User Manual

for MPC5634M LIN Driver

Document Number: UM14LINASR3.0R2.0.0
Rev. 2.0

freescale™

Contents

Section Number **Title** **Page**

Chapter 1
Revision History

Chapter 2
Introduction

Chapter 3
Driver

Chapter 4
Tresos Configuration Plug-in

Chapter 1
Revision History

Table 1-1. Revision History

Revision	Date	Author	Description
1.0	07-Feb-2011	Giuseppe Stefano Fazio	Document generation
2.0	14-Dec-2011	Anuj Gupta	Updated for Monaco RTM 2.0.0

Chapter 2
Introduction

This User Manual describes Freescale Semiconductor AUTOSAR Local Interconnect Network (LIN) for MPC5634M .

AUTOSAR LIN driver configuration parameters and deviations from the specification are described in LIN Driver chapter of this document. AUTOSAR LIN driver requirements and APIs are described in the AUTOSAR LIN driver software specification document.

2.1 Supported Derivatives

The software described in this document is intented to be used with the following microcontroller devices of Freescale Semiconductor .

Table 2-1. MPC5634M Derivatives

Freescale Semiconductor	mpc5634m_bga208, mpc5634m_qfp144, mpc5634m_qfp176

All of the above microcontroller devices are collectively named as MPC5634M .

2.2 Overview

AUTOSAR (AUTomotive Open System ARchitecture) is an industry partnership working to establish standards for software interfaces and software modules for automobile electronic control systems.

AUTOSAR

- paves the way for innovative electronic systems that further improve performance, safety and environmental friendliness.

- is a strong global partnership that creates one common standard: "Cooperate on standards, compete on implementation".
- is a key enabling technology to manage the growing electrics/electronics complexity. It aims to be prepared for the upcoming technologies and to improve cost-efficiency without making any compromise with respect to quality.
- facilitates the exchange and update of software and hardware over the service life of the vehicle.

2.3 About this Manual

This Technical Reference employs the following typographical conventions:

Boldface type: Bold is used for important terms, notes and warnings.

Italic font: Italic typeface is used for code snippets in the text. Note that C language modifiers such "const" or "volatile" are sometimes omitted to improve readability of the presented code.

Notes and warnings are shown as below:

Note

This is a note.

2.4 Acronyms and Definitions

Table 2-2. Acronyms and Definitions

Term	Definition
API	Application Programming Interface
ASM	Assembler
AUTOSAR	Automotive Open System Architecture
BSMI	Basic Software Make file Interface
C/CPP	C and C++ Source Code
DEM	Diagnostic Event Manager
DET	Development Error Tracer
EcuM	ECU state Manager
GUI	Graphical User Interface
ISR	Interrupt Service Routine
LIN	Local Interconnect Network

Table continues on the next page...

Table 2-2. Acronyms and Definitions (continued)

Term	Definition
MCU	Micro Controller Unit
N/A	Not Applicable
OS	Operating System
VLE	Variable Length Encoding

2.5 Reference List

Table 2-3. Reference List

#	Title	Version
1	AUTOSAR 3.0LIN Driver Software Specification Document.	V2.2.0 R3.0 Rev 0001
2	MPC5634M Reference Manual	Rev. 6, 4 October 2011

Chapter 3
Driver

3.1 Requirements

Requirements for this driver are detailed in the AUTOSAR 3.0LIN Driver Software Specification document (See Table Reference List).

3.2 Driver Design Summary

The LIN driver is part of the microcontroller abstraction layer (MCAL), performs the hardware access and offers a hardware independent API to the upper layer.

The only upper layer, which has access to the LIN driver, is the LIN Interface.

A LIN driver can support more than one channel.

This means that the LIN driver can handle one or more LIN channels as long as they belong to the same LIN hardware unit.

The LIN Driver for MPC5634M, uses the eSCI on-chip hardware module which provides special support for the LIN protocol.

It can be used to automate most tasks of a LIN master.

It is possible to transmit entire frames (or sequences of frames) and receive data from LIN slaves without any CPU intervention.

The LIN physical interface should be connected to the eSCI module pins in order to get the LIN bus voltage levels.

The MPC5634M contains up to two blocks.

The eSCI has the following major features:

- LIN Master Node functionality (master and slave task)
- Compatible with LIN slaves from revisions 1.x and 2.0 of the LIN standard
- Detection of Bit Errors, Physical Bus Errors and Checksum Errors
- All status bit can generate maskable interrupts
- Application layer CRC support
- Programmable CRC polynom
- Double Stop Flag insertion after Bit Errors
- Detection and generation of wakeup characters
- Programmable wakeup delimiter time
- Can be configured to include header bits in checksum
- LIN DMA interface

3.3 Deviation from Requirements

Not Applicable.

3.4 Runtime Errors

The driver generates the following DEM errors at runtime.

Table 3-1. Runtime Errors

Function	Error Code	Condition triggering the error
Lin_GoToSleep()	LIN_E_TIMEOUT	Timeout caused by hardware error waiting for cancellation of current frame.
Lin_GoToSleepInternal()	LIN_E_TIMEOUT	Timeout caused by hardware error waiting for cancellation of current frame.
Lin_SendHeader()	LIN_E_TIMEOUT	Timeout caused by hardware error waiting for cancellation of current frame.

3.5 Software specification

The following sections contains driver software specifications.

3.5.1 Define Reference

Constants supported by the driver are as per AUTOSAR LIN Driver software specification Version 3.0 .

3.5.2 Enum Reference

Enumeration of all constants supported by the driver are as per AUTOSAR LIN Driver software specification Version 3.0 .

3.5.3 Function Reference

Functions of all functions supported by the driver are as per AUTOSAR LIN Driver software specification Version 3.0 .

3.5.3.1 Function Lin_DeInitChannel

De-Inits a LIN channel.

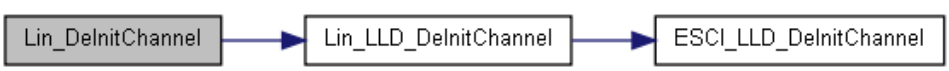

Figure 3-1. Function Lin_DeInitChannel References.

Details:

This function disables the LIN module which is assigned to Channel, clears the channel shadow buffer and update the state machines.

Note

Autosar Service ID: 0x03. Synchronous, non reentrant function.

Satisfied Requirements: LIN009, LIN086, LIN152, LIN178, LIN116.

Prototype: `void Lin_DeInitChannel(const uint8 Channel);`

Table 3-2. Lin_DeInitChannel Arguments

Type	Name	Direction	Description
const uint8	Channel	**input**	LIN channel to be de-initialized.

3.5.3.2 Function Lin_GetStatus

Gets the status of the LIN driver.

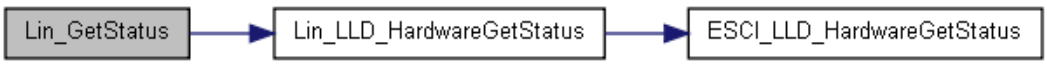

Figure 3-2. Function Lin_GetStatus References.

Figure 3-3. Inverse references of fuction Lin_GetStatus

Details:

This function returns the state of the current transmission, reception or operation status. If the reception of a Slave response was successful then this service provides a pointer to the buffer where the data is stored.

Note

Autosar Service ID: 0x08. Synchronous, non reentrant function.

Satisfied Requirements: LIN022, LIN024, LIN060, LIN087, LIN168, LIN091, LIN092, LIN141, LIN142, LIN143, LIN144.

Prototype: `Lin_StatusType Lin_GetStatus(const uint8 Channel, uint8 **Lin_SduPtr);`

Table 3-3. Lin_GetStatus Arguments

Type	Name	Direction	Description
const uint8	Channel	**input**	LIN channel to be checked.
uint8 **	Lin_SduPtr	**output**	Lin_SduPtr pointer to pointer to a shadow buffer or memory mapped LIN Hardware receive buffer where the current SDU is stored.

3.5.3.3 Function Lin_GetVersionInfo

Returns the version information of this module.

Note

Autosar Service ID: 0x01. Synchronous, non reentrant function.

Satisfied Requirements: LIN161, LIN001, LIN110, LIN111, LIN067.

Prototype: `void Lin_GetVersionInfo(Std_VersionInfoType *versioninfo);`

Table 3-4. Lin_GetVersionInfo Arguments

Type	Name	Direction	Description
Std_VersionInfoType *	versioninfo	**output**	Pointer to a Std_VersionInfoType structure. .

3.5.3.4 Function Lin_GoToSleep

The service instructs the driver to transmit a go-to-sleep-command on the addressed LIN channel.

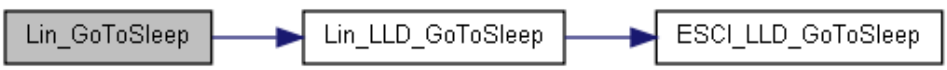

Figure 3-4. Function Lin_GoToSleep References.

Details:

This function stops any ongoing transmission and initiates the transmission of the sleep command (master command frame with ID = 0x3C and data = (0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF). State transition in LIN_CH_SLEEP_STATE shall be done after the completion of the sleep command transmission regardless of the success (therefore the ISR is responsible to put the channel in LIN_CH_SLEEP_STATE).

Note

Autosar Service ID: 0x06. Synchronous, non reentrant function.

Satisfied Requirements: LIN172, LIN032, LIN033, LIN166, LIN089, LIN073, LIN034, LIN074, LIN129, LIN130, LIN131, LIN132.

Prototype: `Std_ReturnType Lin_GoToSleep(const uint8 Channel);`

Table 3-5. Lin_GoToSleep Arguments

Type	Name	Direction	Description
const uint8	Channel	**input**	LIN channel to be addressed.

3.5.3.5 Function Lin_GoToSleepInternal

Put a Lin channel in the internal sleep state.

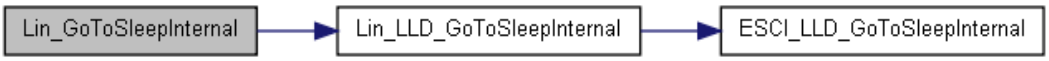

Figure 3-5. Function Lin_GoToSleepInternal References.

Details:

Stops any ongoing transmission, sets the channel state to LIN_CH_SLEEP and put the LIN hardware unit to a reduced power operation mode.

Note

Autosar Service ID: 0x09. Synchronous, non reentrant function.

Satisfied Requirements: LIN167, LIN032, LIN033, LIN095, LIN133, LIN134, LIN135, LIN136.

Prototype: `Std_ReturnType Lin_GoToSleepInternal(const uint8 Channel);`

Table 3-6. Lin_GoToSleepInternal Arguments

Type	Name	Direction	Description
const uint8	Channel	**input**	LIN channel to be addressed.

3.5.3.6 Function Lin_Init

Initializes the LIN module.

Details:

This function performs software initialization of LIN driver:

- Clears the shadow buffer of all available Lin channels
- Set LIN channel state machine of all available Lin channels to LIN_CH_UNINIT_STATE
- Set frame operation state machine of all available LIN channels to LIN_CH_NOT_READY_STATE
- Set driver state machine to LIN_INIT.

Post: Driver status = LIN_INIT, channel status = LIN_CH_UNINIT_STATE.

Note

Autosar Service ID: 0x00. Synchronous, non reentrant function.

Satisfied Requirements: LIN146, LIN171, LIN006, LIN084, LIN150, LIN008, LIN106, LIN099, LIN105.

Prototype: `void Lin_Init(const Lin_ConfigType *Config);`

Table 3-7. Lin_Init Arguments

Type	Name	Direction	Description
const Lin_ConfigType *	Config	**input**	Pointer to LIN driver configuration set.

3.5.3.7 Function Lin_InitChannel

(Re-)initializes a LIN channel.

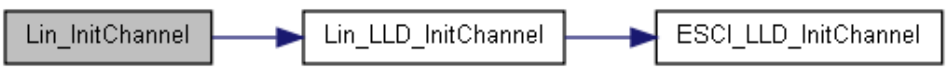

Figure 3-6. Function Lin_InitChannel References.

Note

Autosar Service ID: 0x02. Synchronous, non reentrant function.

Satisfied Requirements: LIN012, LIN147, LIN007, LIN112, LIN113, LIN100, LIN114, LIN115.

Prototype: `void Lin_InitChannel(const uint8 Channel, const Lin_ChannelConfigType *Config);`

Table 3-8. Lin_InitChannel Arguments

Type	Name	Direction	Description
const uint8	Channel	**input**	LIN channel to be initialized.
const Lin_ChannelConfigType *	Config	**input**	Pointer to LIN channel configuration set.

3.5.3.8 Function Lin_SendHeader

Sends a LIN header.

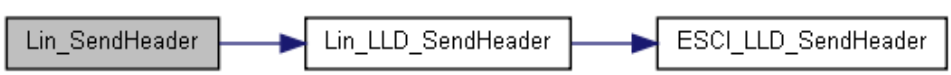

Figure 3-7. Function Lin_SendHeader References.

Note

Autosar Service ID: 0x04. Synchronous, non reentrant function.

Satisfied Requirements: LIN016, LIN017, LIN018, LIN019, LIN021, LIN164, LIN087, LIN117, LIN118, LIN119, LIN120, LIN121, LIN122.

Report Production error

Prototype: `Std_ReturnType Lin_SendHeader(const uint8 Channel, const Lin_PduType *PduInfoPtr);`

Table 3-9. Lin_SendHeader Arguments

Type	Name	Direction	Description
const uint8	Channel	**input**	LIN channel to be addressed.
const `Lin_PduType` *	PduInfoPtr	**input**	Pointer to PDU containing the PID, Checksum model, Response type, Dl and SDU data pointer.

3.5.3.9 Function Lin_SendResponse

Sends a LIN response.

Figure 3-8. Function Lin_SendResponse References.

Note

Autosar Service ID: 0x05. Synchronous, non reentrant function.

Satisfied Requirements: LIN018, LIN025, LIN026, LIN027, LIN028, LIN165, LIN088, LIN128, LIN153, LIN123, LIN124, LIN125, LIN126, LIN127.

Prototype: `Std_ReturnType Lin_SendResponse(const uint8 Channel, const Lin_PduType *PduInfoPtr);`

Table 3-10. Lin_SendResponse Arguments

Type	Name	Direction	Description
const uint8	Channel	**input**	LIN channel to be addressed.
const `Lin_PduType` *	PduInfoPtr	**input**	Pointer to PDU containing the PID, Checksum model, Response type, Dl and SDU data pointer.

3.5.3.10 Function Lin_WakeUp

Generates a wake up pulse.

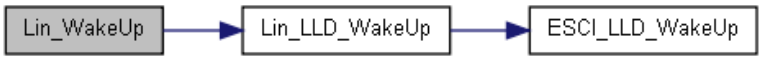

Figure 3-9. Function Lin_WakeUp References.

Details:

This function shall sent a wake up signal to the LIN bus and put the LIN channel in LIN_CH_OPERATIONAL_STATE state.

Note

Autosar Service ID: 0x07. Synchronous, non reentrant function.

Satisfied Requirements: LIN174, LIN043, LIN090, LIN169, LIN154, LIN137, LIN138, LIN139, LIN140.

Prototype: `Std_ReturnType Lin_WakeUp(const uint8 Channel);`

Table 3-11. Lin_WakeUp Arguments

Type	Name	Direction	Description
const uint8	Channel	**input**	LIN channel to be addressed.

3.5.3.11 Function Lin_WakeUpValidation

Validates for upper layers the wake up of LIN channels.

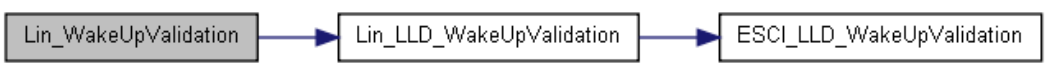

Figure 3-10. Function Lin_WakeUpValidation References.

Details:

This function identifies which LIN channel has been woken up by the LIN bus transceiver This API is used when the LIN channel wake up functionality is disabled (wake up interrupt is disabled) it checks the wake up flag from all LIN channels which are in sleep mode and which have the wake up functionality disabled.

Note

Autosar Service ID: 0x0A. Synchronous, non reentrant function.

Satisfied Requirements: LIN160, LIN098, LIN107, LIN108, LIN109.

Prototype: `void Lin_WakeUpValidation(void);`

3.5.4 Structs Reference

Data structures supported by the driver are as per AUTOSAR LIN Driver software specification Version 3.0 .

3.5.5 Types Reference

Types supported by the driver are as per AUTOSAR LIN Driver software specification Version 3.0 .

3.5.6 Variables Reference

Variables supported by the driver are as per AUTOSAR LIN Driver software specification Version 3.0 .

3.6 Symbolic Names DISCLAIMER

All containers having the symbolic name tag set as true in the Autosar schema will generate defines like:

#define <Container_Short_Name> <Container_ID>

For this reason it is forbidden to duplicate the name of such containers across the MCAL configuration, or to use names that may trigger other compile issues (e.g. match existing #ifdefs arguments).

Chapter 4
Tresos Configuration Plug-in

This chapter describes the Tresos configuration plug-in for the LIN Driver. The most of the parameters are described below.

4.1 Configuration elements of Lin

Included forms :

- IMPLEMENTATION_CONFIG_VARIANT
- CommonPublishedInformation
- NonAutosar
- LinGeneral
- LinGlobalConfig

Table 4-1. Revision table

Revision	Date
revision2.0.0	2011-12-02T17:00:00

4.2 Form IMPLEMENTATION_CONFIG_VARIANT

VariantPreCompile: Only precompile time configuration parameters.

VariantPostBuild: Mix of precompile and postbuild time configuration parameters.

If Config Variant = VariantPreCompile, the files Lin_Cfg.h and Lin_Cfg.c should be used.

If Config Variant = VariantPostBuild, the files Lin_Cfg.h, Lin_Cfg.c and Lin_PBcfg.c should be used.

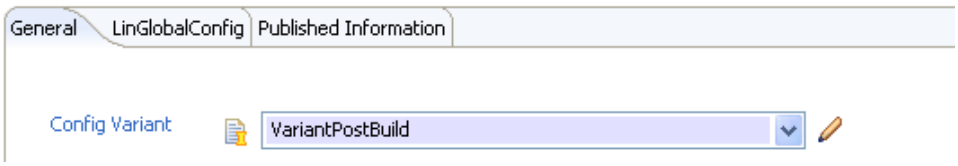

Figure 4-1. Tresos Plugin snapshot for IMPLEMENTATION_CONFIG_VARIANT form.

Table 4-2. Attribute IMPLEMENTATION_CONFIG_VARIANT detailed description

Property	Value
Label	Config Variant
Symbolic Name	false
Default	VariantPreCompile
Lower Multiplicity	1
Upper Multiplicity	1
Range	VariantPreCompile VariantPostBuild

4.3 Form CommonPublishedInformation

Common container, aggregated by all modules. It contains published information about vendor and versions.

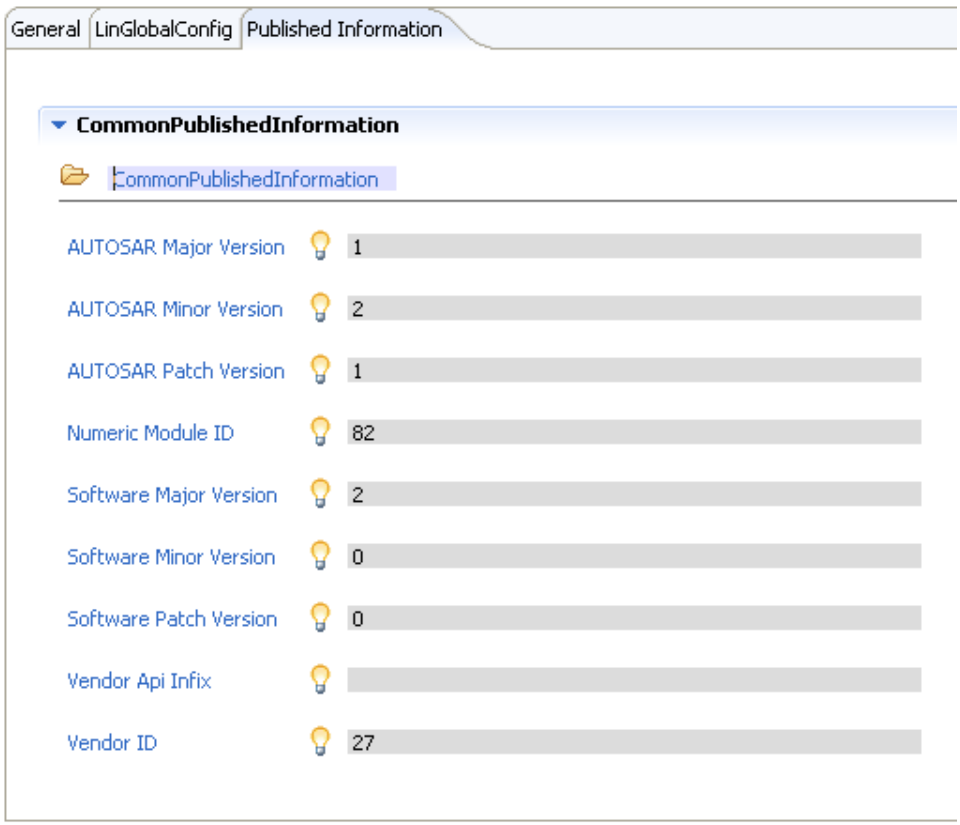

Figure 4-2. Tresos Plugin snapshot for CommonPublishedInformation form.

4.3.1 ArMajorVersion (CommonPublishedInformation)

Major version number of AUTOSAR specification on which the appropriate implementation is based on.

Table 4-3. Attribute ArMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Major Version
Origin	Custom
Symbolic Name	false
Default	1
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=1 <=1

4.3.2 ArMinorVersion (CommonPublishedInformation)

Minor version number of AUTOSAR specification on which the appropriate implementation is based on.

Table 4-4. Attribute ArMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Minor Version
Origin	Custom
Symbolic Name	false
Default	2
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=2 <=2

4.3.3 ArPatchVersion (CommonPublishedInformation)

Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.

Table 4-5. Attribute ArPatchVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Patch Version
Origin	Custom
Symbolic Name	false
Default	1
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=1 <=1

4.3.4 ModuleId (CommonPublishedInformation)

Module ID of this module from Module List.

Table 4-6. Attribute ModuleId (CommonPublishedInformation) detailed description

Property	Value
Label	Numeric Module ID
Origin	Custom
Symbolic Name	false
Default	82
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=82 <=82

4.3.5 SwMajorVersion (CommonPublishedInformation)

Major version number of the vendor specific implementation of the module. The numbering is vendor specific.

Table 4-7. Attribute SwMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Major Version
Origin	Custom
Symbolic Name	false
Default	2
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=2 <=2

4.3.6 SwMinorVersion (CommonPublishedInformation)

Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.

Table 4-8. Attribute SwMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Minor Version
Origin	Custom
Symbolic Name	false
Default	0
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=0 <=0

4.3.7 SwPatchVersion (CommonPublishedInformation)

Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.

Table 4-9. Attribute SwPatchVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Patch Version
Origin	Custom
Symbolic Name	false
Default	0
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=0 <=0

4.3.8 VendorApiInfix (CommonPublishedInformation)

In driver modules which can be instantiated several times on a single ECU, BSW00347 requires that the name of APIs is extended by the VendorId and a vendor specific name. This parameter is used to specify the vendor specific name. In total, the implementation specific name is generated as follows: <ModuleName>_>VendorId>_<VendorApiInfix><Api name from SWS>. E.g. assuming that the VendorId of the implementor is 123 and the implementer chose a

VendorApiInfix of "v11r456" a api name Can_Write defined in the SWS will translate to Can_123_v11r456Write. This parameter is mandatory for all modules with upper multiplicity > 1. It shall not be used for modules with upper multiplicity =1.

Table 4-10. Attribute VendorApiInfix (CommonPublishedInformation) detailed description

Property	Value
Label	Vendor Api Infix
Origin	Custom
Symbolic Name	false
Default	
Lower Multiplicity	1
Upper Multiplicity	1

4.3.9 VendorId (CommonPublishedInformation)

Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list.

Table 4-11. Attribute VendorId (CommonPublishedInformation) detailed description

Property	Value
Label	Vendor ID
Origin	Custom
Symbolic Name	false
Default	27
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=27 <=27

4.4 Form NonAutosar

NonAutosar

Autosar Requirements:

This container contains the global configuration parameters of the Non-Autosar Lin driver. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.

Figure 4-3. Tresos Plugin snapshot for NonAutosar form.

4.4.1 LinEnableDualClockMode (NonAutosar)

LinEnableDualClockMode

Autosar Requirements:

Switches the Dual Clock Mode ON or OFF.

Table 4-12. Attribute LinEnableDualClockMode (NonAutosar) detailed description

Property	Value
Label	Lin Enable Dual Clock Mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.5 Form LinGeneral

LinGeneral

Autosar Requirements: LIN177

This container contains the global configuration parameter of the Lin driver. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exit once per configuration set.

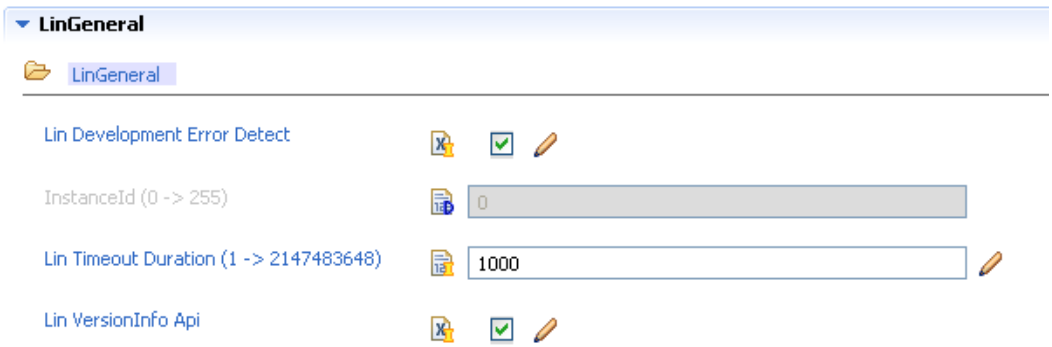

Figure 4-4. Tresos Plugin snapshot for LinGeneral form.

4.5.1 LinDevErrorDetect (LinGeneral)

LinDevErrorDetect

Autosar Requirements: LIN066

Switches the Development Error Detection and Notification ON or OFF.

Table 4-13. Attribute LinDevErrorDetect (LinGeneral) detailed description

Property	Value
Label	Lin Development Error Detect
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.5.2 LinIndex (LinGeneral)

Note

Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.

This parameter is not used in the current implementation.

Table 4-14. Attribute LinIndex (LinGeneral) detailed description

Property	Value
Label	InstanceId
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	0
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range <=255 >=0

4.5.3 LinTimeoutDuration (LinGeneral)

Specifies the maximum number of loops for blocking function until a timeout is raised in short term wait loops

Table 4-15. Attribute LinTimeoutDuration (LinGeneral) detailed description

Property	Value
Label	Lin Timeout Duration
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	1000
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=1 <=0x80000000

4.5.4 LinVersionInfoApi (LinGeneral)

LinVersionInfoApi

Autosar Requirements: LIN067

Switches the Lin_GetVersionInfo function ON or OFF.

Table 4-16. Attribute LinVersionInfoApi (LinGeneral) detailed description

Property	Value
Label	Lin VersionInfo Api
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.6 Form LinGlobalConfig

This container contains the global configuration parameter of the Lin driver. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exit once per configuration set.

Included forms :

- LinChannel

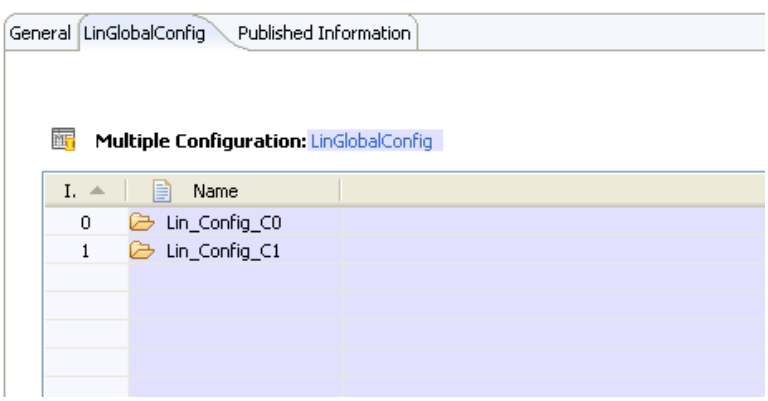

Figure 4-5. Tresos Plugin snapshot for LinGlobalConfig form.

4.6.1 Form LinChannel

Note

Identifies the LIN channel. Replaces LIN_CHANNEL_INDEX_NAME from the LIN SWS.

"User should use unique names for naming the LIN channels across different LinGlobalConfig Sets."

Is included by form : LinGlobalConfig

Included forms :

- MicroSecondChannel
- DMAsettings

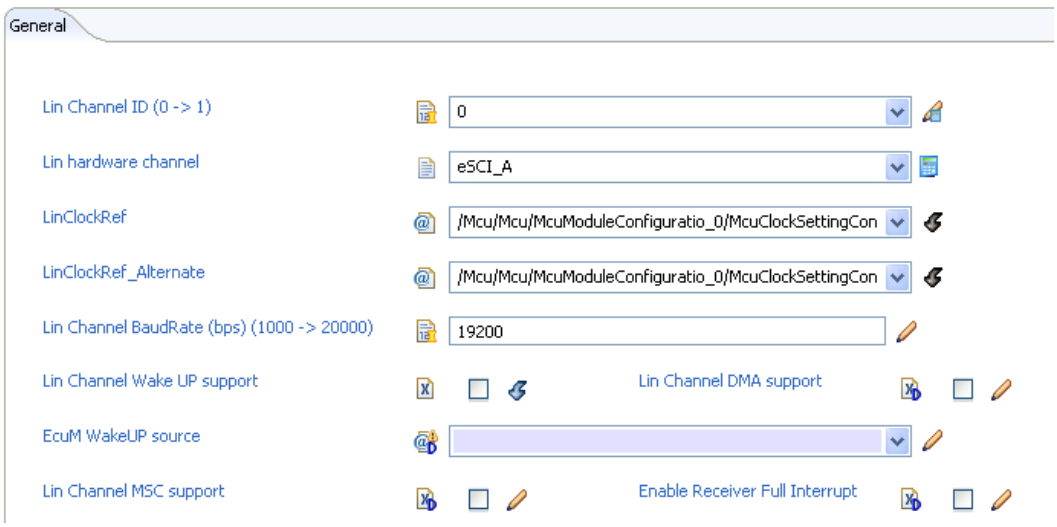

Figure 4-6. Tresos Plugin snapshot for LinChannel form.

4.6.1.1 LinChannelId (LinChannel)

Identifies the LIN channel. Replaces LIN_CHANNEL_INDEX_NAME from the LIN SWS.

Table 4-17. Attribute LinChannelId (LinChannel) detailed description

Property	Value
Label	Lin Channel ID
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	true
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=0 <=1

4.6.1.2 LinHwChannel (LinChannel)

Note

Selects the physical LIN Channel.

This Parameter is an Implementation Specific Parameter.

Table 4-18. Attribute LinHwChannel (LinChannel) detailed description

Property	Value
Label	Lin hardware channel
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Lower Multiplicity	1
Upper Multiplicity	1

4.6.1.3 LinChannelBaudRate (LinChannel)

LinChannelBaudRate

Autosar Requirements: LIN180

Specifies the baud rate of the LIN channel in 'bps'. Valid range: 1000..20000.

Table 4-19. Attribute LinChannelBaudRate (LinChannel) detailed description

Property	Value
Label	Lin Channel BaudRate (bps)
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	9600
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range <=20000 >=1000

4.6.1.4 LinChannelWakeUpSupport (LinChannel)

LinChannelWakeUpSupport

Autosar Requirements: LIN182

Specifies if the LIN hardware channel supports wake up functionality.

Table 4-20. Attribute LinChannelWakeUpSupport (LinChannel) detailed description

Property	Value
Label	Lin Channel Wake UP support
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.6.1.5 LinChannelDmaActivation (LinChannel)

LinChannelDmaActivation

Switches the DMA usage for this LIN channel ON or OFF.

Table 4-21. Attribute LinChannelDmaActivation (LinChannel) detailed description

Property	Value
Label	Lin Channel DMA support
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.6.1.6 LinChannelMSCSupport (LinChannel)

LinChannelMSCSupport

Autosar Requirements:

Specifies if the LIN hardware channel supports the MicroSecond Channel functionality.

Table 4-22. Attribute LinChannelMSCSupport (LinChannel) detailed description

Property	Value
Label	Lin Channel MSC support
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.6.1.7 ReceiverFullInterruptEnable (LinChannel)

Receiver Full Interrupt Enable

This option controls receiver full interrupt request generation. It controls **RIE** bit in **SCI_CR1** register.

Table 4-23. Attribute ReceiverFullInterruptEnable (LinChannel) detailed description

Property	Value
Label	Enable Receiver Full Interrupt
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.6.1.8 LinClockRef (LinChannel)

Table 4-24. Attribute LinClockRef (LinChannel) detailed description

Property	Value
Type	REFERENCE

Table continues on the next page...

Table 4-24. Attribute LinClockRef (LinChannel) detailed description (continued)

Property	Value
Origin	AUTOSAR_ECUC
Lower Multiplicity	1
Upper Multiplicity	1

4.6.1.9 LinClockRef_Alternate (LinChannel)

Table 4-25. Attribute LinClockRef_Alternate (LinChannel) detailed description

Property	Value
Type	REFERENCE
Origin	Custom
Lower Multiplicity	1
Upper Multiplicity	1

4.6.1.10 LinChannelEcuMWakeUpSource (LinChannel)

Table 4-26. Attribute LinChannelEcuMWakeUpSource (LinChannel) detailed description

Property	Value
Label	EcuM WakeUP source
Type	SYMBOLIC-NAME-REFERENCE
Origin	AUTOSAR_ECUC
Lower Multiplicity	1
Upper Multiplicity	1

4.6.1.11 Form MicroSecondChannel

MicroSecond Channel

Autosar Requirements:

This container include the needed parameters to configure the eSCI hardware channel to work with MicroSecond Channel functionality. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.

Is included by form : LinChannel

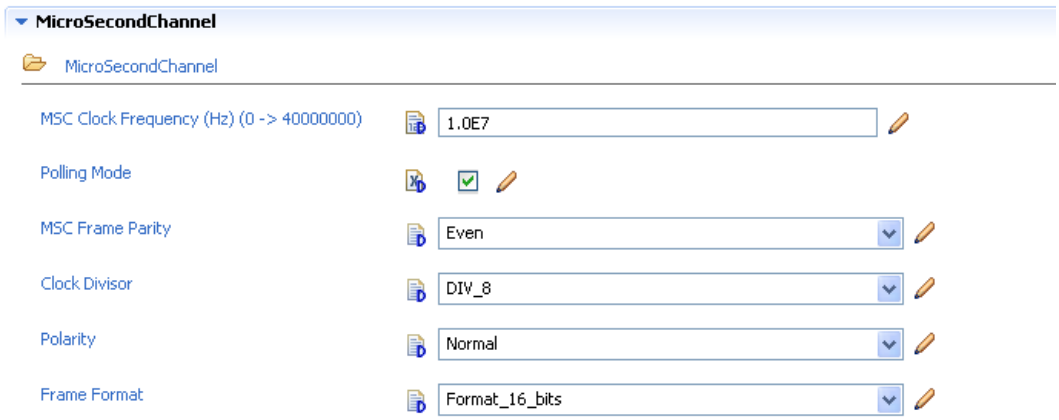

Figure 4-7. Tresos Plugin snapshot for MicroSecondChannel form.

4.6.1.11.1 ClockFrequency (MicroSecondChannel)

ClockFrequency

Autosar Requirements:

This is the frequency for the specific instance of the upstream MSC. It shall be given in Hz. ClockFrequency has to be less than or equal to 40 MHz because max serial output clock frequency is 40MHz.

Table 4-27. Attribute ClockFrequency (MicroSecondChannel) detailed description

Property	Value
Label	MSC Clock Frequency (Hz)
Type	FLOAT
Origin	Custom
Symbolic Name	false
Default	10000000
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range <=40000000 >=0

4.6.1.11.2 PollingMode (MicroSecondChannel)

MSC PollingMode

Autosar Requirements:

Specifies if the eSCI hardware channel supports the Polling mode for data reception.

Table 4-28. Attribute PollingMode (MicroSecondChannel) detailed description

Property	Value
Label	Polling Mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.6.1.11.3 Parity (MicroSecondChannel)

MSC frame parity

Autosar Requirements:

Specifies the parity as Even or Odd.

Table 4-29. Attribute Parity (MicroSecondChannel) detailed description

Property	Value
Label	MSC Frame Parity
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	Even
Lower Multiplicity	1
Upper Multiplicity	1
Range	Even Odd

4.6.1.11.4 ClockDivisor (MicroSecondChannel)

ClockDivisor

Autosar Requirements:

Defines the divisor of the clock for MSC functionality.

Table 4-30. Attribute ClockDivisor (MicroSecondChannel) detailed description

Property	Value
Label	Clock Divisor
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	DIV_8
Lower Multiplicity	1
Upper Multiplicity	1
Range	DIV_8 DIV_16 DIV_32 DIV_64 DIV_128 DIV_256 DIV_512

4.6.1.11.5 Polarity (MicroSecondChannel)

Polarity

Autosar Requirements:

Defines the polarity of the MSC.

Table 4-31. Attribute Polarity (MicroSecondChannel) detailed description

Property	Value
Label	Polarity
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	Normal
Lower Multiplicity	1
Upper Multiplicity	1
Range	Normal Inverted

4.6.1.11.6 FrameFormat (MicroSecondChannel)

FrameFormat

Autosar Requirements:

Defines the frame format of the MSC.

Table 4-32. Attribute FrameFormat (MicroSecondChannel) detailed description

Property	Value
Label	Frame Format
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	Format_16_bits
Lower Multiplicity	1
Upper Multiplicity	1
Range	Format_12_bits Format_16_bits

4.6.1.12 Form DMAsettings

DMA Settings

This container include the needed parameters to configure the DMA hardware channel to work with MicroSecond Channel functionality. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.

Is included by form : LinChannel

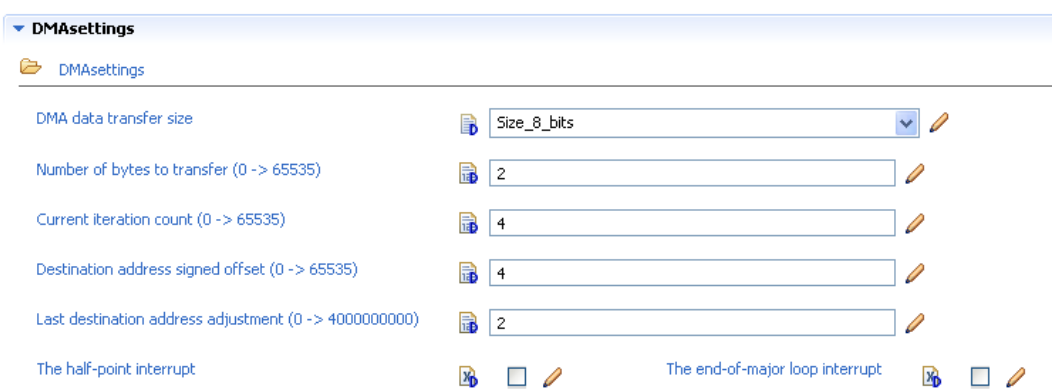

Figure 4-8. Tresos Plugin snapshot for DMAsettings form.

4.6.1.12.1 DMADataSize (DMAsettings)

DMA data transfer size

Defines the size of transfered data. (ssize, dsize bit of TCD register)

note: for more detail about option 16-byte and 32-byte see datasheet

Table 4-33. Attribute DMADataSize (DMAsettings) detailed description

Property	Value
Label	DMA data transfer size
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	Size_8_bits
Lower Multiplicity	1
Upper Multiplicity	1
Range	Size_8_bits Size_16_bits Size_32_bits Size_64_bits Size_16_bytes Size_32_bytes

4.6.1.12.2 nBytesToTransfer (DMAsettings)

Number of bytes to be transferred

Defines the Number of bytes to be transferred in each service request of the channel. (nbytes bit of TCD register)

Table 4-34. Attribute nBytesToTransfer (DMAsettings) detailed description

Property	Value
Label	Number of bytes to transfer
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	2
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=0 <=65535

4.6.1.12.3 iterationCount (DMAsettings)

Current major iteration count

This count represents the current loop count for the channel. It is decremented each time the minor loop is completed and updated in the transfer control descriptor memory.(citer, biter bit of TCD register)

Table 4-35. Attribute iterationCount (DMAsettings) detailed description

Property	Value
Label	Current iteration count
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	4
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=0 <=65535

4.6.1.12.4 destinationAddressOffset (DMAsettings)

Destination address signed offset

Sign-extended offset applied to the current destination address to form the next-state value as each destination write is completed..(doff bit of TCD register)

Table 4-36. Attribute destinationAddressOffset (DMAsettings) detailed description

Property	Value
Label	Destination address signed offset
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	4
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=0 <=65535

4.6.1.12.5 lastDestinationAddress (DMAsettings)

Last destination address adjustment

Adjustment value added to the destination address at the completion of the outer major iteration count. (dlast bit of TCD register)

Table 4-37. Attribute lastDestinationAddress (DMAsettings) detailed description

Property	Value
Label	Last destination address adjustment
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	2
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=0 <=4000000000

4.6.1.12.6 TheHalfPointInterrupt (DMAsettings)

The half-point interrupt

Enable an interrupt when major counter is half complete.

If this flag is set, the channel generates an interrupt request by setting the appropriate bit in the DMAINT register when the current major iteration count reaches the halfway point.The halfway complete interrupt is disabled when biter values are less than two.

Table 4-38. Attribute TheHalfPointInterrupt (DMAsettings) detailed description

Property	Value
Label	The half-point interrupt
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.6.1.12.7 TheEndOfMajorLoopInterrupt (DMAsettings)

The end-of-major loop interrupt

Enable an interrupt when major iteration count completes.

If this flag is set, the channel generates an interrupt request by setting the appropriate bit in the DMAINT register when the current major iteration count reaches zero.

Table 4-39. Attribute TheEndOfMajorLoopInterrupt (DMAsettings) detailed description

Property	Value
Label	The end-of-major loop interrupt
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: UM14LINASR3.0R2.0.0
Rev. 2.0