documentations of the hardware manufacturers are listed. The LIN Driver is based upon these documents in the given version.

Derivative: This can be a single information or a list of derivatives, the LIN Driver can be used on.

Compiler: List of Compilers the LIN Driver is working with

Hardware Manufacturer Document Name: List of hardware documentation the LIN Driver is based on.

Version: To be able to reference to this hardware documentation its version is very important.

Derivative	TC2xx
Compiler	GNU, Tasking
Hardware manual	tc27x_ts_V3.0_OPEN_MARKET.pdf
Version	See PDF File Name
Chapter	1.19 Asynchronous/Synchronous Interface (ASCLIN)
Errata sheet	TC27x_BC_Errata_Sheet_rev1v5_2015_09_16.pdf TC26x Step: ES-BC, BC Rel. 1.0, 2016-12-22 TC22x/TC21x Step: ES-AB, AB, Rel. 1.4, 2017-10-10 ASCLIN_TC.H001, ASCLIN_TC.H002, ASCLIN_TC.004, ASCLIN_TC.005, ASCLIN_TC.006, ASCLIN_TC.007, ASCLIN_TC.009, are not relevant
Safety manuals	AURIX Safety Manual AP32224 V1.5 2017-05
Specifics	End-init protection is used as specified in the HW manual and the MSR architecture design.
Used registers and bits (offset)	0x00 to 0x4C
Operating modes	refer to the states specified in the CAD note: parts of the state machine are implemented in HW
Hardware features related to independence or partitioning	none
Access mechanism	memory mapped i/o registers
Hardware diagnostics	none

Table 1-1 Hardware interface structures for memory-mapped i/o

1.1 Deviations of Platform

**Caution**

The application must implement the `Appl_UnlockEndinit()` and `Appl_LockEndinit()` callback functions.

**Caution**

The `os.h` header file must be provided.

**Caution**

If the ENDINIT protected registers could not write in user mode the `osWritePeripheral32()` must be provided.

**Caution**

Interrupt 'Category 1' is not supported. If selected Interrupt Function is defined as `void Func(void)` function.

**Caution**

The hardware cannot detect a disturbance on the break field.

**Caution**

On some Infineon MCAL no valid `[ANY]/Mcu/McuModuleConfiguration/McuClockSettingConfig/McuClockReferencePoint` exist. In this case the Clock for the Lin Channel can be configured using the parameter `/MICROSAR/Lin_Tricore/Lin/LinGlobalConfig/LinChannel/LinClockFrequency`.

**Caution**

This hardware platform doesn't report the `LIN_E_TIMEOUT` error.

2 API Description

2.1 Services used by LIN

2.1.1 Appl_UnlockEndinit

Prototype	
void Appl_UnlockEndinit (void)	
Parameter	
-	-
Return code	
-	-
Functional Description	
Routine to unlock registers that are normally protected through Endinit-Protect.	
Particularities and Limitations	
> Interrupts must be disabled.	
Call context	
> Called by LIN driver.	

Table 2-1 Appl_UnlockEndinit

2.1.2 Appl_LockEndinit

Prototype	
void Appl_LockEndinit (void)	
Parameter	
-	-
Return code	
-	-
Functional Description	
Routine to lock registers that are normally protected through Endinit-Protect.	
Particularities and Limitations	
> Interrupts must be disabled.	
Call context	
> Called by LIN driver.	

Table 2-2 Appl_LockEndinit

2.1.3 osWritePeripheral32

Prototype	
osWritePeripheral32 (LIN_PROTECTED_AREA_ENDINIT, uint32 addr, uint32 value)	
Parameter	
LIN_PROTECTED_AREA_ENDINIT	Value and Type of LIN_PROTECTED_AREA_ENDINIT
addr	Hardware register address
value	Value to write on hardware register
Return code	
-	Unused on Lin Driver
Functional Description	
Writing a value to a protected hardware register into the protected area given by LIN_PROTECTED_AREA_ENDINIT parameter. The Lin protected area must include all CLC register of the used Lin Channels. The function prototype and defines must be added to "os.h" or the "user config file" of Lin Driver.	
Particularities and Limitations	
<ul style="list-style-type: none"> > If osWritePeripheral32 not defined the CLC register are written from Lin Driver direct > If osWritePeripheral32 defined LIN_PROTECTED_AREA_ENDINIT must be defined too > Appl_UnlockEndinit are called before call of osWritePeripheral32/write to CLC register > Appl_LockEndinit are called after return of osWritePeripheral32/write to CLC register 	
Call context	
<ul style="list-style-type: none"> > Called by LIN driver. 	

Table 2-3 osWritePeripheral32

3 Abbreviations

Abbreviation	Description
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
ISR	Interrupt Service Routine
MICROSAR	Microcontroller Open System Architecture (the Vector AUTOSAR solution)

Table 3-1 Abbreviations

4 Contact

Visit our website for more information on

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

www.vector.com