

MCAL Configuration Verification Manual for Lin_17_AscLin

32-bit TriCore™ AURIX™ TC3xx microcontroller family

About this document

Scope and purpose

This Configuration Data Reference document is applicable to all TC3xx devices in the TriCore™ AURIX™ family of 32-bit microcontrollers.

The purpose of this document is to facilitate the integrator to verify the generated code based on the input configuration parameters. This document describes details of structures, defines, macros and variables generated from the configuration parameters.

Intended audience

This document is intended for integrators who need to understand the logic of the generated configuration code of AURIX™ AUTOSAR MCAL.

Reference documents

This document should be read in conjunction with the following documents:

- AURIX™ TC3xx MCAL User Manual Lin_17_AscLin

Table of contents

About this document.....	1
Table of contents.....	2
1 Lin_17_AscLin driver	4
1.1 File: Lin_17_AscLin_Cfg.h	4
1.1.1 Macro: LIN_17_ASCLIN_AR_RELEASE_MAJOR_VERSION.....	4
1.1.2 Macro: LIN_17_ASCLIN_AR_RELEASE_MINOR_VERSION	4
1.1.3 Macro: LIN_17_ASCLIN_AR_RELEASE_REVISION_VERSION.....	4
1.1.4 Macro: LIN_17_ASCLIN_SW_MAJOR_VERSION.....	5
1.1.5 Macro: LIN_17_ASCLIN_SW_MINOR_VERSION	5
1.1.6 Macro: LIN_17_ASCLIN_SW_PATCH_VERSION	6
1.1.7 Macro: LIN_17_ASCLIN_DEV_ERROR_DETECT	6
1.1.8 Macro: LIN_17_ASCLIN_MULTICORE_ERROR_DETECT	6
1.1.9 Macro: LIN_17_ASCLIN_VERSION_INFO_API	7
1.1.10 Macro: LIN_17_ASCLIN_INDEX	7
1.1.11 Macro: LIN_17_ASCLIN_MAX_HW_UNIT.....	7
1.1.12 Macro: LIN_17_ASCLIN_MAXIMUM_CHANNEL_CONFIGURED.....	8
1.1.13 Macro: LIN_17_ASCLIN_CLC	8
1.1.14 Macro: LIN_17_ASCLIN<x>	8
1.1.15 Macro: LIN_17_ASCLIN_REGADDR.....	10
1.1.16 Macro: LIN_17_ASCLIN_CSRREG_CLKSEL.....	11
1.1.17 Macro: LIN_17_ASCLIN_INIT_API_MODE.....	11
1.1.18 Macro: LIN_17_ASCLIN_MCAL_SUPERVISOR_MODE.....	11
1.1.19 Macro: LIN_17_ASCLIN_MCAL_USER1_MODE	12
1.1.20 Macro: LIN_17_ASCLIN_INTERRUPT_ENABLE	12
1.1.21 LIN_17_ASCLIN_MAX_CHANNELS_CORE<x>	12
1.1.22 Macro: LIN_17_ASCLIN_GLOBAL_CHANNEL_WAKEUP_SUPPORT	13
1.1.23 Macro: LIN_17_ASCLIN_HW_ERROR_DEM_REPORT	13
1.1.24 Macro: LIN_17_ASCLIN_E_TIMEOUT	14
1.1.25 Macro: LIN_17_ASCLIN_MASTER_CH_PRESENT.....	14
1.1.26 Macro: LIN_17_ASCLIN_SLAVE_CH_PRESENT	15
1.1.27 Macro: LIN_17_ASCLIN_TIMEOUT_COUNT	15
1.1.28 Macro: LIN_17_ASCLIN_CSRREG_CLKSEL_SLOWCLK	15
1.1.29 Macro: LIN_17_ASCLIN_CSRREG_CLKSEL_FASTCLK.....	16
1.1.30 Macro: LinConf_LinChannel_<LinChannelName>	16
1.2 File: Lin_17_AscLin[_<variant>]_PBcfg.c	16
1.2.1 Structure: Lin_17_AscLin_Config[_<variant>]	17
1.2.1.1 Member: Lin_CoreConfigCore[x][_<variant>].....	19
1.2.1.2 Member: Lin_17_AscLin_Channel_To_Core_SpecificArray [_<variant>] [LIN_17_ASCLIN_MAXIMUM_CHANNEL_CONFIGURED]	20
1.2.1.3 Member: Lin_17_AscLin_Hw_To_ChannelIdMap[_<variant>] [LIN_17_ASCLIN_MAX_HW_UNIT]	21
1.2.1.4 Member: TotalChannel	21
1.2.2 Structure: Lin_CoreConfigCore[x] [_<variant>]	22
1.2.2.1 Member: MaxCoreChannels.....	23
1.2.2.2 Member: Lin_kChannelConfigurationCore[x][_<variant>][TotalChannel]	23
1.2.3 Structure: Lin_kChannelConfigurationCore[x][_<variant>][TotalChannel]	24

Lin_17_AscLin driver

1.2.3.1	Member: WakeupSourceId	27
1.2.3.2	Member: HwModule	27
1.2.3.3	Member: Wakeup	28
1.2.3.4	Member: RxAlternatePinSelect	28
1.2.3.5	Member: LogicalId	29
1.2.3.6	Member: Node	29
1.2.4	Structure: Lin_17_AscLin_Hw_To_ChannelIdMap[_<variant>] [LIN_17_ASCLIN_MAX_HW_UNIT]	29
1.2.5	Structure: Lin_17_AscLin_Channel_To_Core_SpecificArray [_<variant>] [LIN_17_ASCLIN_MAXIMUM_CHANNEL_CONFIGURED]	31
1.3	File: Lin_17_AscLin[_<variant>]_PBcfg.h	32
1.3.1	Structure: Lin_17_AscLin_Config[_<variant>]	32
Revision history.....		33

1 Lin_17_AscLin driver

This chapter describes the details of the configuration data generated from the Lin_17_AscLin driver.

1.1 File: Lin_17_AscLin_Cfg.h

The generated header file contains all pre-compile configuration parameters. Pre-compile time configuration allows decoupling of the static configuration from implementation. The file is generated in 'inc' folder.

1.1.1 Macro: LIN_17_ASCLIN_AR_RELEASE_MAJOR_VERSION

Table 1 LIN_17_ASCLIN_AR_RELEASE_MAJOR_VERSION

Name	LIN_17_ASCLIN_AR_RELEASE_MAJOR_VERSION	
Description	Major version number of AUTOSAR release on which the Lin_17_AscLin implementation is based on.	
Verification method	<p>The macro is generated with the value present in 'CommonPublishedInformation/ArMajorVersion'.</p> <p><i>Note: The macro is not user configurable.</i></p>	
Example(s)	Action	Generated output
	Generate Lin_17_AscLin_Cfg.h file with ArMajorVersion 4	<pre>#define LIN_17_ASCLIN_AR_RELEASE_MAJOR_VERSION (4U)</pre>

1.1.2 Macro: LIN_17_ASCLIN_AR_RELEASE_MINOR_VERSION

Table 2 LIN_17_ASCLIN_AR_RELEASE_MINOR_VERSION

Name	LIN_17_ASCLIN_AR_RELEASE_MINOR_VERSION	
Description	Minor version number of AUTOSAR release on which the Lin_17_AscLin implementation is based on.	
Verification method	<p>The macro is generated with the value present in 'CommonPublishedInformation/ArMinorVersion'.</p> <p><i>Note: The macro is not user configurable.</i></p>	
Example(s)	Action	Generated output
	Generate Lin_17_AscLin_Cfg.h file with ArMinorVersion 2	<pre>#define LIN_17_ASCLIN_AR_RELEASE_MINOR_VERSION (2U)</pre>

1.1.3 Macro: LIN_17_ASCLIN_AR_RELEASE_REVISION_VERSION

Table 3 LIN_17_ASCLIN_AR_RELEASE_REVISION_VERSION

Name	LIN_17_ASCLIN_AR_RELEASE_REVISION_VERSION	
Description	Revision version number of AUTOSAR release on which the Lin_17_AscLin implementation is based on.	
Verification method	<p>The macro is generated with the value present in 'CommonPublishedInformation/ArPatchVersion'.</p> <p><i>Note: The macro is not user configurable.</i></p>	
Example(s)	Action	Generated output
	Generate Lin_17_AscLin_Cfg.h file with ArPatchVersion 2	#define LIN_17_ASCLIN_AR_RELEASE_PATCH_VERSION (2U)

1.1.4 Macro: LIN_17_ASCLIN_SW_MAJOR_VERSION

Table 4 LIN_17_ASCLIN_SW_MAJOR_VERSION

Name	LIN_17_ASCLIN_SW_MAJOR_VERSION	
Description	Major version number of the Lin_17_AscLin module.	
Verification method	<p>The macro is generated with the value present in 'CommonPublishedInformation/SwMajorVersion'.</p> <p><i>Note: The macro is not user configurable.</i></p>	
Example(s)	Action	Generated output
	Generate Lin_17_AscLin_Cfg.h file with SwMajorVersion 10	#define LIN_17_ASCLIN_SW_MAJOR_VERSION (10U)

1.1.5 Macro: LIN_17_ASCLIN_SW_MINOR_VERSION

Table 5 LIN_17_ASCLIN_SW_MINOR_VERSION

Name	LIN_17_ASCLIN_SW_MINOR_VERSION	
Description	Minor version number of the Lin_17_AscLin module.	
Verification method	<p>The macro is generated with the value present in 'CommonPublishedInformation/SwMinorVersion'.</p> <p><i>Note: The macro is not user configurable.</i></p>	
Example(s)	Action	Generated output
	Generate Lin_17_AscLin_Cfg.h file with SwMinorVersion 10	#define LIN_17_ASCLIN_SW_MINOR_VERSION (10U)

1.1.6 Macro: LIN_17_ASCLIN_SW_PATCH_VERSION

Table 6 LIN_17_ASCLIN_SW_PATCH_VERSION

Name	LIN_17_ASCLIN_SW_PATCH_VERSION	
Description	Patch level version number of the Lin_17_AscLin module.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/SwPatchVersion'. <i>Note: The macro is not user configurable.</i>	
Example(s)	Action	Generated output
	Generate Lin_17_AscLin_Cfg.h file with SwPatchVersion 0	<pre>#define LIN_17_ASCLIN_SW_PATCH_VERSION (0U)</pre>

1.1.7 Macro: LIN_17_ASCLIN_DEV_ERROR_DETECT

Table 7 LIN_17_ASCLIN_DEV_ERROR_DETECT

Name	LIN_17_ASCLIN_DEV_ERROR_DETECT	
Description	Enables/disables the Development Error Detection.	
Verification method	The macro is generated as STD_ON if LinDevErrorDetect configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	Action	Generated output
	LinDevErrorDetect = True	<pre>#define LIN_17_ASCLIN_DEV_ERROR_DETECT (STD_ON)</pre>
	LinDevErrorDetect = False	<pre>#define LIN_17_ASCLIN_DEV_ERROR_DETECT (STD_OFF)</pre>

1.1.8 Macro: LIN_17_ASCLIN_MULTICORE_ERROR_DETECT

Table 8 LIN_17_ASCLIN_MULTICORE_ERROR_DETECT

Name	LIN_17_ASCLIN_MULTICORE_ERROR_DETECT	
Description	Enables/disables the Multicore Error Detection.	
Verification method	The macro is generated as STD_ON if LinMultiCoreErrorDetect configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	Action	Generated output
	LinMultiCoreErrorDetect = True	<pre>#define LIN_17_ASCLIN_MULTICORE_ERROR_DETECT (STD_ON)</pre>

LinMultiCoreErrorDetect = False	#define LIN_17_ASCLIN_MULTICORE_ERROR_DETECT (STD_OFF)
------------------------------------	--

1.1.9 Macro: LIN_17_ASCLIN_VERSION_INFO_API

Table 9 LIN_17_ASCLIN_VERSION_INFO_API

Name	LIN_17_ASCLIN_VERSION_INFO_API	
Description	Enables/disables Lin_17_AscLin_GetVersionInfo API	
Verification method	The macro is generated as STD_ON if LinVersionInfoApi configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	Action	Generated output
	LinVersionInfoApi= True	#define LIN_17_ASCLIN_VERSION_INFO_API (STD_ON)
	LinVersionInfoApi= False	#define LIN_17_ASCLIN_VERSION_INFO_API (STD_OFF)

1.1.10 Macro: LIN_17_ASCLIN_INDEX

Table 10 LIN_17_ASCLIN_INDEX

Name	LIN_17_ASCLIN_INDEX	
Description	Instance ID of Lin module instance.	
Verification method	The macro is generated as a numeric value set in the configuration parameter 'LinGeneral/LinIndex'	
Example(s)	Action	Generated output
	Set LinIndex as 0	#define LIN_17_ASCLIN_INDEX (0U)
	Set LinIndex as 240	#define LIN_17_ASCLIN_INDEX (240U)

1.1.11 Macro: LIN_17_ASCLIN_MAX_HW_UNIT

Table 11 LIN_17_ASCLIN_MAX_HW_UNIT

Name	LIN_17_ASCLIN_MAX_HW_UNIT	
Description	Indicates the total number of ASCLIN hardware modules.	
Verification method	The macro is generated as total number of ASCLIN hardware modules.	
	<i>Note: This macro is not configurable by the user.</i>	
Example(s)	Action	Generated output

Choose the device with number of ASCLIN hardware modules as 12	#define LIN_17_ASCLIN_MAX_HW_UNIT (12U)
Choose the device with number of ASCLIN hardware modules as 24	#define LIN_17_ASCLIN_MAX_HW_UNIT (24U)

1.1.12 Macro: LIN_17_ASCLIN_MAXIMUM_CHANNEL_CONFIGURED

Table 12 LIN_17_ASCLIN_MAXIMUM_CHANNEL_CONFIGURED

Name	LIN_17_ASCLIN_MAXIMUM_CHANNEL_CONFIGURED	
Description	Indicates the total number of channels configured.	
Verification method	The macro is generated as a numeric value which corresponds to the number of elements in the list 'LinGlobalConfig/LinChannel'.	
Example(s)	Action	Generated output
	Configure 4 LIN channels (LinChannel0 to LinChannel3)	#define LIN_17_ASCLIN_MAXIMUM_CHANNEL (4)
	Configure 24 LIN channels (LinChannel0 to LinChannel23)	#define LIN_17_ASCLIN_MAXIMUM_CHANNEL (24)

1.1.13 Macro: LIN_17_ASCLIN_CLC

Table 13 LIN_17_ASCLIN_CLC

Name	LIN_17_ASCLIN_CLC	
Description	Enables/disables ASCLIN module sleep request	
Verification method	The macro is generated as a numeric value which corresponds to the ASCLIN module sleep request is enabled or disabled from the list 'LinGeneral/LinHwMcuTrigSleepEnable'.	
Example(s)	Action	Generated output
	LinHwMcuTrigSleepEnable = True	#define LIN_17_ASCLIN_CLC (0x0)
	LinHwMcuTrigSleepEnable = False	#define LIN_17_ASCLIN_CLC (0x8)

1.1.14 Macro: LIN_17_ASCLIN<x>

Table 14 LIN_17_ASCLIN<x>

Name	LIN_17_ASCLIN<x>
Description	Indicates the ASCLIN hardware module available for the device. (x ranges from 0 to maximum number of ASCLIN hardware module available)

Verification method	<p>The macro is generated as a numeric value which corresponds to the ASCLIN hardware modules for a channel.</p> <p><i>Note: This macro is not configurable by the user.</i></p>	
Example(s)	Action	Generated output
	Choose the device with consecutive number of ASCLIN hardware module as 12.	<pre>#define LIN_17_ASCLIN0 (0U) #define LIN_17_ASCLIN1 (1U) #define LIN_17_ASCLIN2 (2U) #define LIN_17_ASCLIN3 (3U) #define LIN_17_ASCLIN4 (4U) #define LIN_17_ASCLIN5 (5U) #define LIN_17_ASCLIN6 (6U) #define LIN_17_ASCLIN7 (7U) #define LIN_17_ASCLIN8 (8U) #define LIN_17_ASCLIN9 (9U) #define LIN_17_ASCLIN10 (10U) #define LIN_17_ASCLIN11 (11U)</pre>
	Choose the device with consecutive number of ASCLIN hardware module as 24.	<pre>#define LIN_17_ASCLIN0 (0U) #define LIN_17_ASCLIN1 (1U) #define LIN_17_ASCLIN2 (2U) #define LIN_17_ASCLIN3 (3U) #define LIN_17_ASCLIN4 (4U) #define LIN_17_ASCLIN5 (5U) #define LIN_17_ASCLIN6 (6U) #define LIN_17_ASCLIN7 (7U) #define LIN_17_ASCLIN8 (8U) #define LIN_17_ASCLIN9 (9U) #define LIN_17_ASCLIN10 (10U) #define LIN_17_ASCLIN11 (11U) #define LIN_17_ASCLIN12 (12U) #define LIN_17_ASCLIN13 (13U) #define LIN_17_ASCLIN14 (14U) #define LIN_17_ASCLIN15 (15U) #define LIN_17_ASCLIN16 (16U) #define LIN_17_ASCLIN17 (17U) #define LIN_17_ASCLIN18 (18U) #define LIN_17_ASCLIN19 (19U) #define LIN_17_ASCLIN20 (20U)</pre>

	<pre>#define LIN_17_ASCLIN21 (21U) #define LIN_17_ASCLIN22 (22U) #define LIN_17_ASCLIN23 (23U)</pre>
Choose the device with 5 non consecutive number of ASCLIN hardware module.	<pre>#define LIN_17_ASCLIN0 (0U) #define LIN_17_ASCLIN1 (1U) #define LIN_17_ASCLIN2 (2U) #define LIN_17_ASCLIN3 (3U) #define LIN_17_ASCLIN8 (8U)</pre>

1.1.15 Macro: LIN_17_ASCLIN_REGADDR

Table 15 LIN_17_ASCLIN_REGADDR

Name	LIN_17_ASCLIN_REGADDR	
Description	Indicates the offset address for all ASCLIN hardware modules	
Verification method	<p>The macro is generated as an offset address which corresponds to the ASCLIN hardware modules.</p> <p><i>Note: This macro is not configurable by the user.</i></p>	
Example(s)	Action	Generated output
	Choose the device with consecutive number of ASCLIN hardware module as 12.	<pre>#define LIN_17_ASCLIN_REGADDR &MODULE_ASCLIN0, &MODULE_ASCLIN1, &MODULE_ASCLIN2, &MODULE_ASCLIN3, &MODULE_ASCLIN4, &MODULE_ASCLIN5, &MODULE_ASCLIN6, &MODULE_ASCLIN7, &MODULE_ASCLIN8, &MODULE_ASCLIN9, &MODULE_ASCLIN10, &MODULE_ASCLIN11</pre>
	Choose the device with consecutive number of ASCLIN hardware module as 24.	<pre>#define LIN_17_ASCLIN_REGADDR &MODULE_ASCLIN0, &MODULE_ASCLIN1, &MODULE_ASCLIN2, &MODULE_ASCLIN3, &MODULE_ASCLIN4, &MODULE_ASCLIN5, &MODULE_ASCLIN6, &MODULE_ASCLIN7, &MODULE_ASCLIN8, &MODULE_ASCLIN9, &MODULE_ASCLIN10, &MODULE_ASCLIN11, &MODULE_ASCLIN12, &MODULE_ASCLIN13, &MODULE_ASCLIN14, &MODULE_ASCLIN15, &MODULE_ASCLIN16, &MODULE_ASCLIN17, &MODULE_ASCLIN18, &MODULE_ASCLIN19, &MODULE_ASCLIN20, &MODULE_ASCLIN21, &MODULE_ASCLIN22, &MODULE_ASCLIN23</pre>
	Choose the device with 5 non consecutive number of ASCLIN hardware module.	<pre>#define LIN_17_ASCLIN_REGADDR &MODULE_ASCLIN0, &MODULE_ASCLIN1, &MODULE_ASCLIN2, &MODULE_ASCLIN3, NULL_PTR, NULL_PTR, NULL_PTR, NULL_PTR, &MODULE_ASCLIN8</pre>

1.1.16 Macro: LIN_17_ASCLIN_CSRREG_CLKSEL

Table 16 LIN_17_ASCLIN_CSRREG_CLKSEL

Name	LIN_17_ASCLIN_CSRREG_CLKSEL	
Description	Indicates the ASCLIN peripheral frequency	
Verification method	The macro is generated as a numeric value which corresponds to the peripheral frequency for ASCLIN module from the list 'LinGeneral/LinCsrClksel'.	
Example(s)	Action	Generated output
	LinCsrClksel = ASCLINF	#define LIN_17_ASCLIN_CSRREG_CLKSEL (LIN_17_ASCLIN_CSRREG_CLKSEL_FASTCLK)
	LinCsrClksel = ASCLINS	#define LIN_17_ASCLIN_CSRREG_CLKSEL (LIN_17_ASCLIN_CSRREG_CLKSEL_SLOWCLK)

1.1.17 Macro: LIN_17_ASCLIN_INIT_API_MODE

Table 17 LIN_17_ASCLIN_INIT_API_MODE

Name	LIN_17_ASCLIN_INIT_API_MODE	
Description	Decides the mode of execution of Init API.	
Verification method	The macro is generated as with the value present in LinGeneral/LinInitApiMode.	
Example(s)	Action	Generated output
	LinInitApiMode = LIN_MCAL_SUPERVISOR	#define LIN_17_ASCLIN_INIT_API_MODE (LIN_17_ASCLIN_MCAL_SUPERVISOR_MODE)
	LinInitApiMode = LIN_MCAL_USER1	#define LIN_17_ASCLIN_INIT_API_MODE (LIN_17_ASCLIN_MCAL_USER1_MODE)

1.1.18 Macro: LIN_17_ASCLIN_MCAL_SUPERVISOR_MODE

Table 18 LIN_17_ASCLIN_MCAL_SUPERVISOR_MODE

Name	LIN_17_ASCLIN_MCAL_SUPERVISOR_MODE	
Description	Indicates the mode of execution of Init API.	
Verification method	The macro is generated as STD_ON if LinInitApiMode configuration parameter is set to 'LIN_MCAL_SUPERVISOR' else the macro is generated as STD_OFF.	
Example(s)	Action	Generated output
	LinInitApiMode = True	#define LIN_17_ASCLIN_MCAL_SUPERVISOR_MODE (STD_ON)
	LinInitApiMode = False	#define LIN_17_ASCLIN_MCAL_SUPERVISOR_MODE (STD_OFF)

1.1.19 Macro: LIN_17_ASCLIN_MCAL_USER1_MODE

Table 19 LIN_17_ASCLIN_MCAL_USER1_MODE

Name	LIN_17_ASCLIN_MCAL_USER1_MODE	
Description	Indicates the mode of execution of Init API.	
Verification method	The macro is generated as STD_ON if LinInitDelInitApiMode configuration parameter is set to 'LIN_MCAL_USER1' else the macro is generated as STD_OFF.	
Example(s)	Action	Generated output
	LinInitApiMode = True	#define LIN_17_ASCLIN_MCAL_USER1_MODE (STD_ON)
	LinInitApiMode = False	#define LIN_17_ASCLIN_MCAL_USER1_MODE (STD_OFF)

1.1.20 Macro: LIN_17_ASCLIN_INTERRUPT_ENABLE

Table 20 LIN_17_ASCLIN_INTERRUPT_ENABLE

Name	LIN_17_ASCLIN_INTERRUPT_ENABLE	
Description	Enable/disable interrupt features	
Verification method	The macro is generated as STD_ON if LinGeneral/ LinMasterInterruptEnable configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	Action	Generated output
	LinMasterInterruptEnable = True	#define LIN_17_ASCLIN_INTERRUPT_ENABLED (STD_ON)
	LinMasterInterruptEnable = False	#define LIN_17_ASCLIN_INTERRUPT_ENABLED (STD_OFF)

1.1.21 LIN_17_ASCLIN_MAX_CHANNELS_CORE<x>

Table 21 LIN_17_ASCLIN_MAX_CHANNELS_CORE<x>

Name	LIN_17_ASCLIN_MAX_CHANNELS_CORE<x>	
Description	Indicates the total number of channels configured for CORE<x>.	
Verification method	The macro is generated as total number of controllers allocated to CORE<x>.	
	<i>Note: Channels not assigned to any core are assigned to master core (ResourceMMasterCore).</i>	

Example(s)	Action	Generated output
	<ul style="list-style-type: none"> Configure 4 LIN channels. Set ResourceMMasterCore as CORE1. Do not assign LIN controllers in any ResourceMAllocation	<pre>#define LIN_17_ASCLIN_MAX_CHANNELS_CORE1 (4U)</pre>
	<ul style="list-style-type: none"> Configure 4 LIN channels. Set ResourceMMasterCore as CORE1. Assign 3 channels configured under ResourceMAllocation with ResourceMCoreID as CORE0. 	<pre>#define LIN_17_ASCLIN_MAX_CHANNELS_CORE0 (3U) #define LIN_17_ASCLIN_MAX_CHANNELS_CORE1 (1U)</pre>

1.1.22 Macro: LIN_17_ASCLIN_GLOBAL_CHANNEL_WAKEUP_SUPPORT

Table 22 LIN_17_ASCLIN_GLOBAL_CHANNEL_WAKEUP_SUPPORT

Name	LIN_17_ASCLIN_GLOBAL_CHANNEL_WAKEUP_SUPPORT	
Description	Enable/disable channel wakeup feature.	
Verification method	The macro is generated as STD_ON if LinGlobalConfig/LinChannel/LinChannelWakeupSupport configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	Action	Generated output
	LinChannelWakeupSupport = True	<pre>#define LIN_17_ASCLIN_GLOBAL_CHANNEL_WAKEUP_SUPPORT (STD_ON)</pre>
	LinChannelWakeupSupport = False	<pre>#define LIN_17_ASCLIN_GLOBAL_CHANNEL_WAKEUP_SUPPORT (STD_OFF)</pre>

1.1.23 Macro: LIN_17_ASCLIN_HW_ERROR_DEM_REPORT

Table 23 LIN_17_ASCLIN_HW_ERROR_DEM_SUPPORT

Name	LIN_17_ASCLIN_HW_ERROR_DEM_REPORT	
Description	Enable or disable Production Error reporting.	
Verification method	The macro is generated as LIN_17_ASCLIN_DEM_REPORT_ENABLED if the reference node LinDemEventParameterRefs exists else the macro is generated as LIN_17_ASCLIN_DEM_REPORT_DISABLED.	
Example(s)	Action	Generated output
	LinDemEventParameterRefs exists = True	<pre>#define LIN_17_ASCLIN_HW_ERROR_DEM_REPORT (LIN_17_ASCLIN_DEM_REPORT_ENABLED)</pre>

	LinDemEventParameterRefs exists = False	#define LIN_17_ASCLIN_HW_ERROR_DEM_REPORT (LIN_17_ASCLIN_DEM_REPORT_DISABLED)
--	--	---

1.1.24 Macro: LIN_17_ASCLIN_E_TIMEOUT

Table 24 LIN_17_ASCLIN_E_TIMEOUT

Name	LIN_17_ASCLIN_E_TIMEOUT	
Description	Production Error reported when timeout caused by hardware bit not set/reset.	
Verification method	The macro is generated as '((Dem_EventIdType) DemConf_DemEventParameter_x)' where x is the node value configured in 'LinDemEventParameterRefs/ LinDemEventParameterRefs_0/ LIN_E_TIMEOUT'.	
Example(s)	Action	Generated output
	Configure the node in 'LinDemEventParameterRefs/LinDemEventParameterRefs_0/LIN_E_TIMEOUT' = DemEventParameter_0	#define LIN_17_ASCLIN_E_TIMEOUT ((Dem_EventIdType) DemConf_DemEventParameter_DemEventParameter_0)
Example(s)	Configure the node in 'LinDemEventParameterRefs/LinDemEventParameterRefs_0/LIN_E_TIMEOUT' = DemEventParameter_2	#define LIN_17_ASCLIN_E_TIMEOUT ((Dem_EventIdType) DemConf_DemEventParameter_DemEventParameter_2)

1.1.25 Macro: LIN_17_ASCLIN_MASTER_CH_PRESENT

Table 25 LIN_17_ASCLIN_MASTER_CH_PRESENT

Name	LIN_17_ASCLIN_MASTER_CH_PRESENT	
Description	Determines whether the MASTER node is configured for at-least 1 or more channels.	
Verification method	This macro is generated as (STD_ON) if the MASTER node is configured for at-least 1 or more channels else the macro is generated as (STD_OFF). Note: For AUTOSAR version 422, this parameter will always be STD_ON.	
Example(s)	Action	Generated output
	AUTOSAR version is 422	#define LIN_17_ASCLIN_MASTER_CH_PRESENT (STD_ON)
	Atleast one master node is present in the configured channels.	#define LIN_17_ASCLIN_MASTER_CH_PRESENT (STD_ON)
Example(s)	All channels are configured as slave i.e. pure slave configuration	#define LIN_17_ASCLIN_MASTER_CH_PRESENT (STD_OFF)

1.1.26 Macro: LIN_17_ASCLIN_SLAVE_CH_PRESENT

Table 26 LIN_17_ASCLIN_SLAVE_CH_PRESENT

Name	LIN_17_ASCLIN_SLAVE_CH_PRESENT	
Description	Determines whether the SLAVE node is configured for at-least 1 or more channels.	
Verification method	This macro is generated as (STD_ON) if the SLAVE node is configured for at-least 1 or more channels else the macro is generated as (STD_OFF).	
Example(s)	Action	Generated output
	Atleast one slave node is present in the configured channels.	#define LIN_17_ASCLIN_SLAVE_CH_PRESENT (STD_ON)
	All channels are configured as master i.e. pure master configuration	#define LIN_17_ASCLIN_SLAVE_CH_PRESENT (STD_OFF)

1.1.27 Macro: LIN_17_ASCLIN_TIMEOUT_COUNT

Table 27 LIN_17_ASCLIN_TIMEOUT_COUNT

Name	LIN_17_ASCLIN_TIMEOUT_COUNT	
Description	Specifies the wait count for the hardware bit set/reset.	
Verification method	This macro is generated as a numeric value which corresponds to the value of the configuration parameter 'LinGeneral/LinTimeoutDuration'.	
Example(s)	Action	Generated output
	LinTimeoutDuration = 300	#define LIN_17_ASCLIN_TIMEOUT_COUNT (300U)
	LinTimeoutDuration = 100	#define LIN_17_ASCLIN_TIMEOUT_COUNT (100U)

1.1.28 Macro: LIN_17_ASCLIN_CSRREG_CLKSEL_SLOWCLK

Table 28 LIN_17_ASCLIN_CSRREG_CLKSEL_SLOWCLK

Name	LIN_17_ASCLIN_CSRREG_CLKSEL_SLOWCLK	
Description	This macro is generated as a numeric value which is used when ASCLIN peripheral frequency is configured for slow mode.	
Verification method	The macro is generated with the value 4.	
	<i>Note: The macro is not user configurable.</i>	
Example(s)	Action	Generated output
	Generate Lin_17_AscLin_Cfg.h file	#define LIN_17_ASCLIN_CSRREG_CLKSEL_SLOWCLK (4U)

1.1.29 Macro: LIN_17_ASCLIN_CSRREG_CLKSEL_FASTCLK

Table 29 LIN_17_ASCLIN_CSRREG_CLKSEL_FASTCLK

Name	LIN_17_ASCLIN_CSRREG_CLKSEL_FASTCLK	
Description	This macro is generated as a numeric value which is used when ASCLIN peripheral frequency is configured for fast mode.	
Verification method	The macro is generated with the value 2. <i>Note: The macro is not user configurable.</i>	
Example(s)	Action	Generated output
	Generate Lin_17_AscLin_Cfg.h file	#define LIN_17_ASCLIN_CSRREG_CLKSEL_FASTCLK (2U)

1.1.30 Macro: LinConf_LinChannel_<LinChannelName>

Table 30 LinConf_LinChannel_<LinChannelName>

Name	LinConf_LinChannel_<LinChannelName>	
Description	Specifies the symbolic name with LinChannelId for each configured LinChannel.	
Verification method	This macro generates the symbolic name with LinChannelId for each configured LinChannel.	
Example(s)	Action	Generated output
	<ul style="list-style-type: none"> Configure LIN channel 0 and channel 1. 	<pre>#ifndef Lin_17_AscLinConf_LinChannel_LinChannel_0 #define Lin_17_AscLinConf_LinChannel_LinChannel_0 (0U) #endif #ifndef Lin_17_AscLinConf_LinChannel_LinChannel_1 #define Lin_17_AscLinConf_LinChannel_LinChannel_1 (1U) #endif</pre>

1.2 File: Lin_17_AscLin[_<variant>]_PBcfg.c

The generated source file contains all post-build configuration parameters. Post-build time configuration mechanism allows configurable functionality of LIN driver that is deployed as object code. The file is generated in 'src' folder.

1.2.1 Structure: Lin_17_AscLin_Config[_<variant>]

Table 31 Lin_17_AscLin_Config [_<variant>]

Name	Lin_17_AscLin_Config[_<variant>]	
Type	Lin_17_AscLin_ConfigType	
Description	Root configuration structure of LIN driver which will be used during initialization.	
Verification method	The generated structure is present in Lin_17_AscLin[_<variant>]_PBcfg.c file. The <variant> indicates the name of the post-build variant. For a variant-aware configuration the structure name is appended with the variant name. For variant-unaware configuration <variant> is ignored.	
Example(s)	Action	Generated output
	<ul style="list-style-type: none"> Configure LIN channel 0, channel 1, channel 2 and channel 3 assigned to hardware module ASCLIN0, ASCLIN2, ASCLIN3 and ASCLIN11 respectively. (variant-unaware) Set ResourceMMasterCore as CORE5. Assign 2 channels configured under ResourceMAllocation with ResourceMCoreID as CORE5. Assign 1 channel configured under ResourceMAllocation with ResourceMCoreID as CORE0. The channel not assigned to any 	<pre>const Lin_17_AscLin_ConfigType Lin_17_AscLin_Config = { { (Lin_17_AscLin_CoreConfigType*) &Lin_CoreConfigCore0 , NULL_PTR, NULL_PTR, NULL_PTR, NULL_PTR, (Lin_17_AscLin_CoreConfigType*) &Lin_CoreConfigCore5 }, &Lin_17_AscLin_Channel_To_Core_SpecificArray[0], &Lin_17_AscLin_Hw_To_ChannelIdMap[0], 4U } };</pre>

core is by default allocated to master core	
<ul style="list-style-type: none"> Configure LIN channel0 assigned to hardware module ASCLIN0 Configure LIN channel1 assigned to hardware module ASCLIN1 (variant-unaware) Set ResourceMMasterCore as CORE0. Assign both channels configured under ResourceMAllocation with ResourceMCoreID as CORE0. 	<pre>const Lin_17_AscLin_ConfigType Lin_17_AscLin_Config { { (Lin_17_AscLin_CoreConfigType*) &Lin_CoreConfigCore0 , NULL_PTR, NULL_PTR, NULL_PTR, NULL_PTR, NULL_PTR }, &Lin_17_AscLin_Channel_To_Core_SpecificArray[0], &Lin_17_AscLin_Hw_To_ChannelIdMap[0], 2U } };</pre>
<ul style="list-style-type: none"> Configure LIN channel0 assigned to hardware module ASCLIN0 (variant-aware. Variant name is 'Petrol') Set ResourceMMasterCore as CORE0. 	<pre>const Lin_17_AscLin_ConfigType Lin_17_AscLin_Config_Petrol { { (Lin_17_AscLin_CoreConfigType*) &Lin_CoreConfigCore0_Petrol, NULL_PTR, NULL_PTR, NULL_PTR, NULL_PTR, NULL_PTR }, &Lin_17_AscLin_Channel_To_Core_SpecificArray_Petrol [0], &Lin_17_AscLin_Hw_To_ChannelIdMap_Petrol[0],</pre>

	1U
	};

1.2.1.1 Member: Lin_CoreConfigCore[x][_<variant>]

Table 32 Lin_CoreConfigCore[x][_<variant>]

Name	Lin_CoreConfigCore[x][_<variant>]	
Type	Lin_17_AscLin_CoreConfigType	
Description	Array of core-specific configuration. (x ranges from 0 to 5)	
Verification method	The generated structure member is present in the Lin_17_AscLin_Config[_<variant>] structure. If a Core<x> is allocated at least one channel, then the element <x> shall be generated as 'Lin_kChannelConfigurationCore[x][_<variant>][TotalChannel]'.	
Example(s)	Action	Generated output
	All the LIN channels are allocated to Core 0 (variant-unaware)	<pre>{ (Lin_17_AscLin_CoreConfigType*) &Lin_CoreConfigCore0 NULL_PTR, NULL_PTR, NULL_PTR, NULL_PTR, NULL_PTR }</pre>
	Four LIN channels are allocated to Core 0 (variant-aware. Variant name is 'Petrol')	<pre>{ (Lin_17_AscLin_CoreConfigType*) &Lin_CoreConfigCore0 _Petrol, NULL_PTR, NULL_PTR, NULL_PTR, NULL_PTR, NULL_PTR }</pre>
	Four channels are split between all cores except Core 0. (variant-unaware)	<pre>{ NULL_PTR, (Lin_17_AscLin_CoreConfigType*) &Lin_CoreConfigCore1, (Lin_17_AscLin_CoreConfigType*) &Lin_CoreConfigCore2, (Lin_17_AscLin_CoreConfigType*) &Lin_CoreConfigCore3, (Lin_17_AscLin_CoreConfigType*) &Lin_CoreConfigCore4, (Lin_17_AscLin_CoreConfigType*) &Lin_CoreConfigCore5 }</pre>

1.2.1.2 Member: Lin_17_AscLin_Channel_To_Core_SpecificArray [_<variant>] [LIN_17_ASCLIN_MAXIMUM_CHANNEL_CONFIGURED]

Table 33 Lin_17_AscLin_Channel_To_Core_SpecificArray [_<variant>]
[LIN_17_ASCLIN_MAXIMUM_CHANNEL_CONFIGURED]

Name	Lin_17_AscLin_Channel_To_Core_SpecificArray [_<variant>] [LIN_17_ASCLIN_MAXIMUM_CHANNEL_CONFIGURED]	
Type	uint8*	
Description	Pointer to the base of array which stores the index of each configured channel with in the allocated core.	
Verification method	<p>The generated structure member is present in the Lin_17_AscLin_Config [_<variant>] structure. The structure member is generated with base address of lookup table for channel to the allocated core mapping. (LIN_17_ASCLIN_MAXIMUM_CHANNEL_CONFIGURED ranges from 0 to maximum number of channels configured)</p> <p><Variant> indicates the name of the post-build variant. For a variant aware configuration the structure name is appended with the variant name. For variant unaware configuration <variant> is ignored.</p>	
Example(s)	Action	Generated output
	<ul style="list-style-type: none"> Configure channel0 allocated to ASCLIN1 Configure channel1 allocated to ASCLIN11 (variant-unaware) 	<pre>{ & Lin_17_AscLin_Channel_To_Core_SpecificArray[0] }</pre>
	<ul style="list-style-type: none"> Configure channel0 is allocated to ASCLIN1 Configure channel1 allocated to ASCLIN11 (variant-aware. Variant name is 'Petrol') 	<pre>{ & Lin_17_AscLin_Channel_To_Core_SpecificArray_Petrol[0] }</pre>

1.2.1.3 Member: Lin_17_AscLin_Hw_To_ChannelIdMap[_<variant>] [LIN_17_ASCLIN_MAX_HW_UNIT]

Table 34 Lin_17_AscLin_Hw_To_ChannelIdMap[_<variant>] [LIN_17_ASCLIN_MAX_HW_UNIT]

Name	Lin_17_AscLin_Hw_To_ChannelIdMap[_<variant>] [LIN_17_ASCLIN_MAX_HW_UNIT]	
Type	uint8*	
Description	Pointer to the base of array which stores the mapping of ASCLIN hardware module configured for the channel.	
Verification method	<p>The generated structure member is present in the Lin_17_AscLin_Config[_<variant>] structure. The structure member is generated with base address of lookup table for channel to the ASCLIN hardware mapping. (LIN_17_ASCLIN_MAX_HW_UNIT ranges from 0 to total number of Asclin hardware modules configured)</p> <p><Variant> indicates the name of the post-build variant. For a variant aware configuration the structure name is appended with the variant name. For variant unaware configuration <variant> is ignored.</p>	
Example(s)	Action	Generated output
	<ul style="list-style-type: none"> Configure channel0 allocated to ASCLIN1 Configure channel1 allocated to ASCLIN11 (variant-unaware) 	<pre>{ &Lin_17_AscLin_Hw_To_ChannelIdMap[0] }</pre>
	<ul style="list-style-type: none"> Configure channel0 is allocated to ASCLIN1 Configure channel1 allocated to ASCLIN11 (variant-aware. Variant name is 'Petrol') 	<pre>{ &Lin_17_AscLin_Hw_To_ChannelIdMap_Petrol[0] }</pre>

1.2.1.4 Member: TotalChannel

Table 35 TotalChannel

Name	TotalChannel	
Type	uint8	
Description	Indicates the number of the channels configured	
Verification method	The structure member is generated as total number of channels configured	
Example(s)	Action	Generated output
	Configure 3 LIN channels	3U
	Configure 12 LIN channels	12U

1.2.2 Structure: Lin_CoreConfigCore[x] [_<variant>]

Table 36 Lin_CoreConfigCore[x] [_<variant>]

Name	Lin_CoreConfigCore[x] [_<variant>]	
Type	Lin_17_AscLin_CoreConfigType	
Description	Configuration structure of LIN driver for configured core referenced in root configuration structure. (x ranges from 0 to 5)	
Verification method	The generated file has this structure if core is configured and at least one channel is allocated to it. <Variant> indicates the name of the post-build variant. For a variant aware configuration the structure name is appended with the variant name. For variant unaware configuration <variant> is ignored.	
Example(s)	Action	Generated output
	<ul style="list-style-type: none"> Configure 3 LIN channels Set Resource MMasterCore as CORE5. Allocated all the channels to Core0 (variant-unaware)	<pre>static const Lin_17_AscLin_CoreConfigType Lin_CoreConfigCore0 = { 3U, (const Lin_17_AscLin_ChannelType*)&Lin_kChannelConfigurationCore 0[0] };</pre>
Example(s)	<ul style="list-style-type: none"> Configure 4 LIN channels. Set Resource MMasterCore as CORE5. Allocate 1 channel to Core0 and 2 channels to Core 5 (variant-aware. Variant name is 'Petrol')	<pre>static const Lin_17_AscLin_CoreConfigType Lin_CoreConfigCore0_Petrol = { 1U, (const Lin_17_AscLin_ChannelType*)&Lin_kChannelConfigurationCore 0_Petrol[0] }; static const Lin_17_AscLin_CoreConfigType Lin_CoreConfigCore5_Petrol = { 3U, /* Two channel allocated to Core5, the channel not allocated to any core is allocated to the master core */ (const Lin_17_AscLin_ChannelType*)&Lin_kChannelConfigurationCore 5_Petrol[0] };</pre>

1.2.2.1 Member: MaxCoreChannels

Table 37 MaxCoreChannels

Name	MaxCoreChannels	
Type	uint8	
Description	Indicates the maximum number of channels allocated to a core	
Verification method	The structure member is generated as maximum channels used for the core.	
Example(s)	Action	Generated output
	Configure the Core 0 with maximum channels 1	1U
	Configure the Core 0 with maximum channels 3	3U

1.2.2.2 Member: Lin_kChannelConfigurationCore[x][_<variant>][TotalChannel]

Table 38 Lin_kChannelConfigurationCore[x][_<variant>][TotalChannel]

Name	Lin_kChannelConfigurationCore[x][_<variant>][TotalChannel]	
Type	Lin_17_AscLin_ChannelType	
Description	Pointer to the base of array which stores the data of each channel configured for a core. (x ranges from 0 to 5)	
Verification method	<p>The generated structure member is present in the Lin_CoreConfigCore[x][_<variant>] structure. The structure member is generated with base address of array which stores data of the configured channel for a core. (TotalChannel ranges from 0 to total number of channels configured)</p> <p><Variant> indicates the name of the post-build variant. For a variant aware configuration the structure name is appended with the variant name. For variant unaware configuration <variant> is ignored.</p>	
Example(s)	Action	Generated output
	Configure 1 LIN channel assigned to hardware module ASCLIN0 for Core 0 (variant-unaware)	<pre>/*Pointer to channel configuration*/ &Lin_kChannelConfigurationCore0[0]</pre>
	Configure 1 LIN channel assigned to hardware module ASCLIN0 for Core 2 (variant-aware. Variant name is 'Petrol')	<pre>/*Pointer to channel configuration*/ &Lin_kChannelConfigurationCore2_Petrol[0]</pre>

1.2.3 Structure:

Lin_kChannelConfigurationCore[x][_<variant>][TotalChannel]

Table 39 Lin_kChannelConfigurationCore [x][_<variant>][TotalChannel]

Name	Lin_kChannelConfigurationCore[x][_<variant>][TotalChannel]	
Type	Lin_17_AscLin_ChannelType	
Description	Configuration structure of LIN driver for configured channels which will be referenced in root configuration structure. (x value ranges from 0 to 5 and TotalChannel value ranges from 0 to the total number of channels configured)	
Verification method	<p>The generated file has this structure if at least one channel is configured.</p> <p><Variant> indicates the name of the post-build variant. For a variant aware configuration the structure name is appended with the variant name. For variant unaware configuration <variant> is ignored.</p> <ul style="list-style-type: none"> If LinAutoCalcBaudParams 'False' then baudrate parameters generates with value configured in parameter <ul style="list-style-type: none"> 'LinGlobalConfig/LinChannel/ LinChannelBaudNumerator' for numerator 'LinGlobalConfig/LinChannel/ LinChannelBaudDenominator' for denominator 'LinGlobalConfig/LinChannel/ LinChannelBaudPreScalar' for prescalar 'LinGlobalConfig/LinChannel/ LinInterByteSpace' for interbyte or response space. WakeupPrescalar and WakeuploclrDepth depends upon $f^{ASCLINF}$ or $f^{ASCLINS}$ which remains same across devices. The values for WakeupPrescalar and WakeuploclrDepth is fixed. If LinAutoCalcBaudParams is set 'True' then value is calculated based on input frequency and baud rate value. Baudrate parameters are generated as per following formula: $f^{PD} = f^A / (\text{ChanBaudRatePrescalar} + 1)$ $f^{OVS} = f^{PD} * (\text{ChanBaudRateNumerator} / \text{ChanBaudRateDenominator})$ $f^{SHIFT} (\text{Baud Rate}) = f^{OVS} / (\text{LinChanBaudOverSampling} + 1)$ <p>$f^{ASCLINF}$ or $f^{ASCLINS}$ is used as input clock frequency (f^A).</p>	
Example(s)	Action	Generated output
	<ul style="list-style-type: none"> Configure LIN channel 0 assigned to ASCLIN1 and channel1 assigned to ASCLIN0 for Core 0 Set LinCsrClksel as ASCLINS Input frequency $f^{ASCLINS}$ set as 2.0E7 Set LinChannelBaudRate as 19200 for channel0 and channel1 as 10000 Set LinAutoCalcBaudParams is true 	<pre>static const Lin_17_AscLin_ChannelType Lin_kChannelConfigurationCore0[2] = { /* LIN Channel ID: 0 Configuration */ { /* BaudRate : 19200 Hz */ { 384U, /* BRG.NUMERATOR value */ 1000U, /* BRG.DENOMINATOR value */ 24U, /* BITCON.PRESCALAR value*/ 46U, /*Prescalar value for wakeup*/ </pre>

<ul style="list-style-type: none"> • Set LinInterByteSpace as 0 for channel0 and 4 for channel1 • The value of WakeupPrescalar = 46 and WakeupIocrDepth = 63 • Set wakeup support as disabled • The Rx alternate pin is select G • The LogicalId is 0 (variant-unaware) • The channel node type is Slave. 	<pre> 0U, /*Interbyte or response space*/ 63U /*IOCR DEPTH for wakeup */ }, 0xFFU, /* EcuM Wakeup Source ID */ LIN_17_ASCLIN1, /* Hw Module used */ LIN_17_ASCLIN_CHANNEL_WAKEUP_DISABLED, /* Wakeup Support */ LIN_17_ASCLIN_SELECT_G /*Rx alternate pin select */ 0U /*Logical channel Id */ LIN_17_ASCLIN_SLAVE_NODE /*Channel node type */ }, /* LIN Channel ID: 1 Configuration */ { /* BaudRate : 10000 Hz */ { 8U, /* BRG.NUMERATOR value */ 1000U, /* BRG.DENOMINATOR value*/ 0U, /*BITCON.PRESCALAR value */ 46U, /*Prescalar value for wakeup*/ 4U, /*Interbyte or response space*/ 63U /*IOCR DEPTH for wakeup */ }, 0xFFU, /* EcuM Wakeup Source ID */ LIN_17_ASCLIN0, /* Hw Module used */ LIN_17_ASCLIN_CHANNEL_WAKEUP_DISABLED, /* Wakeup Support */ LIN_17_ASCLIN_SELECT_A /*Rx alternate pin select */ 0U /*Logical channel Id */ LIN_17_ASCLIN_SLAVE_NODE /*Channel node type */ } } </pre>
<ul style="list-style-type: none"> • Configure 1 LIN channel assigned to ASCLIN1 for Core 0 	<pre> static const Lin_17_AscLin_ChannelType Lin_kChannelConfigurationCore0_Petrol[1] = </pre>

<ul style="list-style-type: none"> • Set LinCsrClksel as ASCLINS • Input frequency $f^{ASCLINS}$ set as 2.0E7 • Set LinChannelBaudRate as 19200 • Set LinInterByteSpace as 0 • The value of WakeupPrescaler = 46 and WakeupIocrDepth = 63 • Set wakeup support as disabled • LinAutoCalcBaudParams is true and the Rx alternate pin select G(variant-aware). • The LogicalId is 0 (Variant name is 'Petrol') • The LogicalId is 0 (Variant name is 'Petrol') • The channel node type is Master 	<pre> { /* LIN Channel ID: 0 Configuration */ { /* BaudRate : 19200 Hz */ { 384U, /* BRG.NUMERATOR value */ 1000U, /* BRG.DENOMINATOR value*/ 24U, /* BITCON.PRESCALAR value*/ 46U, /*Prescaler value for wakeup*/ 0U, /*Interbyte or response space*/ 63U /*IOCR DEPTH for wakeup */ }, 0xFFU, /* EcuM Wakeup Source ID */ LIN_17_ASCLIN1, /* Hw Module used */ LIN_17_ASCLIN_CHANNEL_WAKEUP_DISABLED, /* Wakeup Support */ LIN_17_ASCLIN_SELECT_G /*Rx alternate pin select */ 0U /*Logical channel Id */ LIN_17_ASCLIN_MASTER_NODE /*Channel node type */ } } </pre>
<ul style="list-style-type: none"> • Configure 1 LIN channel assigned to ASCLIN0 for Core 2 • Wakeup support is enabled • LinAutoCalcBaudParams is false • Set LinChannelBaudNumerator as 10 • Set LinChannelBaudDenominator as 100 • Set LinChannelBaudPreScalar as 24 • The value of WakeupPrescaler = 46 and WakeupIocrDepth = 63 • Rx alternate pin select A 	<pre> static const Lin_17_AscLin_ChannelType Lin_kChannelConfigurationCore2[1] = { /* LIN Channel ID: 0 Configuration */ { /* BaudRate : 5000 Hz */ { 10U, /* BRG.NUMERATOR value */ 100U, /* BRG.DENOMINATOR value*/ 24U, /* BITCON.PRESCALAR value*/ 46U, /*Prescaler value for wakeup*/ 0U, /*Interbyte or response space*/ 63U /*IOCR DEPTH for wakeup */ }, 0xFFU, /* EcuM Wakeup Source ID */ LIN_17_ASCLIN0, /* Hw Module used */ </pre>

<ul style="list-style-type: none"> The LogicalId is 0 (variant-unaware) The channel node type is Master 	<pre> LIN_17_ASCLIN_CHANNEL_WAKEUP_ENABLED, /* Wakeup Support */ LIN_17_ASCLIN_SELECT_A /*Rx alternate pin select */ 0U /*Logical channel Id */ LIN_17_ASCLIN_MASTER_NODE /*Channel node type */ } } </pre>
---	---

1.2.3.1 Member: WakeupSourceId

Table 40 WakeupSourceId

Name	WakeupSourceId	
Type	EcuM_WakeupSourceType	
Description	Indicates the Source Id of the EcuM wakeup	
Verification method	<p>The structure member is generated as 0xFF if 'LinGlobalConfig/LinChannel/LinChannelEcuMWakeupSource' is set to False.</p> <p>If LinGlobalConfig/LinChannel/ LinChannelEcuMWakeupSource is set to true, the member is generated with the value of EcuMWakeupSourceId referenced using LinChannelWakeupInfo of that channel.</p>	
Example(s)	Action	Generated output
	Configure the LIN channel with LinChannelWakeupSupport is False	<pre> { 0xFF } </pre>
	<ul style="list-style-type: none"> Configure the LIN channel with LinChannelWakeupSupport is True EcuMWakeupSourceId referenced by LinChannelWakeupInfo is 3 	<pre> { 0x3 } </pre>

1.2.3.2 Member: HwModule

Table 41 HwModule

Name	HwModule
Type	uint8
Description	Indicates the ASCLIN hardware associated with the configured channel

Verification method	The structure member is generated as ASCLIN hardware module used for the channel.	
Example(s)	Action	Generated output
	Configure LIN channel 0 to ASCLIN hardware 0.	LIN_17_ASCLIN0
	Configure LIN channel 0 to ASCLIN hardware 11.	LIN_17_ASCLIN11

1.2.3.3 Member: Wakeup

Table 42 Wakeup

Name	Wakeup	
Type	uint8	
Description	Indicates if the ASCLIN hardware supports wakeup functionality.	
Verification method	The structure member is generated as LIN_17_ASCLIN_CHANNEL_WAKEUP_ENABLED if 'LinGlobalConfig/LinChannel / LinChannelWakeupSupport' is set to True else it is generated as LIN_17_ASCLIN_CHANNEL_WAKEUP_DISABLED.	
Example(s)	Action	Generated output
	Configure an LIN channel with LinChannelWakeupSupport = True	LIN_17_ASCLIN_CHANNEL_WAKEUP_ENABLED
	Configure an LIN channel with LinChannelWakeupSupport = False	LIN_17_ASCLIN_CHANNEL_WAKEUP_DISABLED

1.2.3.4 Member: RxAlternatePinSelect

Table 43 RxAlternatePinSelect

Name	RxAlternatePinSelect	
Type	uint8	
Description	Indicates the alternate input for the RX signal for the given Lin channel.	
Verification method	The structure member is generated as RX signal configured in LinGlobalConfig/LinChannel/ LinRxAlternateInputSignal parameter	
Example(s)	Action	Generated output
	Configure an LIN channel with LinRxAlternateInputSignal = SELECT_A_PORT15_PIN1	LIN_17_ASCLIN_SELECT_A

Configure an LIN channel with LinRxAlternateInputSignal = SELECT_G_PORT02_PIN3	LIN_17_ASCLIN_SELECT_G
--	------------------------

1.2.3.5 Member: LogicalId

Table 44 LogicalId

Name	LogicalId	
Type	uint8	
Description	Indicates the logical channel Id for the given Lin channel.	
Verification method	The structure member is generated as logicalId for a channel.	
Example(s)	Action	Generated output
	Configure the first LIN channel for Core 0 with logicalId	1U
	Configure the second LIN channel for Core 5 with logicalId	2U

1.2.3.6 Member: Node

Table 45 Node

Name	Node	
Type	uint8	
Description	Indicates the node type for the given Lin channel i.e. Master or Slave. This is applicable only in the AS440. For AS 422, by default the type of the node is Master.	
Verification method	The structure member is generated as node type for a channel.	
Example(s)	Action	Generated output
	Configure the first LIN channel for Core 0 with node type as master	LIN_17_ASCLIN_MASTER_NODE
	Configure the second LIN channel for Core 5 with node type as slave	LIN_17_ASCLIN_SLAVE_NODE

1.2.4 Structure: Lin_17_AscLin_Hw_To_ChannelIdMap[_<variant>] [LIN_17_ASCLIN_MAX_HW_UNIT]

Table 46 Lin_17_AscLin_Hw_To_ChannelIdMap[_<variant>] [LIN_17_ASCLIN_MAX_HW_UNIT]

	<pre> 0xFFU, 0xFFU, 0xFFU, 0xFFU }; </pre>
--	--

1.2.5 Structure: Lin_17_AscLin_Channel_To_Core_SpecificArray [_<variant>] [LIN_17_ASCLIN_MAXIMUM_CHANNEL_CONFIGURED]

Table 47 Lin_17_AscLin_Channel_To_Core_SpecificArray [_<variant>] [LIN_17_ASCLIN_MAXIMUM_CHANNEL_CONFIGURED]

Name	Lin_17_AscLin_Channel_To_Core_SpecificArray [_<variant>] [LIN_17_ASCLIN_MAXIMUM_CHANNEL_CONFIGURED]	
Type	uint8	
Description	Array of channel specific data, which stores the mapping of LIN channels configured to a specific core.	
Verification method	<p>The generated file has this structure if at least one channel is associated to one Core. (LIN_17_ASCLIN_MAXIMUM_CHANNEL_CONFIGURED ranges from 0 to maximum number of channels configured)</p> <p><Variant> indicates the name of the post-build variant. For a variant aware configuration the structure name is appended with the variant name. For variant unaware configuration <variant> is ignored.</p>	
Example(s)	Action	Generated output
	<ul style="list-style-type: none"> Configure LIN channel0 associated to Core 0 LIN channel1 associated to Core 5 LIN channel2 associated Core 5 LIN channel3 associated Core 5 (variant-unaware) 	<pre> static const uint8 Lin_17_AscLin_Channel_To_Core_SpecificArray [LIN_17_ASCLIN_MAXIMUM_CHANNEL_CONFIGURED] = { 0x0U, /*0 index belongs to Core0*/ 0x0U, /*0 index belongs to Core5*/ 0x1U, /*1 index belongs to Core5*/ 0x2U /*2 index belongs to Core5*/ }; </pre>
	<ul style="list-style-type: none"> Configure LIN channel0 associated to Core 0 (variant-aware. Variant name is 'Petrol') 	<pre> static const uint8 Lin_17_AscLin_Channel_To_Core_SpecificArray_Petrol [LIN_17_ASCLIN_MAXIMUM_CHANNEL_CONFIGURED] = { 0x0U, /*0 index belongs to Core0*/ }; </pre>

1.3 File: Lin_17_AscLin[_<variant>]_PBcfg.h

The generated header file contains the declaration of the root configuration structure. Post-build time configuration mechanism allows configurable functionality of LIN driver that is deployed as object code. The file is generated in 'inc' folder.

1.3.1 Structure: Lin_17_AscLin_Config[_<variant>]

Table 1 Lin_17_AscLin_Config[_<variant>]

Name	Lin_17_AscLin_Config[_<variant>]	
Type	Lin_17_AscLin_ConfigType	
Description	Declaration of root configuration structure of LIN driver which will be used during initialization.	
Verification method	The generated structure is present in Lin_17_AscLin[_<variant>]_PBcfg.h file. The <variant> indicates the name of the post-build variant. For a variant-aware configuration the structure name is appended with the variant name. For variant-unaware configuration <variant> is ignored.	
Example(s)	Action	Generated output
	Configure atleast one LIN channel and generate (variant-unaware)	extern const Lin_17_AscLin_ConfigType Lin_17_AscLin_Config;
	Configure atleast one LIN channel and generate (variant-aware. Variant name is 'Petrol')	extern const Lin_17_AscLin_ConfigType Lin_17_AscLin_Config_Petrol;

Revision history

Revision history

Major changes since the last revision

Date	Version	Description
2023-05-26	2.0	<ul style="list-style-type: none"> Document Released Revision history updated.
2023-05-26	1.1	<ul style="list-style-type: none"> In section 1.1.23, description has been updated to change DEM to production error. In section 1.1.24, description has been updated to change Error to production error.
2020-12-09	1.0	<ul style="list-style-type: none"> Released
2020-12-09	0.1	<ul style="list-style-type: none"> Lin_17_AscLin driver chapter moved from MC-ISAR_TC3xx_Config_Verification_Manual_BASIC.pdf to this document Added LIN_17_ASCLIN_HW_ERROR_DEM_REPORT, LIN_17_ASCLIN_E_TIMEOUT, LIN_17_ASCLIN_MASTER_CH_PRESENT, LIN_17_ASCLIN_SLAVE_CH_PRESENT, LIN_17_ASCLIN_TIMEOUT_COUNT, LinConf_LinChannel_<LinChannelName> macros

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2023-05-26

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2023 Infineon Technologies AG.

All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

Doc_Number

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.