MCAL Configuration Verification Manual for CanTrcv_17_V9251

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_V9251

Table of contents

CANTRCV_17_V9251 driver

This chapter describes the details of the configuration data generated from the CAN transceiver V9251 driver.

1.1 File: CANTRCV_17_V9251_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_V9251_AR_RELEASE_MAJOR_VERSION

Table 1 CANTRCV_17_V9251_AR_RELEASE_MAJOR_VERSION

Name	CANTRCV_17_V9251_AR_RELEASE_MAJOR_VERSION	
Description	Major version number of AUTOSAR release on which the CANTRCV_17_V9251 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_V9251_Cfg.h file with ArMajorVersion 4	`#define CANTRCV_17_V9251_AR_RELEASE_MAJOR_VERSION (4U)`

1.1.2 Macro: CANTRCV_17_V9251_AR_RELEASE_MINOR_VERSION

Table 2 CANTRCV_17_V9251_AR_RELEASE_MINOR_VERSION

Name	CANTRCV_17_V9251_AR_RELEASE_MINOR_VERSION	
Description	Minor version number of AUTOSAR release on which the CANTRCV_17_V9251 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_V9251_Cfg.h file with ArMinorVersion 2	`#define CANTRCV_17_V9251_AR_RELEASE_MINOR_VERSION (2U)`

1.1.3 Macro: CANTRCV_17_V9251_AR_RELEASE_REVISION_VERSION

Table 3 CANTRCV_17_V9251_AR_RELEASE_REVISION_VERSION

Name	CANTRCV_17_V9251_AR_RELEASE_REVISION_VERSION
Description	Revision version number of AUTOSAR release on which the CANTRCV_17_V9251 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_V9251_Cfg.h file with ArPatchVersion 2	`#define CANTRCV_17_V9251_AR_RELEASE_REVISION_VERSION (2U)`

1.1.4 Macro: CANTRCV_17_V9251_SW_MAJOR_VERSION

Table 4 CANTRCV_17_V9251_SW_MAJOR_VERSION

Name	CANTRCV_17_V9251_SW_MAJOR_VERSION
Description	Major version number of the CANTRCV_17_V9251 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_V9251_Cfg.h file with SwMajorVersion 10	`#define CANTRCV_17_V9251_SW_MAJOR_VERSION (10U)`

1.1.5 Macro: CANTRCV_17_V9251_SW_MINOR_VERSION

Table 5 CANTRCV_17_V9251_SW_MINOR_VERSION

Name	CANTRCV_17_V9251_SW_MINOR_VERSION
Description	Minor version number of the CANTRCV_17_V9251 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_V9251_Cfg.h file with SwMinorVersion 30	`#define CANTRCV_17_V9251_SW_MINOR_VERSION (30U)`

1.1.6 Macro: CANTRCV_17_V9251_SW_PATCH_VERSION

Table 6 CANTRCV_17_V9251_SW_PATCH_VERSION

Name	CANTRCV_17_V9251_SW_PATCH_VERSION	
Description	Patch version number of the CANTRCV_17_V9251 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_V9251_Cfg.h file with SwPatchVersion 0	`#define CANTRCV_17_V9251_SW_PATCH_VERSION (0U)`

1.1.7 Macro: CANTRCV_17_V9251_DEV_ERROR_DETECT

Table 7 CANTRCV_17_V9251_DEV_ERROR_DETECT

Name	CANTRCV_17_V9251_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_V9251_DEV_ERROR_DETECT (STD_ON)`
	CanTrcvDevErrorDetect = False	`#define CANTRCV_17_V9251_DEV_ERROR_DETECT (STD_OFF)`

1.1.8 Macro: CANTRCV_17_V9251_WAIT_COUNT

Table 8 CANTRCV_17_V9251_WAIT_COUNTME

Name	CANTRCV_17_V9251_WAIT_COUNT	
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**

	Configure CanTrcvWaitTime = 2e-5. The parameter is configured in seconds.	`#define CANTRCV_17_V9251_WAIT_COUNT (20000U)`

1.1.9 Macro: CANTRCV_17_V9251_GET_VERSION_INFO

Table 9 CANTRCV_17_V9251_GET_VERSION_INFO

Name	CANTRCV_17_V9251_GET_VERSION_INFO	
Description	Enables/disables CANTRCV_17_V9251_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_V9251_GET_VERSION_INFO (STD_ON)`
	CanTrcvGetVersionInfo = False	`#define CANTRCV_17_V9251_GET_VERSION_INFO (STD_OFF)`

1.1.10 Macro: CANTRCV_17_V9251_GENERAL_WAKE_UP_SUPPORT

Table 10 CANTRCV_17_V9251_GENERAL_WAKE_UP_SUPPORT

Name	CANTRCV_17_V9251_GENERAL_WAKE_UP_SUPPORT	
Description	Indicates wake up support type of CANTRCV_17_V9251 module.	
Verification method	The macro is generated as CANTRCV_17_V9251_WAKE_UP_BY_INTERRUPT since only interrupt mode is supported.	
Example(s)	**Action**	**Generated output**
	Configure CanTrcvWakeUpSupport = CANTRCV_17_V9251_WAKE_UP_BY_INTERRUPT	`#define CANTRCV_17_V9251_GENERAL_WAKE_UP_SUPPORT \ (CANTRCV_17_V9251_WAKE_UP_BY_INTERRUPT)`

1.1.11 Macro: CANTRCV_17_V9251_INSTANCE_ID

Table 11 CANTRCV_17_V9251_INSTANCE_ID

Name	CANTRCV_17_V9251_INSTANCE_ID
Description	Instance ID of CANTRCV_17_V9251 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_V9251_INSTANCE_ID ((uint8)0)`
	Set CanTrcvIndex as 2	`#define CANTRCV_17_V9251_INSTANCE_ID ((uint8)2)`

1.1.12 Macro: CanTrcv_17_V9251Conf_CanTrcvChannel_<channel name>

Table 12 CanTrcv_17_V9251Conf_CanTrcvChannel_<channel name>

Name	CanTrcv_17_V9251Conf_CanTrcvChannel_<channel name>	
Description	The macro is the symbolic name generated for the configuration parameter 'CanTrcvConfigSet/ CanTrcvChannel/CanTrcvChannelId'	
Verification method	The macro is generated as a numeric value which is configured in 'CanTrcvConfigSet/ CanTrcvChannel'. <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_V9251Conf_CanTrcvChannel_CanTrcvChannel_1`
		`#define CanTrcv_17_V9251Conf_CanTrcvChannel_CanTrcvChannel_1 ((uint8)1U)`
		`#endif`
		
		`#ifndef CanTrcv_17_V9251Conf_CanTrcvChannel_CanTrcvChannel_3`
		`#define CanTrcv_17_V9251Conf_CanTrcvChannel_CanTrcvChannel_3 ((uint8)3U)`
		`#endif`

	• Configure 4 transceiver channels (CanTrcvChannel_0 to CanTrcvChannel_3)
	• Configure CanTrcvChannel_0 and CanTrcvChannel_2 as used channels
		`#ifndef CanTrcv_17_V9251Conf_CanTrcvChannel_CanTrcvChannel_0`
		`#define CanTrcv_17_V9251Conf_CanTrcvChannel_CanTrcvChannel_0 ((uint8)0U)`
		`#endif`
		
		`#ifndef CanTrcv_17_V9251Conf_CanTrcvChannel_CanTrcvChannel_2`

		#define CanTrcv_17_V9251Conf_CanTrcvChannel_CanTrcvChannel_2 ((uint8)2U) #endif

1.1.13 Macro: CANTRCV_17_V9251_CH_<x>_MAX_BAUDRATE_SUPPORT

Table 13 CANTRCV_17_V9251_CH_<x>_MAX_BAUDRATE_SUPPORT

Name	CANTRCV_17_V9251_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 1000	#define CANTRCV_17_V9251_CH_1_MAX_BAUDRATE_SUPPORT (1000U) #define CANTRCV_17_V9251_CH_3_MAX_BAUDRATE_SUPPORT (1000U)
	• 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 5000	#define CANTRCV_17_V9251_CH_0_MAX_BAUDRATE_SUPPORT (5000U) #define CANTRCV_17_V9251_CH_2_MAX_BAUDRATE_SUPPORT (5000U)

1.1.14 Macro: CANTRCV_17_V9251_CH_<x>_ICU_REF

Table 14 CANTRCV_17_V9251_CH_<x>_ICU_REF

Name	CANTRCV_17_V9251_CH_<x>_ICU_REF	
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 referenced by the transceiver channel <x> if CanTrcvIcuChannelRef parameter is referenced and is generated as ICU_REFERENCE_NOT_CONFIGURED if not referenced. *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 'CanTrcvWakeUpSupport' as CANTRCV_17_V9251_WAKE_UP_BY_INTERRUPT
	• Enable CanTrcvChannelUsed for channels CanTrcvChannel_0 and CanTrcvChannel_3
	• Enable 'CanTrcvWakeupByBusUsed' for channel CanTrcvChannel_0 and disable it for channel CanTrcvChannel_3
		`#define CANTRCV_17_V9251_CH_0_ICU_REF (IcuConf_IcuChannel_IcuChannel_0)`
		
		`#define CANTRCV_17_V9251_CH_3_ICU_REF (ICU_REFERENCE_NOT_CONFIGURED)`

	• Configure 4 transceiver channels (CanTrcvChannel_0 to CanTrcvChannel_3)
	• Configure 'CanTrcvWakeUpSupport' as CANTRCV_17_V9251_WAKE_UP_BY_INTERRUPT
	• Enable CanTrcvChannelUsed for channels CanTrcvChannel_1 and CanTrcvChannel_2
	• Enable 'CanTrcvWakeupByBusUsed' for channel CanTrcvChannel_1 and disable it for channel CanTrcvChannel_2
		`#define CANTRCV_17_V9251_CH_1_ICU_REF (IcuConf_IcuChannel_IcuChannel_0)`
		
		`#define CANTRCV_17_V9251_CH_2_ICU_REF (ICU_REFERENCE_NOT_CONFIGURED)`

1.1.15 Macro: CANTRCV_17_V9251_CHANNELS_USED

Table 15 CANTRCV_17_V9251_CHANNELS_USED

Name	CANTRCV_17_V9251_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. • Set 'CanTrcvChannelUsed' parameter to 'True' for any 2 configured channels	`#define CANTRCV_17_V9251_CHANNELS_USED ((uint8)2U)`
	• Configure 4 transceiver channels. • Set 'CanTrcvChannelUsed' parameter to 'True' for all the configured channels	`#define CANTRCV_17_V9251_CHANNELS_USED ((uint8)4U)`

1.1.16 Macro: CANTRCV_17_V9251_CHANNELS_CFG

Table 16 CANTRCV_17_V9251_CHANNELS_CFG

Name	CANTRCV_17_V9251_CHANNELS_CFG	
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_V9251_CHANNELS_CFG ((uint8)4U)`
	Configure 8 transceiver channels. (CanTrcvChannel_0 to CanTrcvChannel_7)	`#define CANTRCV_17_V9251_CHANNELS_CFG ((uint8)8U)`

1.2 File: CANTRCV_17_V9251_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_V9251_ChannelConfig [CANTRCV_17_V9251_CHANNELS_USED]

Table 17 CanTrcv_17_V9251_ChannelConfig[CANTRCV_17_V9251_CHANNELS_USED]

Name	CanTrcv_17_V9251_ChannelConfig[CANTRCV_17_V9251_CHANNELS_USED]
Type	CanTrcv_17_V9251_ChannelInfoType
Description	Channel configuration structure of CANTRCV_17_V9251 driver
Verification method	The generated structure is present in CANTRCV_17_V9251_Cfg.c file. The macro CANTRCV_17_V9251_CHANNELS_USED is defined in CANTRCV_17_V9251_Cfg.h file. The size of the structure depends on the number of channels enabled.

	Note: *A channel can be enabled/disabled by the 'CanTrcvChannel/CanTrcvChannelUsed' configuration parameter.*	
Example(s)	**Action**	**Generated output**

	Configure 4 channels and enable channels with Ids 1 and 3 among them.	`const CanTrcv_17_V9251_ChannelInfoType CanTrcv_17_V9251_ChannelConfig[CANTRCV_17_V9251_CHANNELS_USED] =`
		`{`
		`  /* CanTransceiver Channel 1 Specific Information */`
		`  {`
		`  CANTRCV_TRCVMODE_STANDBY,`
		`  4U,`
		`  DioConf_DioChannel_DioChannel_STB_0,`
		`  CANTRCV_17_V9251_STANDBY_MODE,`
		`  1U,`
		`  TRUE`
		`  },`
		`  /* CanTransceiver Channel 3 Specific Information */`
		`  {`
		`  CANTRCV_TRCVMODE_NORMAL,`
		`  WAKEUP_SOURCE_NOT_CONFIGURED,`
		`  DioConf_DioChannel_DioChannel_STB_2,`
		`  CANTRCV_17_V9251_NORMAL_MODE,`
		`  3U,`
		`  FALSE`
		`  }`
		`};`

	Configure 4 channels and enable channels with Ids 0 and 2 among them	`const CanTrcv_17_V9251_ChannelInfoType CanTrcv_17_V9251_ChannelConfig[CANTRCV_17_V9251_CHANNELS_USED] =`
		`{`
		`  /* CanTransceiver Channel 0 Specific Information */`
		`  {`
		`  CANTRCV_TRCVMODE_NORMAL,`
		`  4U,`
		`  DioConf_DioChannel_DioChannel_STB_0,`
		`  CANTRCV_17_V9251_NORMAL_MODE,`
		`  0U,`
		`  TRUE`

```
  },
  /* CanTransceiver Channel 2 Specific Information */
  {
    CANTRCV_TRCVMODE_NORMAL,
    WAKEUP_SOURCE_NOT_CONFIGURED,
    DioConf_DioChannel_DioChannel_STB_2,
    CANTRCV_17_V9251_NORMAL_MODE,
    2U,
    FALSE
  }
};
```

1.2.1.1 Member: CanTrcv_17_V9251_NetworkMode

Table 18 CanTrcv_17_V9251_NetworkMode

Name	CanTrcv_17_V9251_NetworkMode	
Type	CanTrcv_TrcvModeType	
Description	State of the transceiver channel after initialization.	
Verification method	The structure member is generated as an enum value of type CanTrcv_TrcvModeType according to the mode configured in the CanTrcvInitState parameter.	
Example(s)	**Action**	**Generated output**
	Configure CanTrcvInitState parameter of Channel 1 as CANTRCV_17_V9251_OP_MODE_NORMAL	`CANTRCV_TRCVMODE_NORMAL`
	Configure CanTrcvInitState parameter of Channel 1 as CANTRCV_17_V9251_OP_MODE_STANDBY	`CANTRCV_TRCVMODE_STANDBY`

1.2.1.2 Member: CanTrcv_17_V9251_WakeupSourceRef

Table 19 CanTrcv_17_V9251_WakeupSourceRef

Name	CanTrcv_17_V9251_WakeupSourceRef
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 WAKEUP_SOURCE_NOT_CONFIGURED.

	Note: *The configuration parameter CanTrcvWakeupSourceRef is not user configurable if CanTrcvWakeupByBusUsed configuration parameter is disabled.*	
Example(s)	**Action**	**Generated output**
	Enable 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.	`WAKEUP_SOURCE_NOT_CONFIGURED`

1.2.1.3 Member: CanTrcv_17_V9251_DioChannel

Table 20 CanTrcv_17_V9251_DioChannel

Name	CanTrcv_17_V9251_DioChannel	
Type	Dio_ChannelType	
Description	Reference to the DIO channel configured for the respective transceiver channel.	
Verification method	The structure member is generated with the symbolic name of the DIO channel referenced by the CanTrcvDioSymNameRef configuration parameter for the respective transceiver channel.	
Example(s)	**Action**	**Generated output**
	Refer DioChannelId 1 in CanTrcvDioSymNameRef configuration parameter of transceiver Channel Id 0	`DioConf_DioChannel_DioChannel_STB_0`
	Refer DioChannelId 4 in CanTrcvDioSymNameRef configuration parameter of transceiver Channel Id 1	`DioConf_DioChannel_DioChannel_STB_2`

1.2.1.4 Member: CanTrcv_17_V9251_DioPinLevel

Table 21 CanTrcv_17_V9251_DioPinLevel

Name	CanTrcv_17_V9251_DioPinLevel	
Type	Dio_LevelType	
Description	Internal value for setting DIO Level based on the init state of the transceiver.	
Verification method	The structure member is generated as a macro name depending on the configuration in 'CanTrcvChannel/CanTrcvInitState' parameter. CANTRCV_17_V9251_NORMAL_MODE is printed if CanTrcvInitState is set to CANTRCV_17_V9251_OP_MODE_NORMAL and CANTRCV_17_V9251_STANDBY_MODE is printed if CanTrcvInitState is set to CANTRCV_17_V9251_OP_MODE_STANDBY	
Example(s)	**Action**	**Generated output**
	Configure CanTrcvInitState parameter of Channel 1 as CANTRCV_17_V9251_OP_MODE_NORMAL	`CANTRCV_17_V9251_NORMAL_MODE`

	Configure CanTrcvInitState parameter of Channel 1 as CANTRCV_17_V9251_OP_MODE_STANDBY	CANTRCV_17_V9251_STANDBY_MODE

1.2.1.5 Member: CanTrcv_17_V9251_ChannelId

Table 22 CanTrcv_17_V9251_ChannelId

Name	CanTrcv_17_V9251_ChannelId	
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_0 in the CanTrcvChannel container and configure value 0 in the CanTrcvChannelId configuration parameter of that instance	0U
	Add a channel instance CanTrcvChannel_1 in the CanTrcvChannel container and configure value 1 in the CanTrcvChannelId configuration parameter of that instance	1U

1.2.1.6 Member: CanTrcv_17_V9251_WakeupbyBus

Table 23 CanTrcv_17_V9251_WakeupbyBus

Name	CanTrcv_17_V9251_WakeupbyBus	
Type	boolean	
Description	Wake-up by bus status of the channel.	
Verification method	The structure member is generated as TRUE if the parameter CanTrcvWakeupByBusUsed is set true and FALSE if the parameter is set false.	
Example(s)	**Action**	**Generated output**
	Configure a transceiver channel with CanTrcvWakeupByBusUsed = True	TRUE
	Configure a transceiver channel with CanTrcvWakeupByBusUsed = False	FALSE

1.2.2 Array: CanTrcv_17_V9251_ChannelUsed [CANTRCV_17_V9251_CHANNELS_CFG]

Table 1 CanTrcv_17_V9251_ChannelUsed[CANTRCV_17_V9251_CHANNELS_CFG]

Name	CanTrcv_17_V9251_ChannelUsed[CANTRCV_17_V9251_CHANNELS_CFG]

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_V9251_ChannelConfig' channel 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_V9251_ChannelUsed[CANTRCV_17_V9251_CHANNELS_CFG] =`
		`{`
		`  /* Index of CAN Transceiver Channel_0 is used */`
		`  0U,`
		`  /* CAN Transceiver Channel_1 is not used */`
		`  0xFFU,`
		`  /* Index of CAN Transceiver Channel_2 is used */`
		`  1U`
		`};`

	• Configure 3 transceiver channels.
	• Set 'CanTrcvChannelUsed' parameter to 'True' for channel with Id 0
		`const uint8 CanTrcv_17_V9251_ChannelUsed[CANTRCV_17_V9251_CHANNELS_CFG] =`
		`{`
		`  /* Index of CAN Transceiver Channel_0 is used */`
		`  0U,`
		`  /* CAN Transceiver Channel_1 is not used */`
		`  0xFFU,`
		`  /* CAN Transceiver Channel_2 is not used */`
		`  0xFFU`
		`};`

Revision history

Major changes since the last revision

Date	Version	Description
2020-11-02	1.0	• Released.
2020-11-02	0.1	• Added additional examples for CANTRCV_17_V9251_CH_<x>_MAX_BAUDRATE_SUPPORT, CANTRCV_17_V9251_CH_<x>_ICU_REF, updated examples for CanTrcv_17_V9251_DioChannel, updated type for CanTrcv_17_V9251_DioPinLevel, updated verification method for CanTrcv_17_V9251_WakeupSourceRef, Updated output generated for example 2 in table 1.1.12 • CanTrcv_17_V9251 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.