MCAL Configuration Verification Manual for CanTrcv_17_W9255

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 CanTrcv_17_W9255

Table of contents

CanTrcv_17_W9255 driver

This chapter describes the details of the configuration data generated from the CAN transceiver W9255 driver.

1.1 File: CanTrcv_17_W9255_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: CANTRCV_17_W9255_AR_RELEASE_MAJOR_VERSION

Table 1 CANTRCV_17_W9255_AR_RELEASE_MAJOR_VERSION

Name	CANTRCV_17_W9255_AR_RELEASE_MAJOR_VERSION	
Description	Major version number of AUTOSAR release on which the CanTrcv_17_W9255 implementation is based on.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/ArMajorVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate CanTrcv_17_W9255_Cfg.h file with ArMajorVersion 4	`#define CANTRCV_17_W9255_AR_RELEASE_MAJOR_VERSION (4U)`

1.1.2 Macro: CANTRCV_17_W9255_AR_RELEASE_MINOR_VERSION

Table 2 CANTRCV_17_W9255_AR_RELEASE_MINOR_VERSION

Name	CANTRCV_17_W9255_AR_RELEASE_MINOR_VERSION	
Description	Minor version number of AUTOSAR release on which the CanTrcv_17_W9255 implementation is based on.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/ArMinorVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate CanTrcv_17_W9255_Cfg.h file with ArMinorVersion 2	`#define CANTRCV_17_W9255_AR_RELEASE_MINOR_VERSION (2U)`

1.1.3 Macro: CANTRCV_17_W9255_AR_RELEASE_REVISION_VERSION

Table 3 CANTRCV_17_W9255_AR_RELEASE_REVISION_VERSION

Name	CANTRCV_17_W9255_AR_RELEASE_REVISION_VERSION	
Description	Revision version number of AUTOSAR release on which the CanTrcv_17_W9255 implementation is based on.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/ArPatchVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate CanTrcv_17_W9255_Cfg.h file with ArPatchVersion 2	`#define CANTRCV_17_W9255_AR_RELEASE_REVISION_VERSION (2U)`

1.1.4 Macro: CANTRCV_17_W9255_SW_MAJOR_VERSION

Table 4 CANTRCV_17_W9255_SW_MAJOR_VERSION

Name	CANTRCV_17_W9255_SW_MAJOR_VERSION	
Description	Major version number of the CanTrcv_17_W9255 module.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/SwMajorVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate CanTrcv_17_W9255_Cfg.h file with SwMajorVersion 10	`#define CANTRCV_17_W9255_SW_MAJOR_VERSION (10U)`

1.1.5 Macro: CANTRCV_17_W9255_SW_MINOR_VERSION

Table 5 CANTRCV_17_W9255_SW_MINOR_VERSION

Name	CANTRCV_17_W9255_SW_MINOR_VERSION	
Description	Minor version number of the CanTrcv_17_W9255 module.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/SwMinorVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate CanTrcv_17_W9255_Cfg.h file with SwMinorVersion 30	`#define CANTRCV_17_W9255_SW_MINOR_VERSION (30U)`

1.1.6 Macro: CANTRCV_17_W9255_SW_PATCH_VERSION

Table 6 CANTRCV_17_W9255_SW_PATCH_VERSION

Name	CANTRCV_17_W9255_SW_PATCH_VERSION	
Description	Patch version number of the CanTrcv_17_W9255 module.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/SwPatchVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate CanTrcv_17_W9255_Cfg.h file with SwPatchVersion 0	`#define CANTRCV_17_W9255_SW_PATCH_VERSION (0U)`

1.1.7 Macro: CANTRCV_17_W9255_DEV_ERROR_DETECT

Table 7 CANTRCV_17_W9255_DEV_ERROR_DETECT

Name	CANTRCV_17_W9255_DEV_ERROR_DETECT	
Description	Enables/disables the Development Error Detection.	
Verification method	The macro is generated as STD_ON if CanTrcvDevErrorDetect configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	CanTrcvDevErrorDetect = True	`#define CANTRCV_17_W9255_DEV_ERROR_DETECT (STD_ON)`
	CanTrcvDevErrorDetect = False	`#define CANTRCV_17_W9255_DEV_ERROR_DETECT (STD_OFF)`

1.1.8 Macro: CANTRCV_17_W9255_ RUNTIME_ERROR_DETECT

Table 8 CANTRCV_17_W9255_ RUNTIME_ERROR_DETECT

Name	CANTRCV_17_W9255_ RUNTIME_ERROR_DETECT	
Description	Enables/disables the runtime error detection and reporting. Note: This configuration macro is available only in AUTOSAR 440.	
Verification method	The macro is generated as STD_ON if CanTrcvRunTimeErrorDetect configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	CanTrcvRunTimeErrorDetect= True	`#define CANTRCV_17_W9255_ RUNTIME_ERROR_DETECT (STD_ON)`

	CanTrcvRunTimeErrorDetect = False	`#define CANTRCV_17_W9255_ RUNTIME_ERROR_DETECT (STD_OFF)`

1.1.9 Macro: CANTRCV_17_W9255_WAIT_TIME

Table 9 CANTRCV_17_W9255_WAIT_TIME

Name	CANTRCV_17_W9255_WAIT_TIME	
Description	Indicates wait time for transceiver mode changes	
Verification method	The macro is generated as a numeric value which corresponds to the value configured in 'CanTrcvGeneral/CanTrcvWaitTime/*[1]' in nanoseconds. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	CanTrcvWaitTime = 2e-5	`#define CANTRCV_17_W9255_WAIT_TIME (20000U)`

1.1.10 Macro: CANTRCV_17_W9255_GET_VERSION_INFO

Table 10 CANTRCV_17_W9255_GET_VERSION_INFO

Name	CANTRCV_17_W9255_GET_VERSION_INFO	
Description	Enables/disables CanTrcv_17_W9255_GetVersionInfo API	
Verification method	The macro is generated as STD_ON if CanTrcvGetVersionInfo configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	CanTrcvGetVersionInfo = True	`#define CANTRCV_17_W9255_GET_VERSION_INFO (STD_ON)`
	CanTrcvGetVersionInfo = False	`#define CANTRCV_17_W9255_GET_VERSION_INFO (STD_OFF)`

1.1.11 Macro: CANTRCV_17_W9255_SPI_COMM_RETRIES

Table 11 CANTRCV_17_W9255_SPI_COMM_RETRIES

Name	CANTRCV_17_W9255_SPI_COMM_RETRIES
Description	Indicates the maximum number of communication retries in case of a failed SPI communication.
Verification method	The macro is generated as a numeric value which corresponds to the value configured in 'CanTrcvConfigSet/CanTrcvSPICommRetries'.

Example(s)	Action	Generated output
	CanTrcvSPICommRetries = 0	`#define CANTRCV_17_W9255_SPI_COMM_RETRIES (0U)`
	CanTrcvSPICommRetries = 255	`#define CANTRCV_17_W9255_SPI_COMM_RETRIES (255U)`

1.1.12 Macro: CANTRCV_17_W9255_GENERAL_WAKE_UP_SUPPORT

Table 12 CANTRCV_17_W9255_GENERAL_WAKE_UP_SUPPORT

Name	CANTRCV_17_W9255_GENERAL_WAKE_UP_SUPPORT	
Description	Indicates whether wake up of CanTrcv_17_W9255 module is supported by polling or interrupt.	
Verification method	The macro is generated as CANTRCV_17_W9255_WAKE_UP_BY_INTERRUPT if CanTrcvWakeUpSupport configuration parameter is set to 'CANTRCV_17_W9255_WAKE_UP_BY_INTERRUPT' else the macro is generated as CANTRCV_17_W9255_WAKE_UP_BY_POLLING.	
Example(s)	**Action**	**Generated output**
	CanTrcvWakeUpSupport = CANTRCV_17_W9255_WAKE_UP_BY_INTERRUPT	`#define CANTRCV_17_W9255_GENERAL_WAKE_UP_SUPPORT (CANTRCV_17_W9255_WAKE_UP_BY_INTERRUPT)`
	CanTrcvWakeUpSupport = CANTRCV_17_W9255_WAKE_UP_BY_POLLING	`#define CANTRCV_17_W9255_GENERAL_WAKE_UP_SUPPORT (CANTRCV_17_W9255_WAKE_UP_BY_POLLING)`

1.1.13 Macro: CANTRCV_17_W9255_INSTANCE_ID

Table 13 CANTRCV_17_W9255_INSTANCE_ID

Name	CANTRCV_17_W9255_INSTANCE_ID	
Description	Instance ID of CanTrcv_17_W9255 module.	
Verification method	The macro is generated as a numeric value set in the configuration parameter 'CanTrcvGeneral/CanTrcvIndex'	
Example(s)	**Action**	**Generated output**
	Set CanTrcvIndex as 0	`#define CANTRCV_17_W9255_INSTANCE_ID (0U)`
	Set CanTrcvIndex as 2	`#define CANTRCV_17_W9255_INSTANCE_ID (2U)`

1.1.14 Macro: CanTrcv_17_W9255_CanTrcvConf_CanTrcvChannel_<channel name>

Table 14 CanTrcv_17_W9255_CanTrcvConf_CanTrcvChannel_<channel name>

Name	CanTrcv_17_W9255_CanTrcvConf_CanTrcvChannel_<channel name>	
Description	The macro is the symbolic name generated for the transceiver channel 'CanTrcvConfigSet/ CanTrcvChannel'	
Verification method	The macro is generated as a numeric value which is configured in 'CanTrcvConfigSet/ CanTrcvChannel/CanTrcvChannelId'. <channel name> is the symbolic name of the transceiver channel. *Note: This macro is present only for the channels which have the parameter 'CanTrcvChannel/CanTrcvChannelUsed' set to true.*	
Example(s)	**Action**	**Generated output**

	• Configure 4 transceiver channels (CanTrcvChannel_0 to CanTrcvChannel_3)
	• Configure CanTrcvChannel_1 and CanTrcvChannel_3 as used channels
		`#ifndef CanTrcv_17_W9255_CanTrcvConf_CanTrcvChannel_CanTrcvChannel_1`
		`#define CanTrcv_17_W9255_CanTrcvConf_CanTrcvChannel_CanTrcvChannel_1 (1U)`
		`#endif`
		
		`#ifndef CanTrcv_17_W9255_CanTrcvConf_CanTrcvChannel_CanTrcvChannel_3`
		`#define CanTrcv_17_W9255_CanTrcvConf_CanTrcvChannel_CanTrcvChannel_3 (3U)`
		`#endif`

	• Configure 4 transceiver channels (CanTrcvChannel_0 to CanTrcvChannel_3)
	• Configure CanTrcvChannel_0 and CanTrcvChannel_2 as used channels
		`#ifndef CanTrcv_17_W9255_CanTrcvConf_CanTrcvChannel_CanTrcvChannel_0`
		`#define CanTrcv_17_W9255_CanTrcvConf_CanTrcvChannel_CanTrcvChannel_0 (0U)`
		`#endif`
		
		`#ifndef CanTrcv_17_W9255_CanTrcvConf_CanTrcvChannel_CanTrcvChannel_2`
		`#define CanTrcv_17_W9255_CanTrcvConf_CanTrcvChannel_CanTrcvChannel_2 (2U)`

		#endif

1.1.15 Macro: CANTRCV_17_W9255_CH_<x>_MAX_BAUDRATE_SUPPORT

Table 15 CANTRCV_17_W9255_CH_<x>_MAX_BAUDRATE_SUPPORT

Name	CANTRCV_17_W9255_CH_<x>_MAX_BAUDRATE_SUPPORT	
Description	Indicates the baudrate configured for channel<x>.	
Verification method	The macro is generated as a numeric value which is configured in 'CanTrcvMaxBaudrate' for every channel. *Note: This macro is present only for the channels which have the parameter 'CanTrcvChannel/CanTrcvChannelUsed' set to true.*	
Example(s)	**Action**	**Generated output**

	• Configure 4 transceiver channels (CanTrcvChannel_0 to CanTrcvChannel_3)
	• Configure CanTrcvChannel_1 and CanTrcvChannel_3 as used channels
	• Configure parameter 'CanTrcvMaxBaudrate' for all the channels with the value 5000
		`#define CANTRCV_17_W9255_CH_1_MAX_BAUDRATE_SUPPORT (5000U)`
		
		`#define CANTRCV_17_W9255_CH_3_MAX_BAUDRATE_SUPPORT (5000U)`

	• Configure 4 transceiver channels (CanTrcvChannel_0 to CanTrcvChannel_3)
	• Configure CanTrcvChannel_0 and CanTrcvChannel_2 as used channels
	• Configure parameter 'CanTrcvMaxBaudrate' for all the channels with the value 1000
		`#define CANTRCV_17_W9255_CH_0_MAX_BAUDRATE_SUPPORT (1000U)`
		
		`#define CANTRCV_17_W9255_CH_2_MAX_BAUDRATE_SUPPORT (1000U)`

1.1.16 Macro: CANTRCV_17_W9255_ICU_CHANNEL_CONFIGURED_<x>

Table 16 CANTRCV_17_W9255_ICU_CHANNEL_CONFIGURED_<x>

Name	CANTRCV_17_W9255_ICU_CHANNEL_CONFIGURED_<x>

Description	The macro is the symbolic name generated for the configuration parameter 'CanTrcvConfigSet/ CanTrcvChannel/ CanTrcvIcuChannelRef' for channel <x>.
Verification method	The macro is generated as a symbolic name which corresponds to the ICU channel configured for transceiver channel <x>. *Note: This macro is present only if the wake-up is supported by the interrupt mode. Besides, the macro is present only if the parameters 'CanTrcvChannel/CanTrcvChannelUsed' and 'CanTrcvChannel/ CanTrcvWakeupByBusUsed' are enabled for the respective channels.*

Example(s)	**Action**	**Generated output**

	• Configure 4 transceiver channels (CanTrcvChannel_0 to CanTrcvChannel_3)
	• Configure 'CanTrcvWakeUpSupport' as CANTRCV_17_W9255_WAKE_UP_BY_INTERRUPT
	• Enable 'CanTrcvChannelUsed' for channels CanTrcvChannel_1 and CanTrcvChannel_2
	• Enable 'CanTrcvWakeupByBusUsed' for channels CanTrcvChannel_1 and CanTrcvChannel_2
		`#define CANTRCV_17_W9255_ICU_CHANNEL_CONFIGURED_1 (IcuConf_IcuChannel_IcuChannel_2)`
		
		`#define CANTRCV_17_W9255_ICU_CHANNEL_CONFIGURED_2 (IcuConf_IcuChannel_IcuChannel_1)`

	• Configure 4 transceiver channels (CanTrcvChannel_0 to CanTrcvChannel_3) • Configure 'CanTrcvWakeUpSupport' as CANTRCV_17_W9255_WAKE_UP_BY_INTERRUPT • Enable 'CanTrcvChannelUsed' for channels CanTrcvChannel_0 and CanTrcvChannel_1 • Enable 'CanTrcvWakeupByBusUsed' for channel CanTrcvChannel_1 and disable it for channel CanTrcvChannel_0	`#define CANTRCV_17_W9255_ICU_CHANNEL_CONFIGURED_1 (IcuConf_IcuChannel_IcuChannel_2)`

1.1.17 Macro: CANTRCV_17_W9255_CHANNELS_USED

Table 17 CANTRCV_17_W9255_CHANNELS_USED

Name	CANTRCV_17_W9255_CHANNELS_USED	
Description	Indicates the total number of enabled channels.	
Verification method	The macro is generated as a numeric value which corresponds to the number of channels in the container 'CanTrcvConfigSet/CanTrcvChannel' which have the parameter 'CanTrcvChannelUsed' set to 'True'.	
Example(s)	**Action**	**Generated output**

	• Configure 4 transceiver channels. • Enable 'CanTrcvChannelUsed' for any 2 configured channels	`#define CANTRCV_17_W9255_CHANNELS_USED (2U)`
	• Configure 4 transceiver channels. • Enable 'CanTrcvChannelUsed' for all the configured channels	`#define CANTRCV_17_W9255_CHANNELS_USED (4U)`

1.1.18 Macro: CANTRCV_17_W9255_CHANNELS_CONFIGURED

Table 18 CANTRCV_17_W9255_CHANNELS_CONFIGURED

Name	CANTRCV_17_W9255_CHANNELS_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 channels in the container 'CanTrcvConfigSet/CanTrcvChannel'.	
Example(s)	**Action**	**Generated output**
	Configure 4 transceiver channels. (CanTrcvChannel_0 to CanTrcvChannel_3)	`#define CANTRCV_17_W9255_CHANNELS_CONFIGURED (4U)`
	Configure 8 transceiver channels. (CanTrcvChannel_0 to CanTrcvChannel_7)	`#define CANTRCV_17_W9255_CHANNELS_CONFIGURED (8U)`

1.1.19 Macro: CANTRCV_17_W9255_PN_CONFIGURED_CHANNELS

Table 19 CANTRCV_17_W9255_PN_CONFIGURED_CHANNELS

Name	CANTRCV_17_W9255_PN_CONFIGURED_CHANNELS	
Description	Indicates the total number of PN enabled channels.	
Verification method	The macro is generated as a numeric value which corresponds to the number of channels in the container 'CanTrcvConfigSet/CanTrcvChannel' which have the parameter 'CanTrcvPartialNetwork/CanTrcvPnEnabled' enabled.	
Example(s)	**Action**	**Generated output**
	• Configure 4 transceiver channels. • Enable 'CanTrcvPnEnabled' for	`#define CANTRCV_17_W9255_PN_CONFIGURED_CHANNELS (2U)`

	any 2 configured channels.	
	• Configure 4 transceiver channels. • Disable 'CanTrcvPnEnabled' for all the configured channels.	`#define CANTRCV_17_W9255_PN_CONFIGURED_CHANNELS (0U)`

1.2 File: CanTrcv_17_W9255_Cfg.c

The generated header file contains all pre compile configuration parameters. The file is generated in 'src' folder.

1.2.1 Structure: CanTrcv_17_W9255_PNConfig [CANTRCV_17_W9255_PN_CONFIGURED_CHANNELS]

Table 20 CanTrcv_17_W9255_PNConfig[CANTRCV_17_W9255_PN_CONFIGURED_CHANNELS]

Name	CanTrcv_17_W9255_PNConfig[CANTRCV_17_W9255_PN_CONFIGURED_CHANNELS]	
Type	CanTrcv_17_W9255_PNConfigType	
Description	Configuration structure of PN of the PN enabled channels	
Verification method	The generated structure is present in CanTrcv_17_W9255_Cfg.c file. The generated file has this structure if at least one channel is PN enabled. This structure contains PN configuration of all the PN enabled channels. The size of the structure depends on the number of PN enabled channels. *Note: The PN functionality can be enabled/disabled by the 'CanTrcvPnEnabled' configuration parameter.*	
Example(s)	**Action**	**Generated output**
	Configure 3 channels and enable PN for channel Id 2. Configure the parameters in the containers CanTrcvPartialNetwork and CanTrcvPnFrameDataMaskSpec for channel Id 2. Configure CanTrcvPnFrameDlc=3.	`const CanTrcv_17_W9255_PNConfigType CanTrcv_17_W9255_PNConfig[CANTRCV_17_W9255_PN_CONFIGURED_CHANNELS] = { /* PN configuration of CAN Transceiver Channel Id 2 */ { /* Pointer to baudrate, Id, Id mask of the PN frame */ CanTrcv_17_W9255_Ch2PnIdInfo, /* Pointer to data masks of the PN frame */ CanTrcv_17_W9255_Ch2PnDataInfo, /* Data Length Code of the WUF */ 3U } };`

1.2.1.1 Member: CanTrcvPnIdInfoPtr

Table 21 CanTrcvPnIdInfoPtr

Name	CanTrcvPnIdInfoPtr	
Type	uint16*	
Description	Pointer to the base of array which stores commands to configure baudrate, Id, Id mask of the PN frame of channel<x>.	
Verification method	The structure member is generated with base address of array which stores the PN baudrate, CAN ID and ID mask of the PN frame of the respective channel<x> in the form of commands.	
Example(s)	**Action**	**Generated output**
	For channel Id 1, configure CanTrcvBaudRate, CanTrcvPnFrameCanId and CanTrcvPnFrameCanIdMask parameters	`CanTrcv_17_W9255_Ch1PnIdInfo`
	For channel Id 2, configure CanTrcvBaudRate, CanTrcvPnFrameCanId and CanTrcvPnFrameCanIdMask parameters	`CanTrcv_17_W9255_Ch2PnIdInfo`

1.2.1.2 Member: CanTrcvPnDataInfoPtr

Table 22 CanTrcvPnDataInfoPtr

Name	CanTrcvPnDataInfoPtr	
Type	uint16*	
Description	Pointer to the base of array which stores commands to configure data masks of the PN frame of the channel<x>.	
Verification method	The structure member is generated with base address of array which stores the PN data masks of the PN frame of the respective channel<x> in the form of commands.	
Example(s)	**Action**	**Generated output**
	For channel Id 1, configure CanTrcvPnFrameDlc parameter and parameters in CanTrcvPnFrameDataMaskSpec container	`CanTrcv_17_W9255_Ch1PnDataInfo`
	For channel Id 2, configure CanTrcvPnFrameDlc parameter and parameters in CanTrcvPnFrameDataMaskSpec container	`CanTrcv_17_W9255_Ch2PnDataInfo`

1.2.1.3 Member: CanTrcvPnDlc

Table 23 CanTrcvPnDlc

Name	CanTrcvPnDlc	
Type	uint16	
Description	DLC of the PN frame of the respective channel	
Verification method	The structure member is generated as a numeric value configured in the CanTrcvPnFrameDlc configuration parameter for the respective transceiver channel.	
Example(s)	**Action**	**Generated output**
	Configure CanTrcvPnFrameDlc=7	`7U`
	Configure CanTrcvPnFrameDlc=3	`3U`

1.2.2 Structure: CanTrcv_17_W9255_ChannelConfig [CANTRCV_17_W9255_CHANNELS_USED]

Table 24 CanTrcv_17_W9255_ChannelConfig [CANTRCV_17_W9255_CHANNELS_USED]

Name	CanTrcv_17_W9255_ChannelConfig [CANTRCV_17_W9255_CHANNELS_USED]	
Type	CanTrcv_17_W9255_ChannelConfigType	
Description	Channel configuration structure of CanTrcv_17_W9255 driver	
Verification method	The generated structure is present in CanTrcv_17_W9255_Cfg.c file. The size of the structure depends on the number of channels used. *Note: A channel can be enabled/disabled by the 'CanTrcvChannelUsed' configuration parameter.*	
Example(s)	**Action**	**Generated output**

	Configure 4 channels and enable channel Id 2 among them.	`const CanTrcv_17_W9255_ChannelConfigType CanTrcv_17_W9255_ChannelConfig[CANTRCV_17_W9255_CHANNELS_USED] =`
		`{`
		`  /* CanTransceiver Channel 2 Specific Information */`
		`  {`
		`  /* CAN Transceiver state after driver initialization */`

	/* Command to write to MODE_CTRL register when the requested mode is STANDBY */
	0x8102U,
	
	/* CanTrcvWakeupSource reference */
	1U,
	
	/* CanTrcvPorWakeupSource reference */
	1U,
	
	/* CanTrcvSyserrWakeupSource reference */
	0U,
	
	/* CAN Transceiver Channel Id */
	2U,
	
	/* Sequence Id used */
	2U,
	
	/* Spi Channel Id used */
	2U,
	
	/*
	Wake up by bus status
	- if STD_ON, Bus is used
	- if STD_OFF, Bus is not used
	*/
	STD_ON
	}
	};

Configure 4 channels and enable channels with Ids 0, 1 and 2 among them.	const CanTrcv_17_W9255_ChannelConfigType CanTrcv_17_W9255_ChannelConfig[CANTRCV_17_W9255_CHANNELS_USED] = { /* CanTransceiver Channel 0 Specific Information */ { /* CAN Transceiver state after driver initialization */ /* Command to write to MODE_CTRL register when the requested mode is NORMAL */ 0x8108U,

```
    /* CanTrcvWakeupSource reference */
    CANTRCV_17_W9255_WAKEUP_SOURCE_NOT_CONFIGURED,

    /* CanTrcvPorWakeupSource reference */
    CANTRCV_17_W9255_WAKEUP_SOURCE_NOT_CONFIGURED,

    /* CanTrcvSyserrWakeupSource reference */
    CANTRCV_17_W9255_WAKEUP_SOURCE_NOT_CONFIGURED,

    /* CAN Transceiver Channel Id */
    0U,

    /* Sequence Id used */
    0U,

    /* Spi Channel Id used */
    0U,

    /*
    Wake up by bus status
    - if STD_ON, Bus is used
    - if STD_OFF, Bus is not used
    */
    STD_OFF
  },
  /* CanTransceiver Channel 1 Specific Information */
  {
    /* CAN Transceiver state after driver initialization */
    /* Command to write to MODE_CTRL register when the
requested mode is SLEEP */
    0x8101U,

    /* CanTrcvWakeupSource reference */
    0U,

    /* CanTrcvPorWakeupSource reference */
    0U,
```

```
    /* CanTrcvSyserrWakeupSource reference */
    1U,

    /* CAN Transceiver Channel Id */
    1U,

    /* Sequence Id used */
    1U,

    /* Spi Channel Id used */
    1U,

    /*
    Wake up by bus status
    - if STD_ON, Bus is used
    - if STD_OFF, Bus is not used
    */
    STD_ON
  },
  /* CanTransceiver Channel 2 Specific Information */
  {
    /* CAN Transceiver state after driver initialization */
    /* Command to write to MODE_CTRL register when the
requested mode is STANDBY */
    0x8102U,

    /* CanTrcvWakeupSource reference */
    1U,

    /* CanTrcvPorWakeupSource reference */
    1U,

    /* CanTrcvSyserrWakeupSource reference */
    0U,

    /* CAN Transceiver Channel Id */
    2U,

    /* Sequence Id used */
```

```
    2U,

    /* Spi Channel Id used */
    2U,

    /*
    Wake up by bus status
    - if STD_ON, Bus is used
    - if STD_OFF, Bus is not used
    */
    STD_ON
  }
};
```

1.2.2.1 Member: CanTrcvInitState

Table 25 CanTrcvInitState

Name	CanTrcvInitState	
Type	uint16	
Description	Indicates the mode of the transceiver channel after initialization.	
Verification method	The structure member is generated as a uint16 value which configures the MODE_CTRL register of the transceiver channel according to the mode configured in the CanTrcvInitState parameter.	
Example(s)	**Action**	**Generated output**
	Configure CanTrcvInitState parameter of Channel 1 as CANTRCV_17_W9255_OP_MODE_NORMAL	`0x8108U`
	Configure CanTrcvInitState parameter of Channel 1 as CANTRCV_17_W9255_OP_MODE_STANDBY	`0x8102U`
	Configure CanTrcvInitState parameter of Channel 1 as CANTRCV_17_W9255_OP_MODE_SLEEP	`0x8101U`

1.2.2.2 Member: CanTrcvWakeupSourceRef

Table 26 CanTrcvWakeupSourceRef

Name	CanTrcvWakeupSourceRef
Type	EcuM_WakeupSourceType
Description	Reference to the wakeup source of the channel.
Verification method	If CanTrcvWakeupByBusUsed configuration parameter is enabled, the structure member is generated with the value of EcuMWakeupSourceId referenced using CanTrcvWakeupSourceRef

	parameter. If CanTrcvWakeupByBusUsed configuration parameter is disabled, the structure member is generated as CANTRCV_17_W9255_WAKEUP_SOURCE_NOT_CONFIGURED. *Note: The configuration parameter CanTrcvWakeupSourceRef is not user configurable if CanTrcvWakeupByBusUsed configuration parameter is disabled.*	
Example(s)	**Action**	**Generated output**
	Enbale CanTrcvWakeupByBusUsed for Channel Id 1. Refer EcuMWakeupSourceId 2 in CanTrcvWakeupSourceRef configuration parameter of Channel Id 1.	`2U`
	Disable CanTrcvWakeupByBusUsed configuration parameter for Channel Id 2.	`CANTRCV_17_W9255_WAKEUP_SOURCE_NOT_CONFIGURED`

1.2.2.3 Member: CanTrcvPorWakeupSourceRef

Table 27 CanTrcvPorWakeupSourceRef

Name	CanTrcvPorWakeupSourceRef	
Type	EcuM_WakeupSourceType	
Description	Reference to the wakeup source of the channel in case of POR.	
Verification method	The structure member is generated with the value of EcuMWakeupSourceId referenced using CanTrcvPorWakeupSourceRef of that channel if referenced and is generated as CANTRCV_17_W9255_WAKEUP_SOURCE_NOT_CONFIGURED if not referenced.	
Example(s)	**Action**	**Generated output**
	Refer EcuMWakeupSourceId 12 in CanTrcvPorWakeupSourceRef configuration parameter of Channel Id 4	`12U`
	CanTrcvPorWakeupSourceRef is not referenced	`CANTRCV_17_W9255_WAKEUP_SOURCE_NOT_CONFIGURED`

1.2.2.4 Member: CanTrcvSyserrWakeupSourceRef

Table 28 CanTrcvSyserrWakeupSourceRef

Name	CanTrcvSyserrWakeupSourceRef
Type	EcuM_WakeupSourceType
Description	Reference to the wakeup source of the channel in case of SYSERR.

Verification method	The structure member is generated with the value of EcuMWakeupSourceId referenced using CanTrcvSyserrWakeupSourceRef of that channel if referenced and is generated as CANTRCV_17_W9255_WAKEUP_SOURCE_NOT_CONFIGURED if not referenced.	
Example(s)	**Action**	**Generated output**
	Refer EcuMWakeupSourceId 5 in CanTrcvSyserrWakeupSourceRef configuration parameter of Channel Id 3	`5U`
	CanTrcvSyserrWakeupSourceRef is not referenced	`CANTRCV_17_W9255_WAKEUP_SOURCE_NOT_CONFIGURED`

1.2.2.5 Member: CanTrcvChannelId

Table 29 CanTrcvChannelId

Name	CanTrcvChannelId	
Type	uint8	
Description	Channel Id of the channel.	
Verification method	The structure member is generated as the channel Id of the respective transceiver channel configured in the CanTrcvChannelId configuration parameter.	
Example(s)	**Action**	**Generated output**
	Add a channel instance CanTrcvChannel_1 in the CanTrcvChannel container and configure value 2 in the CanTrcvChannelId configuration parameter of that instance	`2U`
	Add a channel instance CanTrcvChannel_0 in the CanTrcvChannel container and configure value 0 in the CanTrcvChannelId configuration parameter of that instance.	`0U`

1.2.2.6 Member: CanTrcvSpiSequence

Table 30 CanTrcvSpiSequence

Name	CanTrcvSpiSequence
Type	Spi_SequenceType
Description	Spi Sequence Id used by the transceiver channel.
Verification method	The structure member is generated as the value of SPI sequence ID referenced by the instance of 'CanTrcvSpiSequenceName' configuration parameter. *Note: Only one SPI sequence is configured per transceiver channel.* *Note: The user should ensure that the physical hardware SPI channel and the transceiver channel are mapped appropriately.*

Example(s)	Action	Generated output
	Refer an instance in CanTrcvSpiSequenceName container to SpiSequenceId 3	`3U`
	Refer an instance in CanTrcvSpiSequenceName container to SpiSequenceId 6	`6U`

1.2.2.7 Member: CanTrcvSpiChannel

Table 31 CanTrcvSpiChannel

Name	CanTrcvSpiChannel	
Type	Spi_ChannelType	
Description	Spi Channel Id used by the transceiver channel.	
Verification method	The structure member is generated as the value of channel Id of the respective SPI channel linked to the respective SPI sequence referenced in the CanTrcvSpiSequenceName configuration parameter. *Note: Only one SPI sequence is configured per transceiver channel. This sequence consists of one job and this job consists of only one channel.*	
Example(s)	**Action**	**Generated output**
	Refer an instance in CanTrcvSpiSequenceName container to SpiSequenceId 3. Configure an SPI job and a channel with Id 3 to this sequence.	`3U`
	Refer an instance in CanTrcvSpiSequenceName container to SpiSequenceId 5. Configure an SPI job and a channel with Id 2 to this sequence.	`2U`

1.2.2.8 Member: CanTrcvWakeupByBusEnable

Table 32 CanTrcvWakeupByBusEnable

Name	CanTrcvWakeupByBusEnable	
Type	boolean	
Description	Wake-up by bus status of the channel.	
Verification method	The structure member is generated as STD_ON if the parameter CanTrcvWakeupByBusUsed is set true and STD_OFF if the parameter is set false.	
Example(s)	**Action**	**Generated output**
	Configure a transceiver channel with CanTrcvWakeupByBusUsed = True	`STD_ON`

	Configure a transceiver channel with CanTrcvWakeupByBusUsed = False	STD_OFF

1.2.3 Array: CanTrcv_17_W9255_ChannelUsed [CANTRCV_17_W9255_CHANNELS_CONFIGURED]

Table 1 CanTrcv_17_W9255_ChannelUsed [CANTRCV_17_W9255_CHANNELS_CONFIGURED]

Name	CanTrcv_17_W9255_ChannelUsed [CANTRCV_17_W9255_CHANNELS_CONFIGURED]	
Type	uint8	
Description	Each array element represents the index for a channel which is enabled.	
Verification method	The array contains the index for each enabled channel. This index is used to access the CanTrcv_17_W9255_ChannelConfig configuration structure. *Note: If a channel is disabled, 0xFFU is generated as the index value for the respective channel. Atleast one channel is to be enabled.*	
Example(s)	**Action**	**Generated output**

	• Configure 3 transceiver channels.
	• Set 'CanTrcvChannelUsed' parameter to 'True' for channels with Ids 0 and 2
		`const uint8 CanTrcv_17_W9255_ChannelUsed[CANTRCV_17_W9255_CHANNELS_CONFIGURED] =`
		`{`
		`/* CAN Transceiver Channel Id 0 is used */`
		`0U,`
		`/* CAN Transceiver Channel Id 1 is not used */`
		`0xFFU,`
		`/* CAN Transceiver Channel Id 2 is used */`
		`1U`
		`};`

	• Configure 3 transceiver channels.
	• Set 'CanTrcvChannelUsed' parameter to 'True' for channel with Id 0
		`const uint8 CanTrcv_17_W9255_ChannelUsed[CANTRCV_17_W9255_CHANNELS_CONFIGURED] =`
		`{`
		`/* CAN Transceiver Channel Id 0 is used */`
		`0U,`
		`/* CAN Transceiver Channel Id 1 is not used */`
		`0xFFU,`
		`/* CAN Transceiver Channel Id 2 is not used */`
		`0xFFU`
		`};`

1.2.4 Array: CanTrcv_17_W9255_PnConfigured [CANTRCV_17_W9255_CHANNELS_CONFIGURED]

Table 2 CanTrcv_17_W9255_PnConfigured[CANTRCV_17_W9255_CHANNELS_CONFIGURED]

Name	CanTrcv_17_W9255_PnConfigured[CANTRCV_17_W9255_CHANNELS_CONFIGURED]	
Type	uint8	
Description	Each array element represents the index for a channel which is PN enabled.	
Verification method	The array contains the index for each PN enabled channel. This index is used to access the PN configuration data of a particular channel which is PN enabled. *Note: If PN is disabled for a channel, 0xFFU is generated as the index value for the respective channel.*	
Example(s)	**Action**	**Generated output**

	• Configure 3 transceiver channels.
	• Set 'CanTrcvPnEnabled' parameter to 'False' for all the channels.
		`const uint8 CanTrcv_17_W9255_PnConfigured[CANTRCV_17_W9255_CHANNELS_CONFIGURED] =`
		`{`
		`  /* PN for CAN Transceiver Channel Id 0 is not configured */`
		`  0xFFU,`
		`  /* PN for CAN Transceiver Channel Id 1 is not configured */`
		`  0xFFU,`
		`  /* PN for CAN Transceiver Channel Id 2 is not configured */`
		`  0xFFU`
		`};`

	• Configure 3 transceiver channels.
	• Set 'CanTrcvPnEnabled' parameter to 'True' for channel with Id 2
		`const uint8 CanTrcv_17_W9255_PnConfigured[CANTRCV_17_W9255_CHANNELS_CONFIGURED] =`
		`{`
		`  /* PN for CAN Transceiver Channel Id 0 is not configured */`
		`  0xFFU,`
		`  /* PN for CAN Transceiver Channel Id 1 is not configured */`
		`  0xFFU,`
		`  /* PN for CAN Transceiver Channel Id 2 is configured */`

		0U };

1.2.5 Array: CanTrcv_17_W9255_Ch[x]PnIdInfo[]

Table 3 CanTrcv_17_W9255_Ch[x]PnIdInfo[]

Name	CanTrcv_17_W9255_Ch[x]PnIdInfo[]	
Type	uint16	
Description	The array elements are commands to configure the registers of TLE9255W for baud rate, CAN ID, ID mask and DLC of the PN frame. 'x' represents the channel Id.	
Verification method	10 array members are generated based on the values configured in the CanTrcvBaudRate, CanTrcvPnFrameCanId, CanTrcvPnFrameCanIdMask and CanTrcvPnFrameDlc parameters. The first member is the command to configure SWK_CTRL_2 register. The next 4 members are to configure ID control registers (SWK_ID0_CTRL - SWK_ID3_CTRL). The next 4 members are to configure ID mask control registers (SWK_MASK_ID0_CTRL - SWK_MASK_ID3_CTRL). The last member is used to configure the DLC control register (SWK_DLC_CTRL). *Note: This array is generated only for the channels which have the parameter 'CanTrcvPartialNetwork/CanTrcvPnEnabled' enabled.*	
Example(s)	**Action**	**Generated output**

	• Configure 2 transceiver channels.
	• Enable 'CanTrcvPnEnabled' for channel Id 2.
	• Configure CanTrcvBaudRate-500, CanTrcvPnFrameCanId-0xffff, CanTrcvPnFrameCanIdMask-0xfd and CanTrcvPnFrameDlc-3
		`/* PN configuration commands(Baudrate, CAN Id, CAN Id mask and DLC) of CAN Transceiver Channel Id 2 */`
		`static const uint16 CanTrcv_17_W9255_Ch2PnIdInfo[] =`
		`{`
		`/* Baud rate of the WUF */`
		`/* Command to write to SWK_CTRL_2 register with PN enabled, BR ratio 10 and Baud of 500kbps */`
		`0x8694U,`
		
		`/* CAN ID of the WUF */`
		`/* Command to write CAN ID to SWK_ID0_CTRL register */`
		`0x8a00U,`
		`/* Command to write CAN ID to SWK_ID1_CTRL register */`
		`0x8900U,`
		`/* Command to write CAN ID to SWK_ID2_CTRL register */`
		`0x88fcU,`

		/* Command to write CAN ID to SWK_ID3_CTRL register */ 0x871fU, /* CAN ID Mask of the WUF */ /* Command to write CAN Id mask to SWK_MASK_ID0_CTRL register */ 0x8e00U, /* Command to write CAN Id mask to SWK_MASK_ID1_CTRL register */ 0x8d00U, /* Command to write CAN Id mask to SWK_MASK_ID2_CTRL register */ 0x8cf4U, /* Command to write CAN Id mask to SWK_MASK_ID3_CTRL register */ 0x8b03U, /* Data Length Code of the WUF */ 0x8f03U };
	• Configure 2 transceiver channels. • Enable PN for channel Ids 1 and 0. • For channel 0, configure CanTrcvBaudRate-1000, CanTrcvPnFrameCanId-0xfd, CanTrcvPnFrameCanIdMask-0x0 and CanTrcvPnFrameDlc- 8 • For channel 1, configure CanTrcvBaudRate- 1000, CanTrcvPnFrameCanId-0xfdf, CanTrcvPnFrameCanIdMask-0xff and CanTrcvPnFrameDlc-2	/* PN configuration commands(Baudrate, CAN Id, CAN Id mask and DLC) of CAN Transceiver Channel Id 0 */ static const uint16 CanTrcv_17_W9255_Ch0PnIdInfo[] = { /* Baud rate of the WUF */ /* Command to write to SWK_CTRL_2 register with PN enabled, BR ratio 10 and Baud of 1Mbps */ 0x8695U, /* CAN ID of the WUF */ /* Command to write CAN ID to SWK_ID0_CTRL register */ 0x8a00U, /* Command to write CAN ID to SWK_ID1_CTRL register */ 0x8900U, /* Command to write CAN ID to SWK_ID2_CTRL register */

```
  0x88f4U,
  /* Command to write CAN ID to
SWK_ID3_CTRL register */
  0x8703U,


  /* CAN ID Mask of the WUF */
  /* Command to write CAN Id mask to
SWK_MASK_ID0_CTRL register */
  0x8e00U,
  /* Command to write CAN Id mask to
SWK_MASK_ID1_CTRL register */
  0x8d00U,
  /* Command to write CAN Id mask to
SWK_MASK_ID2_CTRL register */
  0x8c00U,
  /* Command to write CAN Id mask to
SWK_MASK_ID3_CTRL register */
  0x8b00U,


  /* Data Length Code of the WUF */
  0x8f08U
};
/* PN configuration commands(Baudrate,
CAN Id, CAN Id mask and DLC) of CAN
Transceiver Channel Id 1 */
static const uint16
CanTrcv_17_W9255_Ch1PnIdInfo[] =
{
  /* Baud rate of the WUF */
  /* Command to write to SWK_CTRL_2
register with PN enabled, BR ratio 10
and Baud of 250kbps */
  0x8693U,


  /* CAN ID of the WUF */
  /* Command to write CAN ID to
SWK_ID0_CTRL register */
  0x8a00U,
  /* Command to write CAN ID to
SWK_ID1_CTRL register */
  0x8900U,
  /* Command to write CAN ID to
SWK_ID2_CTRL register */
```

```
  0x887cU,
  /* Command to write CAN ID to
SWK_ID3_CTRL register */
  0x871fU,


  /* CAN ID Mask of the WUF */
  /* Command to write CAN Id mask to
SWK_MASK_ID0_CTRL register */
  0x8e00U,
  /* Command to write CAN Id mask to
SWK_MASK_ID1_CTRL register */
  0x8d00U,
  /* Command to write CAN Id mask to
SWK_MASK_ID2_CTRL register */
  0x8cfcU,
  /* Command to write CAN Id mask to
SWK_MASK_ID3_CTRL register */
  0x8b03U,


  /* Data Length Code of the WUF */
  0x8f02U
};
```

1.2.6 Array: CanTrcv_17_W9255_Ch[x]PnDataInfo[]

Table 4 CanTrcv_17_W9255_Ch[x]PnDataInfo[]

Name	CanTrcv_17_W9255_Ch[x]PnDataInfo[]	
Type	uint16	
Description	The array members are commands to configure the data mask registers of TLE9255W for data mask of the PN frame configured in the 'CanTrcvPartialNetwork/CanTrcvPnFrameDataMaskSpec' container. 'x' represents the channel Id.	
Verification method	The generated array members are used to configure the data control registers (SWK_DATA0_CTRL - SWK_DATA7_CTRL). The array members are generated based on the values configured in the CanTrcvPnFrameDataMask and CanTrcvPnFrameDataMaskIndex parameters. *Note: This array is generated only for the channels which have the parameter 'CanTrcvPartialNetwork/CanTrcvPnEnabled' enabled.* *Note: The size of the array depends on the value of DLC configured in the 'CanTrcvPartialNetwork/CanTrcvPnFrameDlc' configuration parameter.*	
Example(s)	**Action**	**Generated output**

	• Configure 3 transceiver channels.
	• Enable 'CanTrcvPnEnabled' for channel Id 1.
	• Configure 'CanTrcvPnFrameDlc' to 2.
	• Configure data masks with values 8 and 112 and with indices 0 and 1 respectively.
		`/* PN data mask configuration of CAN Transceiver Channel Id 1 */`
		`static const uint16 CanTrcv_17_W9255_Ch1PnDataInfo[] =`
		`{`
		`  /* Command to write Data mask to SWK_DATA0_CTRL register */`
		`  0x9708U,`
		`  /* Command to write Data mask to SWK_DATA1_CTRL register */`
		`  0x9670U`
		`};`

	• Configure 3 transceiver channels.
	• Enable 'CanTrcvPnEnabled' for channel Ids 0 and 2.
	• Configure 'CanTrcvPnFrameDlc' to 8 and 3 for channel Ids 0 and 2 respectively.
	• For channel Id 0, configure data masks 255, 124, 2, 67, 45, 1, 0 and 254 with indices 0 to 7 respectively.
	• For channel Id 2, configure data masks 0, 7 and 77 with indices 0 to 2 respectively.
		`/* PN data mask configuration of CAN Transceiver Channel Id 0 */`
		`static const uint16 CanTrcv_17_W9255_Ch0PnDataInfo[] =`
		`{`
		`  /* Command to write Data mask to SWK_DATA0_CTRL register */`
		`  0x97ffU,`
		`  /* Command to write Data mask to SWK_DATA1_CTRL register */`
		`  0x96ffU,`
		`  /* Command to write Data mask to SWK_DATA2_CTRL register */`
		`  0x95ffU,`
		`  /* Command to write Data mask to SWK_DATA3_CTRL register */`
		`  0x94ffU,`
		`  /* Command to write Data mask to SWK_DATA4_CTRL register */`
		`  0x93ffU,`
		`  /* Command to write Data mask to SWK_DATA5_CTRL register */`
		`  0x92ffU,`
		`  /* Command to write Data mask to SWK_DATA6_CTRL register */`
		`  0x91ffU,`
		`  /* Command to write Data mask to SWK_DATA7_CTRL register */`
		`  0x90ffU`
		`};`

		/* PN data mask configuration of CAN Transceiver Channel Id 2 */
		static const uint16 CanTrcv_17_W9255_Ch2PnDataInfo[] =
		{
		/* Command to write Data mask to SWK_DATA0_CTRL register */
		0x9700U,
		/* Command to write Data mask to SWK_DATA1_CTRL register */
		0x9607U,
		/* Command to write Data mask to SWK_DATA2_CTRL register */
		0x954dU
		};

Revision history

Major changes since the last revision

Date	Version	Description
2020-11-02	1.0	• Released.
2020-11-02	0.1	• Updated verification method for CanTrcvWakeupSourceRef, CANTRCV_17_W9255_PN_CONFIGURED_CHANNELS and CanTrcvSpiSequence. • Added additional examples for CanTrcvSpiSequence, CanTrcvSpiChannel, CanTrcvPnIdInfoPtr, CanTrcvPnDataInfoPtr, CanTrcvPnDlc, Added array descriptions for CanTrcv_17_W9255_Ch[x]PnIdInfo[], CanTrcv_17_W9255_Ch[x]PnDataInfo[] • CANTRCV_17_W9255_ RUNTIME_ERROR_DETECT configuration macro added. • CanTrcv_17_W9255 driver chapter moved from MC-ISAR_TC3xx_Config_Verification_Manual_Basic.pdf to this document

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

Edition 2019-07-17
Published by

Infineon Technologies AG

81726 Munich, Germany

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 (“Beschaffenheitsgarantie”) .

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.