MCAL User Manual for CanTrcv_17_V9251

32-bit TriCore^{TM} AURIX^{TM} TC3xx microcontroller

About this document

Scope and purpose

This User Manual is intended to enable users to integrate the Microcontroller Abstraction Layer (MCAL) software for the TriCore^{TM} AURIX^{TM} family of 32-bit microcontrollers.

This document describes responsibilities of integrator in-charge of integrating MCAL software with the basic software (BSW) stack. This document also provides detailed information on safety, configuration and functions along with examples of usage of significant features.

Note: Detailed information about package installation, safety and other generic information that are common across all modules are provided in MCAL User Manual General.

Intended audience

This document is intended for anyone using the CanTrcv_17_V9251 module of the TC3xx MCAL software.

Document conventions

Table 1 Conventions

Convention	Explanation
Bold	Emphasizes heading levels, column headings, table and figure captions, screen names, windows, dialog boxes, menus, sub-menus
Italics	Denotes variable(s) and reference(s)
`Courier` `New`	Denotes APIs, functions, interrupt handlers, events, data types, error handlers, file/folder names, directories, command line inputs, code snippets
>	Indicates that a cascading sub-menu opens when you select a menu item
[cover parentID=<alpha numeric value>]	Used for traceability completeness. Reader should ignore these.

Reference documents

This User Manual should be read in conjunction with the following documents:

- AURIX^{TM} TC3xx MCAL User Manual General
- Specification of CAN Transceiver Driver, AUTOSAR_SWS_CAN_Transceiver_Driver, AUTOSAR Release 4.2.2
- Specification of CAN Transceiver Driver, AUTOSAR_SWS_CAN_Transceiver_Driver, AUTOSAR Release 4.4.0

Please read the Important Notice and Warnings at the end of this document

Table of contents

Table of contents

Table of contents

1 CanTrcv_17_V9251 driver

1.1 User information

1.1.1 Description

The CAN transceiver is a hardware device, which adapts the signal levels that are used on the CAN bus to the logical (digital) signal levels recognized by a microcontroller. CAN Transceiver is part of the ECU Abstraction layer and works as an interface between the CAN protocol controller and the physical differential bus. CAN Transceiver driver is implemented to support the Infineon TLE9251V hardware. It supports the wake-up functionality through the bus, which wakes up only for valid wake-up pattern (WUP). The DIO interface is used to control the modes of the CAN Transceiver. The CAN transceiver, TLE9251V, supports the NORMAL and STANDBY modes. The CanTrcv_17_V9251 driver provides the services for:

- Initialization of the CAN Transceivers.
- Controlling the operation mode of CAN Transceivers through the DIO.
- Enabling/disabling the wake-up of the CAN Transceivers.

1.1.2 Hardware-software mapping

This section describes the system view of the CanTrcv_17_V9251 driver and peripherals administered by it.

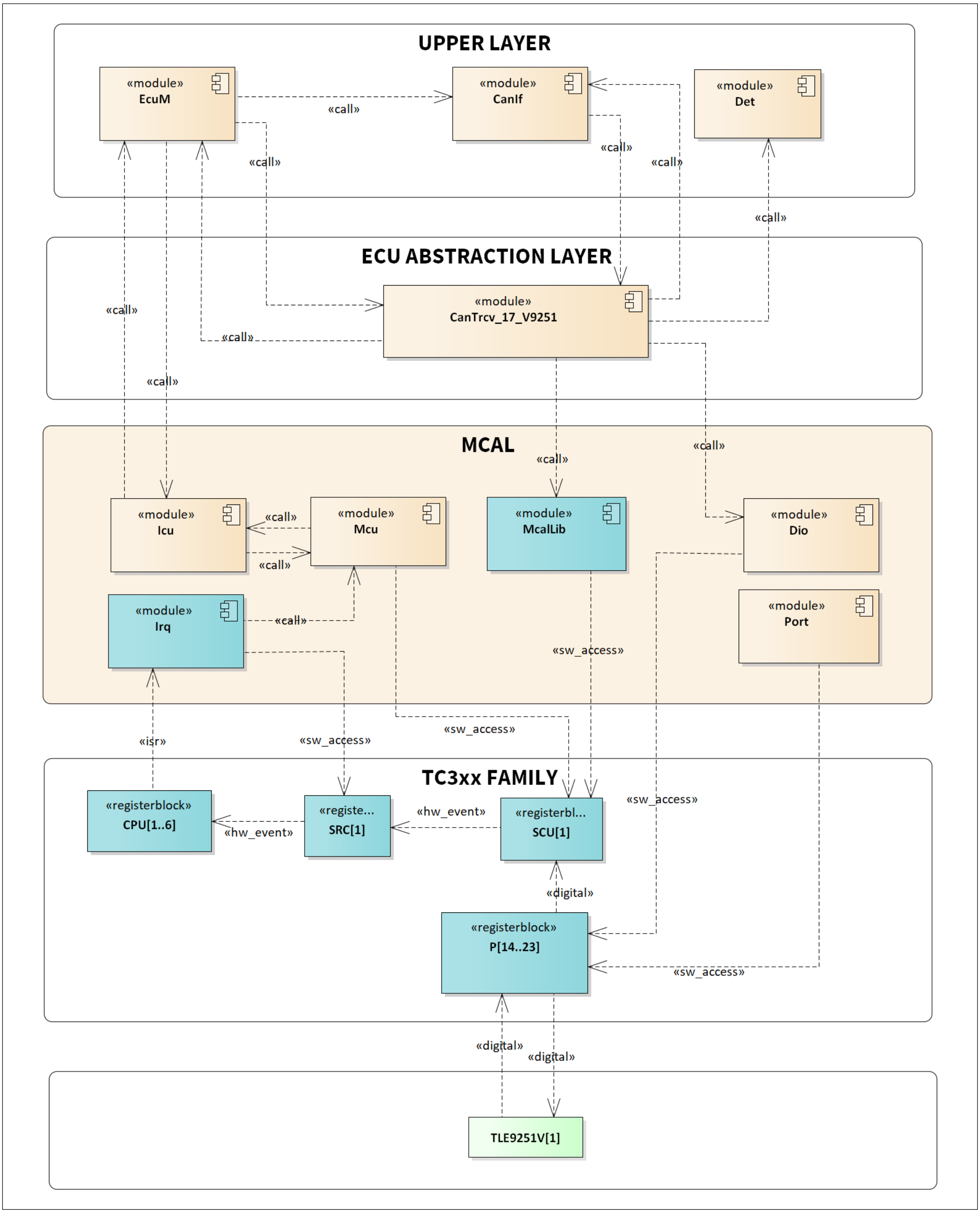

Figure 1 Mapping of hardware-software interfaces

1.1.2.1 Port: dependent hardware peripheral

Hardware functional features

The digital signals are routed to the CAN transceiver hardware through the digital port pads. The port pads are configured and enabled through the PORT driver. The CanTrcv_17_V9251 driver depends on the PORT driver for configuring the RxD, TxD, STB pins of the CAN transceiver hardware.

Users of the hardware

The port pads are configured by the PORT driver.

Hardware diagnostic features

Not applicable.

Hardware events

Not applicable.

1.1.2.2 SCU: dependent hardware peripheral

Hardware functional features

The CAN Transceiver driver depends on the SCU IP for the clock and reset functionalities.

Users of the hardware

The SCU IP supplies clock for all the peripherals and the MCU driver is responsible for configuring the clock tree. To avoid conflicts due to simultaneous writes, update to all the ENDINIT protected registers is performed using the MCALLIB APIs.

Hardware diagnostic features

The SMU alarms configured for the SCU IP are not monitored by the CAN transceiver driver.

Hardware events

Hardware events from the SCU are not used by the CAN transceiver driver.

1.1.2.3 SRC: dependent hardware peripheral

Hardware functional features

The CAN transceiver driver depends on the ICU for interrupt handling. The ICU depends on the interrupt router for raising an interrupt to the CPU based on the wake-up events, which indicates wake-up activity on the RxD pin of the transceiver. The RxD pin is connected to the edge detection channel of the ICU.

Users of the hardware

The interrupt router is configured either by the IRQ driver or the user software.

Hardware diagnostic features

The SMU alarms configured for interrupt router are not monitored by the CAN transceiver driver.

Hardware events

The interrupt events raised by the interrupt router are serviced by the CPU. The CAN transceiver driver depends on the ICU driver, which provides interrupt handlers as software interfaces that must be invoked from the ISR.

1.1.2.4 TLE9251V: primary hardware peripheral

Hardware functional features

The CAN Transceiver driver uses the TLE9251V to provide an interface between the physical CAN bus layer and the CAN protocol controller. The key hardware functional features used by the driver are:

- Interface between CAN controller and CAN physical bus
- Supports Normal and Standby operation modes
- Supports BUS wake up, i.e. wake up by valid Wake-up Pattern only

The unsupported features of the TLE9251V are:

- Forced-receive-only mode

Users of the hardware

The CAN Transceiver driver exclusively utilizes the TLE9251V module.

Hardware diagnostic features

Not applicable

Hardware events

Not applicable

1.1.3 File structure

1.1.3.1 C file structure

This section provides details of the C files of the CanTrcv_17_V9251 driver.

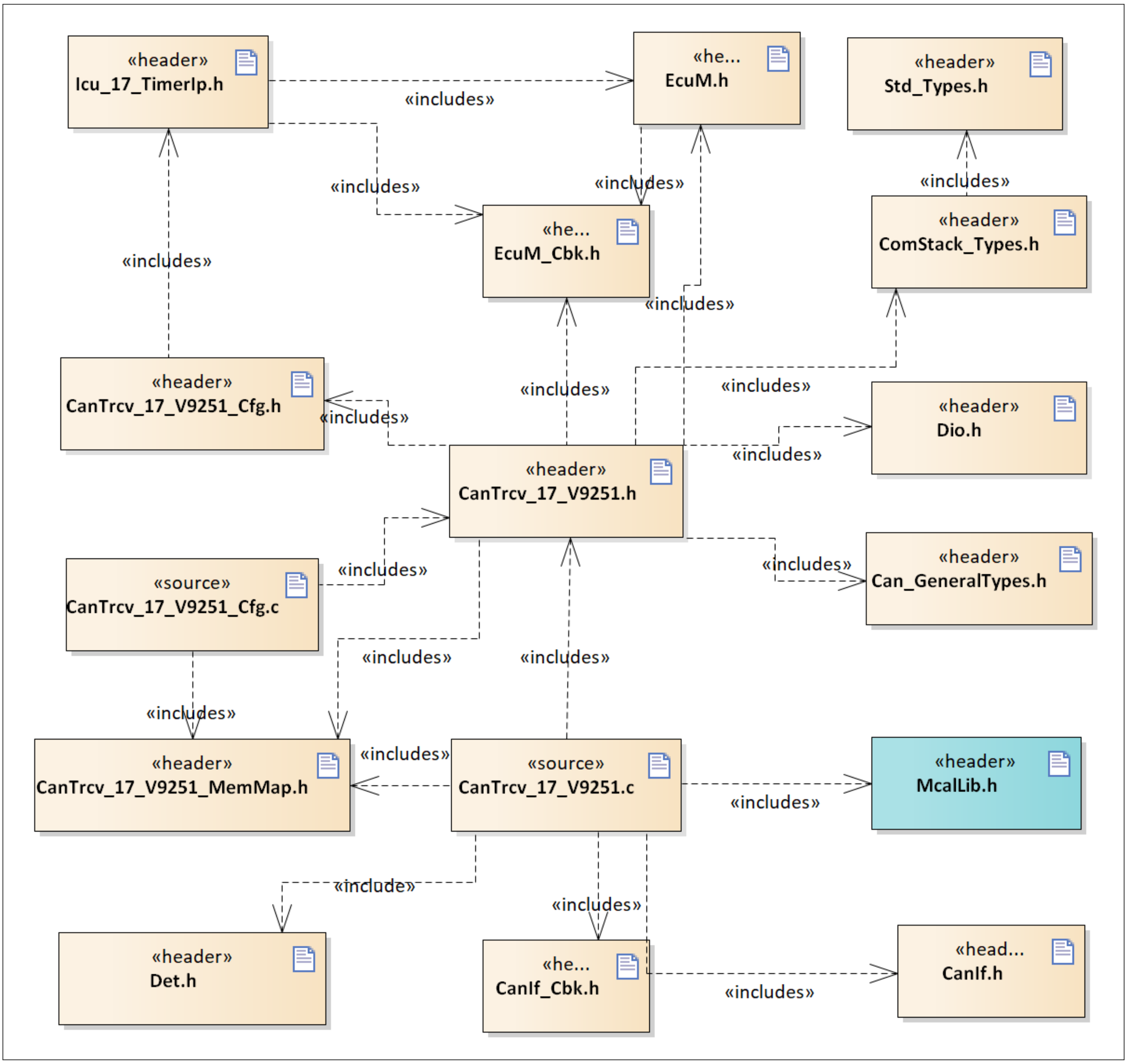
«header» Icu_17_TimerIp.h
«he... EcuM.h
«header» Std_Types.h
«he... EcuM_Cbk.h
«header» ComStack_Types.h
«header» CanTrcv_17_V9251_Cfg.h
«header» Dio.h
«header» CanTrcv_17_V9251.h
«source» CanTrcv_17_V9251_Cfg.c
«header» Can_GeneralTypes.h
«header» CanTrcv_17_V9251_MemMap.h
«source» CanTrcv_17_V9251.c
«header» McalLib.h
«header» Det.h
«he... CanIf_Cbk.h
«head... CanIf.h
«includes»
«include»

Figure 2 CanTrcv_17_V9251_File_Structure-1.png

Table 2 C file structure

File name	Description
`CanIf.h`	Header file containing the exported interfaces of CanIf
`CanIf_CanTrcv.h`	Header file containing declarations of the CanIf callbacks. *Note: This file is available only for AUTOSAR version 4.4.0*
`CanTrcv_17_V9251.c`	File (static) containing implementation of CanTrcv_17_V9251 APIs
`CanTrcv_17_V9251.h`	Header file (static) defining prototypes of data structures and APIs of CanTrcv_17_V9251 driver
`CanTrcv_17_V9251_Cfg.c`	File (generated) containing definition of the configuration data structures for the CanTrcv_17_V9251 driver

Table 2 C file structure (continued)

File name	Description
`CanTrcv_17_V9251_Cfg.h`	Header file (generated) containing CanTrcv_17_V9251 module constants and pre-processor macros as #defines
`CanTrcv_17_V9251_MemMap.h`	File (static) containing the memory section definitions used by the CanTrcv_17_V9251 driver
`Can_GeneralTypes.h`	Contains all types and constants that are shared among the AUTOSAR CAN modules Can, CanIf and CanTrcv
`ComStack_Types.h`	Type Definition for Com stack
`Det.h`	Provides the exported interfaces of Development Error Tracer
`Dio.h`	Header file (Static) defining prototypes of data structures and APIs
`EcuM.h`	Header file exporting the declarations of the EcuM
`Icu_17_TimerIp.h`	Header file (static) defining prototypes of configuration data structures and APIs
`McalLib.h`	Static header file defining prototypes of data structure and APIs exported by the MCALLIB.
`Std_Types.h`	Standard type declaration file as defined by AUTOSAR. It is independent of compiler or platform.

1.1.3.2 Code generator plugin files

This section provides details of the code generator plugin files of the CanTrcv_17_V9251 driver.

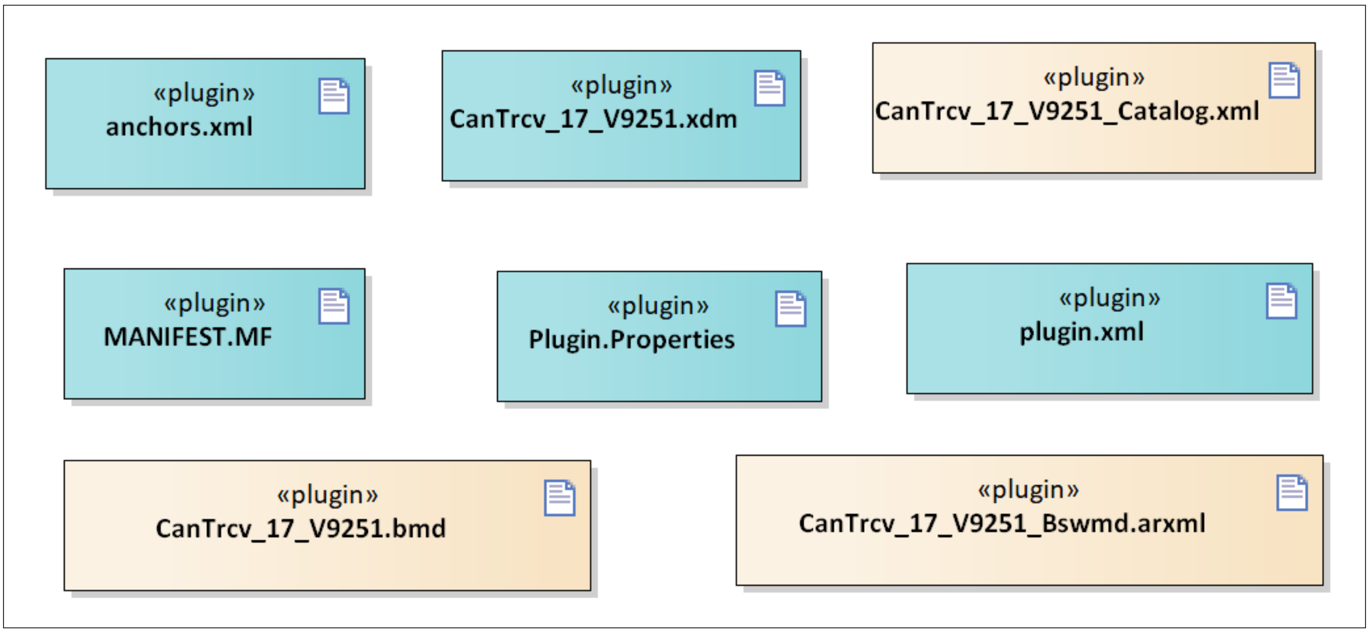

Figure 3 CanTrcv_17_V9251_Code_Generator_Plugin_Files-1.png

Table 3 Code generator plugin files

File name	Description
`CanTrcv_17_V9251.bmd`	AUTOSAR format XML data model schema file

Table 3 Code generator plugin files (continued)

File name	Description
`CanTrcv_17_V9251.xdm`	Tresos format XML data model schema file
`CanTrcv_17_V9251_Bswmd.arxml`	AUTOSAR format module description file
`CanTrcv_17_V9251_Catalog.xml`	AUTOSAR format catalog file as per catalog_V3_0_0.ml.xsd
`MANIFEST.MF`	Tresos plugin support file containing the metadata for the CanTrcv_17_V9251 driver
`Plugin.Properties`	Tresos plugin support file for the CanTrcv_17_V9251 driver
`anchors.xml`	Tresos anchors support file for the CanTrcv_17_V9251 driver
`plugin.xml`	Tresos plugin support file for the CanTrcv_17_V9251 driver

1.1.4 Integration hints

This section lists the key points that an integrator or user of the CanTrcv_17_V9251 driver must consider.

1.1.4.1 Integration with AUTOSAR stack

This section lists the modules, which are not part of the MCAL but are required to integrate the CanTrcv_17_V9251 driver.

- **EcuM**

  The ECU Manager module is a part of the AUTOSAR stack that manages common aspects of ECU. Specifically, in the context of the MCAL, the EcuM is used for initialization and de-initialization of the software drivers. The EcuM module provided in the MCAL package is a stub code and needs to be replaced with a complete EcuM module during the integration phase. The CAN transceiver driver uses the API of EcuM to provide notifications as listed.

  `EcuM_SetWakeupEvent()`: indication to EcuM for a valid wake-up from a transceiver channel.

- **CAN Interface (CanIf)**

  The CanIf module is a part of the AUTOSAR stack that provides upper layers a hardware independent interface to the CAN communication system comprising multiple CAN controllers and CAN transceivers .The `CanIf_Cbk.c` and `CanIf_Cbk.h` files are provided as stub code and needs to be replaced with complete CanIf module during integration phase. The CanTrcv driver uses the API of CanIf to provide notifications as listed.

  `CanIf_TrcvModeIndication()`: notification for a successful mode transition that was triggered for a transceiver channel.

- **Memory mapping**

  Memory mapping is a concept from AUTOSAR that allows relocation of text, variables, constants and configuration data to user-specific memory regions. To achieve this, all the relocatable elements of the driver are encapsulated in different memory-section macros. These macros are defined in the `CanTrcv_17_V9251_MemMap.h` file.

  The `CanTrcv_17_V9251_MemMap.h` file is provided in the MCAL package as a stub code. The integrator must place appropriate compiler pragmas within the memory-section macros. The pragmas ensure that the elements are relocated to the correct memory region. A sample implementation listing the memory section macros is shown as follows.

```
/****GLOBAL DATA SECTION ****/
#if defined CANTRCV_17_V9251_START_SEC_VAR_CLEARED_QM_LOCAL_UNSPECIFIED
/* User Pragma here */
 #undef CANTRCV_17_V9251_START_SEC_VAR_CLEARED_QM_LOCAL_UNSPECIFIED
 #undef MEMMAP_ERROR
#elif defined CANTRCV_17_V9251_STOP_SEC_VAR_CLEARED_QM_LOCAL_UNSPECIFIED
/* User Pragma here */
 #undef CANTRCV_17_V9251_STOP_SEC_VAR_CLEARED_QM_LOCAL_UNSPECIFIED
 #undef MEMMAP_ERROR
#elif defined CANTRCV_17_V9251_START_SEC_VAR_CLEARED_QM_LOCAL_8
/* User Pragma here */
 #undef CANTRCV_17_V9251_START_SEC_VAR_CLEARED_QM_LOCAL_8
 #undef MEMMAP_ERROR
#elif defined CANTRCV_17_V9251_STOP_SEC_VAR_CLEARED_QM_LOCAL_8
/* User Pragma here */
 #undef CANTRCV_17_V9251_STOP_SEC_VAR_CLEARED_QM_LOCAL_8
 #undef MEMMAP_ERROR
#elif defined CANTRCV_17_V9251_START_SEC_VAR_CLEARED_QM_LOCAL_16
/* User Pragma here */
 #undef CANTRCV_17_V9251_START_SEC_VAR_CLEARED_QM_LOCAL_16
 #undef MEMMAP_ERROR
#elif defined CANTRCV_17_V9251_STOP_SEC_VAR_CLEARED_QM_LOCAL_16
/* User Pragma here */
 #undef CANTRCV_17_V9251_STOP_SEC_VAR_CLEARED_QM_LOCAL_16
 #undef MEMMAP_ERROR

/**** CANTRCV_17_V9251 MODULE CONFIG DATA ****/
#elif defined CANTRCV_17_V9251_START_SEC_CONFIG_DATA_QM_LOCAL_UNSPECIFIED
/* User Pragma here */
 #undef CANTRCV_17_V9251_START_SEC_CONFIG_DATA_QM_LOCAL_UNSPECIFIED
 #undef MEMMAP_ERROR
#elif defined CANTRCV_17_V9251_STOP_SEC_CONFIG_DATA_QM_LOCAL_UNSPECIFIED
/* User Pragma here */
 #undef CANTRCV_17_V9251_STOP_SEC_CONFIG_DATA_QM_LOCAL_UNSPECIFIED
 #undef MEMMAP_ERROR
#elif defined CANTRCV_17_V9251_START_SEC_CONFIG_DATA_QM_LOCAL_8
/* User Pragma here */
 #undef CANTRCV_17_V9251_START_SEC_CONFIG_DATA_QM_LOCAL_8
 #undef MEMMAP_ERROR
#elif defined CANTRCV_17_V9251_STOP_SEC_CONFIG_DATA_QM_LOCAL_8
/* User Pragma here */
 #undef CANTRCV_17_V9251_STOP_SEC_CONFIG_DATA_QM_LOCAL_8
 #undef MEMMAP_ERROR

/**** CANTRCV_17_V9251 MODULE CODE SECTION ****/
#elif defined CANTRCV_17_V9251_START_SEC_CODE_QM_LOCAL
/* User Pragma here */
 #undef CANTRCV_17_V9251_START_SEC_CODE_QM_LOCAL
 #undef MEMMAP_ERROR
#elif defined CANTRCV_17_V9251_STOP_SEC_CODE_QM_LOCAL
/* User Pragma here */
 #undef CANTRCV_17_V9251_STOP_SEC_CODE_QM_LOCAL
```

```
  #undef MEMMAP_ERROR
#endif
```

- **DET**

  The DET module is a part of the AUTOSAR stack that handles all the development and runtime errors reported by the BSW modules. The CanTrcv_17_V9251 driver reports all the development errors to the DET module through the `Det_ReportError()` API. The user of the CanTrcv_17_V9251 driver must process all the errors reported to the DET module through the `Det_ReportError()` API. The `Det.h` and `Det.c` files are provided in the MCAL package as a stub code and needs to be replaced with a complete DET module during the integration phase.

- **DEM**

  DEM module is not required for integrating the CanTrcv_17_V9251 driver.

- **SchM**

  The CanTrcv_17_V9251 driver does not use any SchM services.

- **Safety error**

  The CanTrcv_17_V9251 driver does not report any safety errors.

- **Notifications and callbacks**

  The CanTrcv_17_V9251 driver does not implement any notifications. However, the CanTrcv_17_V9251 driver notifies the upper layer with the help of the following functions:

  `CanIf_TrcvModeIndication()`: mode change indication to the CanIf layer after successful mode change of the CAN transceiver.

  `EcuM_SetWakeupEvent()`: indication to the EcuM for a valid wake-up from the CAN transceiver

- **OS**

  The OS or the application must ensure correct type of service and interrupt priority is configured in the SR register. Enabling and disabling of interrupts must also be managed by the OS or application. The OS files provided by the MCAL package are only an example code and must be updated by the integrator with the actual OS files for the desired function.

1.1.4.2 Multicore and Resource Manager

The CanTrcv_17_V9251 driver does not support execution on multiple cores simultaneously.

1.1.4.3 MCU support

The CanTrcv_17_V9251 driver is dependent on the MCU driver for the ERU channel allocation and system clock configuration. The initialization of the CanTrcv_17_V9251 driver must be started only after completion of the MCU initialization. The following must be considered while configuring the MCU driver in the EB tresos:

Select the `McuHardwareResourceAllocationConf` container and allocate the ERU input and output channels to the ICU driver from the `McuEruAllocationConf` subcontainer.

The corresponding ERU input and output channels have to be referred in `ERUInputConfiguration` container in the ICU channel, which is configured for wake-up and edge detection.

1.1.4.4 Port support

The PORT driver configures the port pins of the entire microcontroller. The user must configure port pins used by the CanTrcv_17_V9251 driver through the PORT configuration and initialize the port pins prior to invoking of the CanTrcv_17_V9251 driver initialization. The TxD and RxD pins (corresponding to the Rx pin selection made in the CAN driver) of the different CAN controllers must be configured with respective direction and configuration in the PORT driver. The STB pin of the CanTrcv TLE9251V must be configured as INPUT pin in the PORT driver configuration.

1.1.4.5 DMA support

The CanTrcv_17_V9251 driver does not use any service provided by the DMA driver.

1.1.4.6 Interrupt connections

The CanTrcv_17_V9251 driver does not use any interrupt source.

1.1.4.7 Example usage

This section describes how the CanTrcv_17_V9251 driver can be configured and how to use different APIs provided by the driver. All the APIs should be provided with valid input parameters. To detect the invalid function parameters, the DET should be enabled. The behavior of the APIs is undefined if DET is disabled and wrong parameters are passed.

Configuration of the driver

The CanTrcv_17_V9251 driver configuration involves the following steps.

1. In the MCU driver, configure the system clock and allocate ERU channels for the ICU driver.

2. In the Port driver, configure the port pin referred by the CAN transceiver TLE9251V STB as input pin.

3. In the DIO driver, configure the referred port pin to control the CAN transceiver TLE9251V hardware as an individual channel.

4. In the ICU driver, configure the ICU wake-up capable channel to detect the FALLING EDGE of the CAN transceiver TLE9251V RxD pin. This needs ERU channel configuration.

5. The IRQ driver configuration is required to configure the interrupt priorities for interrupts used by the ICU.

6. The MCALLIB driver configuration is required for the timing services used by the CanTrcv_17_V9251 driver.

7. In the EcuM, configure the wake-up source, and the same wake-up source must be configured in the CanTrcv_17_V9251 and the ICU configuration.

8. In the CanTrcv_17_V9251 driver, configure for required channels with Normal or Standby mode, the `CanTrcvWakeupByBusUsed` parameter must be enabled for wake-up support for the corresponding channel.

9. In the CanTrcv_17_V9251 channel configuration, for `CanTrcvIcuChannelRef` parameter, refer to the ICU channel configured for wake up.

10. In the CanTrcv_17_V9251 channel configuration, `CanTrcvDioChannelAccess` parameter must refer to the DIO channel configured for controlling the CAN transceiver TLE9251 RxD pin.

When the CAN transceiver is in the standby mode, if it receives a valid wake-up pattern, the RxD pin of CAN transceiver will change its state from high to low. This falling edge is detected using the ICU module with the help of the ERU, therefore, pin connection should be ensured from the CAN transceiver RxD pin to the ERU input pin configured for the ERU in the ICU channel.

Initialization of CanTrcv_17_V9251 driver

The CanTrcv_17_V9251 driver is dependent on the ICU driver for edge detection. The initialization of the CanTrcv_17_V9251 driver must be started only after completion of the ICU initialization. Since the CAN transceiver TLE9251 supports wake-up only with interrupt mode, the ICU must be put to the sleep mode, and wake up for the corresponding channel should be enabled to support the wake-up functionality.

```
/*MCU Initialization */
Mcu_Init(&Mcu_Config);
Mcu_InitClock(0U);
while(Mcu_GetPllStatus() != MCU_PLL_LOCKED);
Mcu_DistributePllClock ();
/* Port Initialization */
Port_Init(&Port_Config);
/*ICU Initialization */
Icu_17_TimerIp_Init(&Icu_17_TimerIp_Config);
/* CanTrcv_17_V9251 Initialization */
CanTrcv_17_V9251_Init(NULL_PTR);
/* Further APIs of CanTrcv driver can be called now */
```

CAN Transceiver operation mode change:

After the CAN transceiver initialization, the following sequence can be followed for changing the operation mode.

```
/* CanTrcv_17_V9251 operation mode change */
CanTrcv_17_V9251_SetOpMode(0,CANTRCV_TRCVMODE_NORMAL);
```

CAN Transceiver wake-up mode change:

After the CAN transceiver initialization, the following sequence can be followed for changing the wake-up mode.

```
/* CanTrcv_17_V9251 wake-up mode change */
CanTrcv_17_V9251_SetWakeupMode(0,CANTRCV_WUMODE_ENABLE);
```

1.1.5 Key architectural considerations

1.1.5.1 CAN transceiver wake up: only Interrupt mode is supported

The CAN transceiver driver supports the wake up functionality with the help of interrupts generated by ICU driver. Wake up by polling is not supported due to hardware limitations. In the CAN transceiver TLE9251V hardware, the wake-up activity is indicated by the RxD pin. In the standby mode, if the transceiver receives a valid WUP, the RxD pin of the transceiver changes its state from high to low and follows the CAN bus after a delay (less than 5 micro seconds). The RxD pin is connected to the ERU, once the ICU driver gets the wake-up interrupt from the RxD transition from the ERU, the ICU driver informs the wake-up event to the EcuM.

1.1.5.2 User mode is not supported

The CanTrcv_17_V9251 driver does not support the User mode configuration for any of its APIs. Therefore, all the APIs of the driver shall be executed in the Supervisor mode.

[cover parentID CANTRCVV9251={D9AE5D75-3561-47e5-B0E0-49B9D0C1092A}]

1.2 Assumptions of Use (AoU)

There are no AoUs for the CanTrcv_17_V9251 driver.

1.3 Reference information

1.3.1 Configuration interfaces

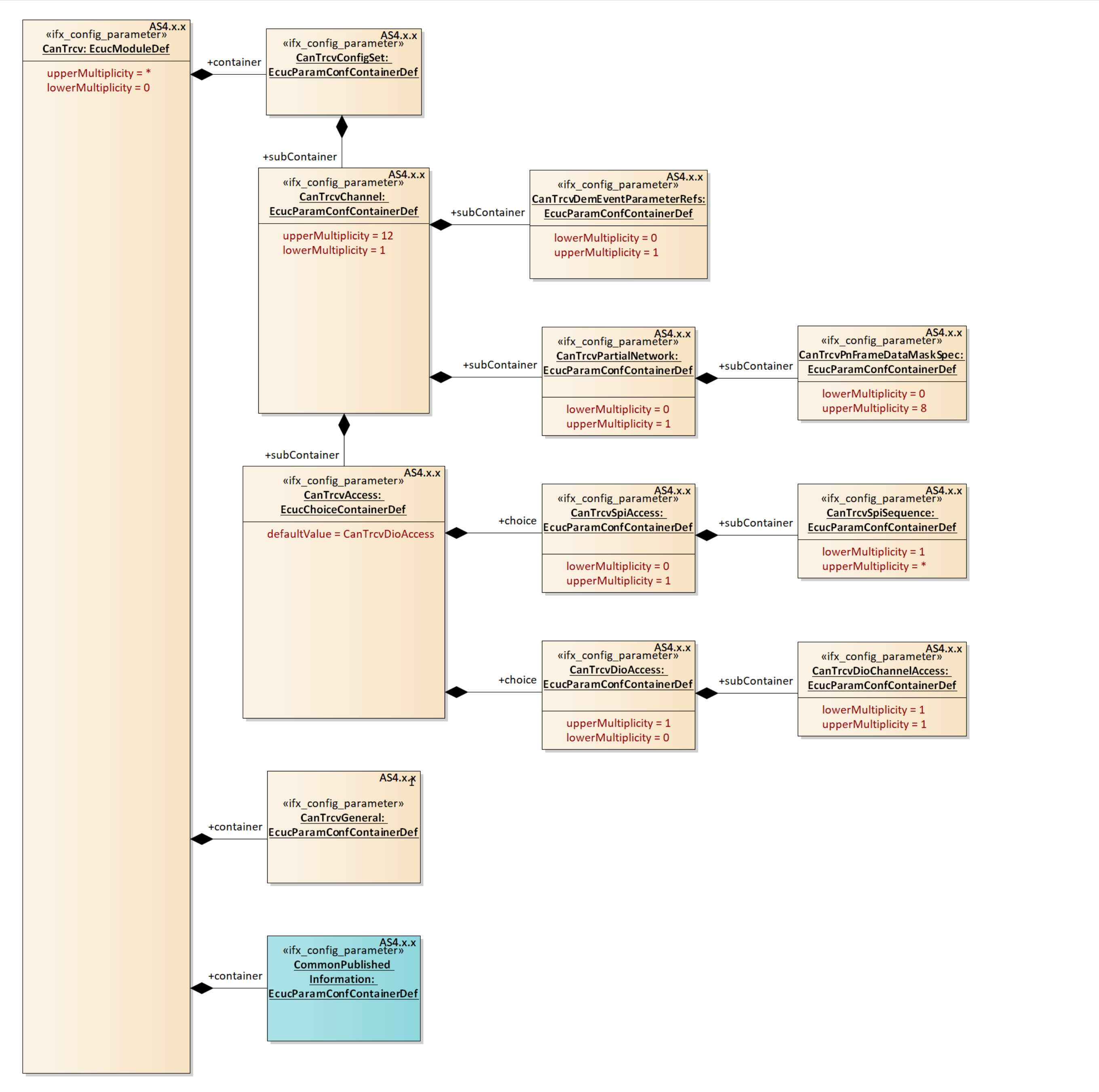

Figure 4 **Container hierarchy along with their configuration parameters**

1.3.1.1 Container: CommonPublished Information

This container contains the common published information of the CanTrcv_17_V9251 driver.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.1.1 ArMajorVersion

Table 4 Specification for ArMajorVersion

Name	`ArMajorVersion`		
Description	This parameter provides the major version of the AUTOSAR specification.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	4		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.2 ArMinorVersion

Table 5 Specification for ArMinorVersion

Name	`ArMinorVersion`		
Description	This parameter provides the minor version of the AUTOSAR specification.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per AUTOSAR minor version		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.3 ArPatchVersion

Table 6 Specification for ArPatchVersion

Name	`ArPatchVersion`
Description	This parameter provides the patch version of the AUTOSAR specification.

Table 6 Specification for ArPatchVersion (continued)

Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per AUTOSAR patch version		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.4 ModuleId

Table 7 Specification for ModuleId

Name	`ModuleId`		
Description	This parameter provides the module Id for the CanTrcv_17_V9251 driver.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 65535		
Default value	70		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.5 Release

Table 8 Specification for Release

Name	`Release`		
Description	This parameter specifies the derivative for which the configuration project is created.		
Multiplicity	1..1	**Type**	EcucStringParamDef
Range	String		
Default value	As per hardware derivative		

Table 8 Specification for Release (continued)

Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.6 SwMajorVersion

Table 9 Specification for SwMajorVersion

Name	`SwMajorVersion`		
Description	This parameter provides the software major version of the CanTrcv_17_V9251 driver.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per driver		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.7 SwMinorVersion

Table 10 Specification for SwMinorVersion

Name	`SwMinorVersion`		
Description	This parameter provides the software minor version of the CanTrcv_17_V9251 driver.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per driver		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-

Table 10 Specification for SwMinorVersion (continued)

Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.8 SwPatchVersion

Table 11 Specification for SwPatchVersion

Name	`SwPatchVersion`		
Description	This parameter provides the software patch version of the CanTrcv_17_V9251 driver.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per driver		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.9 VendorApiInfix

Table 12 Specification for VendorApiInfix

Name	`VendorApiInfix`		
Description	This parameter is used to specify the vendor specific name of the CanTrcv_17_V9251 driver.		
Multiplicity	1..1	**Type**	EcucStringParamDef
Range	String		
Default value	V9251		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL

Table 12 Specification for VendorApiInfix (continued)

Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.10 VendorId

Table 13 Specification for VendorId

Name	`VendorId`		
Description	This parameter provides the vendor Id for CanTrcv_17_V9251 driver.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 65535		
Default value	17		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2 Container: CanTrcv

Specifies the configuration of the CAN Transceiver driver module.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: -

1.3.1.2.1 Config Variant

Table 14 Specification for Config Variant

Name	`Config Variant`		
Description	This parameter indicates the selected configuration variant for CanTrcv_17_V9251 driver.		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	VariantPreCompile: The CanTrcv_17_V9251 driver supports only Pre- Compile variant.		
Default value	VariantPreCompile		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-

Table 14 **Specification for Config Variant (continued)**

Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.3 Container: CanTrcvConfigSet

This container contains the configuration parameters and sub containers of the AUTOSAR CAN transceiver module.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.3.1 CanTrcvSPICommRetries

Table 15 **Specification for CanTrcvSPICommRetries**

Name	`CanTrcvSPICommRetries`		
Description	This parameter indicates the maximum number of communication retries in case of a failed SPI communication (applies both to timed out communication and to errors/NACK in the response data). If configured value is '0', no retry is allowed (communication is expected to succeed at first try). *Note: Since CAN transceiver TLE9251V does not support SPI interface, this parameter is not supported and made non-editable. This parameter is kept only for AUTOSAR schema compatibility.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	0		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.3.2 CanTrcvSPICommTimeout

Table 16 Specification for CanTrcvSPICommTimeout

Name	`CanTrcvSPICommTimeout`		
Description	This parameter indicates the maximum time allowed to the CAN transceiver for replying (either positively or negatively) to a SPI command. Timeout is configured in milliseconds. Timeout value of '0' means that no specific timeout is to be used by CAN transceiver and the communication is executed at the best of the SPI HW capacity. *Note: Since CAN transceiver TLE9251V does not support SPI interface, this parameter is not supported and made non-editable. This parameter is kept only for AUTOSAR schema compatibility.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 100		
Default value	0		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4 Container: CanTrcvChannel

This container gives CAN transceiver driver information for a single CAN transceiver channel.

This container have a lower multiplicity of 1 and upper multiplicity of 12.

Post-Build Variant Multiplicity: FALSE
Multiplicity Configuration Class: Pre-Compile

1.3.1.4.1 CanTrcvAccess

Table 17 Specification for CanTrcvAccess

Name	`CanTrcvAccess`		
Description	This container gives CAN transceiver driver information about access to a single CAN transceiver. *Note: CanTrcv_17_V9251 supports only DIO Interface.*		
Multiplicity	1..1	**Type**	EcucChoiceContainer Def
Range	None		

Table 17 Specification for CanTrcvAccess (continued)

Default value	CanTrcvDioAccess		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.2 CanTrcvChannelEcucPartitionRef

Table 18 Specification for CanTrcvChannelEcucPartitionRef

Name	`CanTrcvChannelEcucPartitionRef`		
Description	Parameter maps the CAN transceiver channel to zero or one ECUC partitions. The ECUC partition referenced is a subset of the ECUC partitions where the CAN transceiver driver is mapped to. *Note: Parameter support is added only for AUTOSAR schema compliance. This parameter is not used in code generation logic, hence this parameter is made editable false.*		
Multiplicity	0..1	**Type**	EcucReferenceDef
Range	Reference to Node: EcucPartition		
Default value	NULL		
Post-build variant value	FALSE	**Post-build variant multiplicity**	FALSE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	ECU
Dependency	-		
Autosar Version	Applicable for Autosar version 4.4.0.		

1.3.1.4.3 CanTrcvChannelId

Table 19 Specification for CanTrcvChannelId

Name	`CanTrcvChannelId`
Description	This parameter specifies unique identifier of the CAN transceiver channel. *Note:* *- The channel Id should be less than the number of channels configured. Zero is selected as the default value.*

Table 19 Specification for CanTrcvChannelId (continued)

	- If the channel Ids are not unique, the user will get a configuration error. *- As per AUTOSAR, the range of this parameter is 0 to 255. Since AURIX TC3xx CAN controller has 12 nodes, CanTrcv_17_V9251 driver supports only 12 channels and the range is from 0 to 11.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 11		
Default value	0		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	ECU
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.4 CanTrcvChannelUsed

Table 20 Specification for CanTrcvChannelUsed

Name	`CanTrcvChannelUsed`		
Description	This parameter specifies if the configured channel is used or not. *Note: This parameter is used to enable/disable the configured channel.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	TRUE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.5 CanTrcvControlsPowerSupply

Table 21 Specification for CanTrcvControlsPowerSupply

Name	`CanTrcvControlsPowerSupply`		
Description	This parameter indicates the ECU power supply controlling method. TRUE = Controlled by transceiver. FALSE = Not controlled by transceiver. *Note: Since CAN transceiver TLE9251V does not control the ECU power supply, this parameter is set FALSE and made non-editable.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.6 CanTrcvHwPnSupport

Table 22 Specification for CanTrcvHwPnSupport

Name	`CanTrcvHwPnSupport`		
Description	This parameter indicates whether the CAN transceiver supports the selective wake-up function TRUE = Selective wake up feature is supported by the transceiver FALSE = Selective wake up feature is not available by the transceiver *Note: Since CAN transceiver TLE9251V does not support selective wake up functionality, the default value of this parameter is set to false and made non-editable.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-

Table 22 Specification for CanTrcvHwPnSupport (continued)

Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.7 CanTrcvIcuChannelRef

Table 23 Specification for CanTrcvIcuChannelRef

Name	`CanTrcvIcuChannelRef`		
Description	Reference to the ICU channel for detecting the wakeups. This parameter is disabled when the configuration parameter CanTrcvWakeupByBusUsed is set to FALSE. *Note: Since the name of the dependent parameter is user configurable, the default value is set to NULL.*		
Multiplicity	0..1	**Type**	EcucReferenceDef
Range	Reference to Node: IcuChannel		
Default value	NULL		
Post-build variant value	FALSE	**Post-build variant multiplicity**	FALSE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	CanTrcvWakeupByBusUsed		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.8 CanTrcvInitState

Table 24 Specification for CanTrcvInitState

Name	`CanTrcvInitState`		
Description	This parameter specifies the state of the CAN transceiver after call to CanTrcv_17_V9251_Init API. *Note: CAN transceiver TLE9251V supports only normal and standby modes. Normal mode is set as default value assuming user expects CAN transceiver to work in normal mode. User is allowed to change the mode after initialization through the configuration parameter CanTrcvInitState.*		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef

Table 24 Specification for CanTrcvInitState (continued)

Range	CANTRCV_17_V9251_OP_MODE_NORMAL: Normal operation mode. CANTRCV_17_V9251_OP_MODE_STANDBY: Standby operation mode.		
Default value	CANTRCV_17_V9251_OP_MODE_NORMAL		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.9 CanTrcvMaxBaudrate

Table 25 Specification for CanTrcvMaxBaudrate

Name	`CanTrcvMaxBaudrate`		
Description	This parameter specifies the max baud rate supported by the CAN transceiver. Value shall be configured by configuration tool based on the transceiver hardware type. *Note: Default value is the maximum baud rate supported by the CAN transceiver. The baud rate will be in kbps. The baud rate range exceeds the AUTOSAR specified range. This parameter does not have any significance and it gives the information on maximum baud rate supported, so this parameter is not used anywhere in the implemented design.* *Note: For Autosar 4.2.2, the range and default value of this parameter is modified. Range is extended to 5Mbps since the hardware supports CAN FD data rates upto 5Mbps.* *Note: For Autosar 4.4.0, the range of this parameter is extended to 12Mbps. But, the default value is set to 5Mbps due to hardware constraints.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 5000		
Default value	5000		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.10 CanTrcvPorWakeupSourceRef

Table 26 Specification for CanTrcvPorWakeupSourceRef

Name	`CanTrcvPorWakeupSourceRef`		
Description	This parameter specifies the symbolic name reference to indicate the wake up sources configured to report the wake up source events. This reference is mandatory if the CAN transceiver supports POR flag. Since the name of the dependent parameter is user configurable, the default value is set to NULL. *Note: Since CAN transceiver TLE9251V does not support POR detection, this parameter is not supported and made non-editable. This configuration parameter is not used in the code but it is listed for AUTOSAR compatibility.*		
Multiplicity	0..1	**Type**	EcucSymbolicNameReferenceDef
Range	Reference to Node: EcuMWakeupSource		
Default value	NULL		
Post-build variant value	FALSE	**Post-build variant multiplicity**	FALSE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	ECU
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.11 CanTrcvSyserrWakeupSourceRef

Table 27 Specification for CanTrcvSyserrWakeupSourceRef

Name	`CanTrcvSyserrWakeupSourceRef`		
Description	This parameter specifies the symbolic name reference to indicate the wake up sources configured to report the wake up source events. This reference is mandatory if the CAN transceiver supports SYSERR flag. Since the name of the dependent parameter is user configurable, the default value is set to NULL. *Note: Since CAN transceiver TLE9251V does not support SYSERR detection, this parameter is not supported and made non-editable. This configuration parameter is not used in the code but it is listed for AUTOSAR compatibility.*		
Multiplicity	0..1	**Type**	EcucSymbolicNameReferenceDef
Range	Reference to Node: EcuMWakeupSource		
Default value	NULL		

Table 27 Specification for CanTrcvSyserrWakeupSourceRef (continued)

Post-build variant value	FALSE	**Post-build variant multiplicity**	FALSE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	ECU
Dependency	CanTrcvWakeupByBusUsed		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.12 CanTrcvWakeupByBusUsed

Table 28 Specification for CanTrcvWakeupByBusUsed

Name	`CanTrcvWakeupByBusUsed`		
Description	This parameter indicates whether wake up by bus is supported or not. If CAN transceiver hardware does not support wake up by bus, value is always FALSE. If CAN transceiver hardware supports wake up by bus, value is TRUE or FALSE depending on whether it is used or not. *Note: Since CAN transceiver TLE9251V supports wake up only by bus, user can use this parameter to enable or disable the wake up functionality of CAN transceiver.* *TRUE = Is used and wake up functionality is supported for respective channel.* *FALSE = Is not used and wake up functionality is not supported for respective channel.* *If CanTrcvWakeupByBusUsed is FALSE, then user is not allowed to configure wake up related configuration parameters like CanTrcvWakeupSourceRef and CanTrcvIcuChannelRef.* *Since CanTrcv_17_V9251 driver depends on this parameter, this parameter is always needed, hence the lower multiplicity of this parameter is set to 1.* *Since CAN transceiver TLE9251V supports wake up functionality, this parameter does not depend on CanTrcvWakeUpSupport parameter.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	CanTrcvWakeUpSupport		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.13 CanTrcvWakeupSourceRef

Table 29 **Specification for CanTrcvWakeupSourceRef**

Name	`CanTrcvWakeupSourceRef`		
Description	This parameter is a reference to a wake up source configured in the EcuM configuration. This reference is only needed when CanTrcvWakeupByBusUsed is true. *Note: Since the name of the dependent parameter is user configurable, the default value is set to NULL.* *This parameter is made non editable when CanTrcvWakeupByBusUsed is configured as FALSE.*		
Multiplicity	0..1	**Type**	EcucReferenceDef
Range	Reference to Node: EcuMWakeupSource		
Default value	NULL		
Post-build variant value	FALSE	**Post-build variant multiplicity**	FALSE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	ECU
Dependency	CanTrcvWakeupByBusUsed		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5 Container: CanTrcvDemEventParameterRefs

This container contains the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus in case the corresponding error occurs. The Event Id is taken from the referenced DemEventParameter's DemEventId value.

Note: Since CAN transceiver TLE9251V does not support production errors, this container is not supported. This container is kept only for AUTOSAR schema compatibility.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: Pre-Compile

1.3.1.5.1 CANTRCV_E_BUS_ERROR

Table 30 **Specification for CANTRCV_E_BUS_ERROR**

Name	`CANTRCV_E_BUS_ERROR`		
Description	Reference to the DemEventParameter which will be issued when bus error has occurred. *Note: Since CAN transceiver TLE9251V does not support production errors this configuration parameter is not supported and made non-editable. This parameter is kept only for AUTOSAR schema compatibility.*		
Multiplicity	0..1	**Type**	EcucReferenceDef
Range	Reference to Node: DemEventParameter		

Table 30 Specification for CANTRCV_E_BUS_ERROR (continued)

Default value	NULL		
Post-build variant value	FALSE	**Post-build variant multiplicity**	FALSE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.6 Container: CanTrcvDioAccess

This container gives CAN transceiver driver information about accessing ports and port pins. In addition relation between CAN transceiver hardware pin names and DIO port access information is given.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: Pre-Compile

1.3.1.7 Container: CanTrcvDioChannelAccess

This container gives DIO channel access by single CAN transceiver channel.

The lower multiplicity of this container is 1.

The upper multiplicity of this container is restricted to 1 since one DIO channel is enough to refer STB pin of one transceiver channel.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: -

1.3.1.7.1 CanTrcvDioSymNameRef

Table 31 Specification for CanTrcvDioSymNameRef

Name	`CanTrcvDioSymNameRef`		
Description	This parameter gives the symbolic name reference to a configured DIO Port, DIO channel or DIO channel group. *Note: CanTrcv_17_V9251 driver supports reference only to a DIO Channel.* *Note: CanTrcvDioSymNameRef should be unique.* *Note: If the symbolic name references are not unique, the user will get a configuration error.*		
Multiplicity	1..1	**Type**	EcucChoiceReferenceDef
Range	Reference to Node: DioChannel		
Default value	NULL		

Table 31 **Specification for CanTrcvDioSymNameRef (continued)**

Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.2 CanTrcvHardwareInterfaceName

Table 32 **Specification for CanTrcvHardwareInterfaceName**

Name	`CanTrcvHardwareInterfaceName`		
Description	This parameter specifies CAN transceiver hardware interface name. It is typically the name of a CAN transceiver pin. *Note: Since CanTrcv_17_V9251 driver uses STB pin of CAN transceiver TLE9251V hardware for mode control, STB is the default name set for this parameter and made non-editable.*		
Multiplicity	1..1	**Type**	EcucStringParamDef
Range	String		
Default value	STB		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8 Container: CanTrcvGeneral

This container gives CAN transceiver driver basic information.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.8.1 CanTrcvDevErrorDetect

Table 33 **Specification for CanTrcvDevErrorDetect**

Name	`CanTrcvDevErrorDetect`

Table 33 **Specification for CanTrcvDevErrorDetect (continued)**

Description	Parameter enables or disables the Default Error Tracer (DET) detection and reporting. TRUE: Detection and reporting is enabled. FALSE: Detection and reporting is disabled. *Note: The default value of this parameter is set to false to minimize the executable code size.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.2 CanTrcvEcucPartitionRef

Table 34 **Specification for CanTrcvEcucPartitionRef**

Name	`CanTrcvEcucPartitionRef`		
Description	Parameter maps the CAN transceiver driver to zero or multiple ECUC partitions to make the modules API available in this partition. The module will operate as an independent instance in each of the partitions. *Note: Parameter support is added only for AUTOSAR schema compliance. This parameter is not used in code generation logic, hence this parameter is made editable false.*		
Multiplicity	0..*	**Type**	EcucReferenceDef
Range	Reference to Node: EcucPartition		
Default value	NULL		
Post-build variant value	FALSE	**Post-build variant multiplicity**	FALSE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	ECU
Dependency	-		
Autosar Version	Applicable for Autosar version 4.4.0.		

1.3.1.8.3 CanTrcvGetVersionInfo

Table 35 Specification for CanTrcvGetVersionInfo

Name	`CanTrcvGetVersionInfo`		
Description	Parameter adds or removes the CanTrcv_17_V9251_GetVersionInfo API from the code. *Note: The default value of this parameter is set to false to minimize the executable code size.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar version 4.2.2.		

1.3.1.8.4 CanTrcvIndex

Table 36 Specification for CanTrcvIndex

Name	`CanTrcvIndex`		
Description	This parameter specifies the instance Id of the CanTrcv_17_V9251 module. *Note: Since only one instance is supported, by default it shall have the Id 0.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	0		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.5 CanTrcvMainFunctionDiagnosticsPeriod

Table 37 Specification for CanTrcvMainFunctionDiagnosticsPeriod

Name	`CanTrcvMainFunctionDiagnosticsPeriod`		
Description	This parameter describes the period for cyclic call to CanTrcv_MainFunctionDiagnostics. Unit of this parameter is in seconds. *Note: Since CanTrcv_17_V9251_MainFunctionDiagnostics API is not provided by the driver, this parameter is not applicable and made non-editable. This configuration parameter is not used in the code but is listed for AUTOSAR compatibility.* *Note: As per AUTOSAR 4.2.2, range of CanTrcvMainFunctionDiagnosticsPeriod is 0.001 - 65.535.* *Note: As per AUTOSAR 4.4.0, range of CanTrcvMainFunctionDiagnosticsPeriod is 0 - Infinity.*		
Multiplicity	0..1	**Type**	EcucFloatParamDef
Range	0.001 - 65.535		
Default value	0.005		
Post-build variant value	FALSE	**Post-build variant multiplicity**	FALSE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.6 CanTrcvMainFunctionPeriod

Table 38 Specification for CanTrcvMainFunctionPeriod

Name	`CanTrcvMainFunctionPeriod`		
Description	This parameter describes the period for cyclic call to CanTrcv_17_V9251_MainFunction. Unit of this parameter is in seconds. *Note: Since CAN transceiver TLE9251V does not support polling mode, this parameter is not supported and made non-editable. This parameter is kept only for AUTOSAR schema compatibility.*		
Multiplicity	0..1	**Type**	EcucFloatParamDef
Range	0.001 - 65.535		
Default value	0.005		
Post-build variant value	FALSE	**Post-build variant multiplicity**	FALSE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	LOCAL

Table 38 **Specification for CanTrcvMainFunctionPeriod (continued)**

Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.7 CanTrcvTimerType

Table 39 **Specification for CanTrcvTimerType**

Name	`CanTrcvTimerType`		
Description	This parameter specifies the type of the timer service used in the CAN transceiver driver. *Note: Default value of this parameter is set to 'None' since McalLib APIs are used to realize wait time. The parameter is made non-editable.*		
Multiplicity	0..1	**Type**	EcucEnumerationParamDef
Range	None: No timer type is used. Timer_1us16bit: Specifies 16 bit 1us timer type.		
Default value	None		
Post-build variant value	FALSE	**Post-build variant multiplicity**	FALSE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.8 CanTrcvVersionInfoApi

Table 40 **Specification for CanTrcvVersionInfoApi**

Name	`CanTrcvVersionInfoApi`		
Description	Parameter adds or removes the CanTrcv_17_V9251_GetVersionInfo API from the code. *Note: The default value of this parameter is set to false to minimize the executable code size.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-

Table 40 **Specification for CanTrcvVersionInfoApi (continued)**

Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar version 4.4.0.		

1.3.1.8.9 CanTrcvWaitTime

Table 41 **Specification for CanTrcvWaitTime**

Name	`CanTrcvWaitTime`		
Description	This parameter specifies the wait time for transceiver state changes in seconds. *Note: The maximum time for mode change by CAN transceiver TLE9251V is 20us. Hence, default value of this parameter is set to 20us and made non editable.*		
Multiplicity	1..1	**Type**	EcucFloatParamDef
Range	0.000020 - 0.000255		
Default value	0.000020		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.10 CanTrcvWakeUpSupport

Table 42 **Specification for CanTrcvWakeUpSupport**

Name	`CanTrcvWakeUpSupport`		
Description	This parameter informs the mode of wake up support. *Note:- Since wake up feature of CAN transceiver TLE9251V is supported by the interrupt of ICU module, CANTRCV_17_V9251_WAKEUP_BY_POLLING and CANTRCV_17_V9251_WAKE_UP_NOT_SUPPORTED options are not supported.* *Note: A new option CANTRCV_WAKEUP_BY_INTERRUPT is added and set as default value, which is not-editable.*		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef

Table 42 Specification for CanTrcvWakeUpSupport (continued)

Range	CANTRCV_17_V9251_WAKEUP_BY_INTERRUPT: Wake up by Interrupt.		
Default value	CANTRCV_17_V9251_WAKEUP_BY_INTERRUPT		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.9 Container: CanTrcvPartialNetwork

This container gives CAN transceiver driver information about the configuration of partial networking functionality.

Note: Since CAN transceiver TLE9251V does not support partial networking, this configuration container is not supported and made non-editable. This configuration container and its parameters are not used in the code but are listed for AUTOSAR compatibility.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: Pre-Compile

1.3.1.9.1 CanTrcvBaudRate

Table 43 Specification for CanTrcvBaudRate

Name	`CanTrcvBaudRate`		
Description	This parameter indicates the CAN bus communication baud rate in kbps. *Note: Since CAN transceiver TLE9251V does not control the baud rate, this parameter is not supported and made non-editable.* *Note: As per AUTOSAR 4.2.2, range of CanTrcvBaudRate is 0-1000.* *Note: As per AUTOSAR 4.4.0, range of CanTrcvBaudRate is 0-12000.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 1000		
Default value	500		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL

Table 43 Specification for CanTrcvBaudRate (continued)

Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.9.2 CanTrcvBusErrFlag

Table 44 Specification for CanTrcvBusErrFlag

Name	`CanTrcvBusErrFlag`		
Description	Indicates if the Bus Error (BUSERR) flag is managed by the BSW. This flag is set if a bus failure is detected by the transceiver. TRUE = Supported by transceiver and managed by BSW. FALSE = Not managed by BSW. *Note: Since CAN transceiver TLE9251V does not support partial networking, this configuration parameter is not supported and made non-editable.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.9.3 CanTrcvPnCanIdIsExtended

Table 45 Specification for CanTrcvPnCanIdIsExtended

Name	`CanTrcvPnCanIdIsExtended`		

Description	This parameter indicates whether extended or standard ID is used.
	TRUE = Extended CAN identifier is used
	FALSE = Standard CAN identifier is used
	Note: Since CAN transceiver TLE9251V does not support partial networking, this configuration parameter is not supported and made non-editable.		

Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE		

Table 45 Specification for CanTrcvPnCanIdIsExtended (continued)

	FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.9.4 CanTrcvPnEnabled

Table 46 Specification for CanTrcvPnEnabled

Name	`CanTrcvPnEnabled`		
Description	This parameter indicates whether the selective wake-up feature is enabled or disabled in the CAN transceiver hardware. TRUE = Selective wakeup feature is enabled in the transceiver hardware FALSE = Selective wakeup feature is disabled in the transceiver hardware *Note: Since CAN transceiver TLE9251V hardware does not support partial networking, this configuration parameter is not supported and made non-editable.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.9.5 CanTrcvPnFrameCanId

Table 47 Specification for CanTrcvPnFrameCanId

Name	`CanTrcvPnFrameCanId`		
Description	This parameter indicates the CAN ID of the Wake-up Frame (WUF). *Note: Since CAN transceiver TLE9251V does not support partial networking, this configuration parameter is not supported and made non-editable.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 4294967295		
Default value	0		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.9.6 CanTrcvPnFrameCanIdMask

Table 48 Specification for CanTrcvPnFrameCanIdMask

Name	`CanTrcvPnFrameCanIdMask`		
Description	This parameter indicates ID mask for the selective activation of the transceiver. It is used to enable Frame Wake-up (WUF) on a group of IDs. *Note: Since CAN transceiver TLE9251V does not support partial networking, this configuration parameter is not supported and made non-editable.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 4294967295		
Default value	0		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.9.7 CanTrcvPnFrameDlc

Table 49 Specification for CanTrcvPnFrameDlc

Name	`CanTrcvPnFrameDlc`		
Description	This parameter specifies the data length of the Wake-up Frame (WUF). Default value is set to 1 as it is the minimum value supported. Although WUF with DLC=0 is technically possible, it is explicitly not wanted. *Note: Since CAN transceiver TLE9251V does not support partial networking, this configuration parameter is not supported and made non-editable.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 8		
Default value	1		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.9.8 CanTrcvPowerOnFlag

Table 50 Specification for CanTrcvPowerOnFlag

Name	`CanTrcvPowerOnFlag`		
Description	This parameter indicates if the Power On Reset (POR) flag is available and is managed by the transceiver. TRUE = Supported by hardware FALSE = Not supported by hardware *Note: Since CAN transceiver TLE9251V does not support partial networking, this configuration parameter is not supported and made non-editable.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-

Table 50 **Specification for CanTrcvPowerOnFlag (continued)**

Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.10 Container: CanTrcvPnFrameDataMaskSpec

This parameter defines data payload mask to be used on the received payload in order to determine if the transceiver must be woken up by the received Wake-up Frame (WUF).

Note: Since CAN transceiver TLE9251V does not support partial networking, this configuration parameter is not supported and made non-editable. This configuration container and its parameters are not used in the code but is listed for AUTOSAR compatibility.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: Pre-Compile

1.3.1.10.1 CanTrcvPnFrameDataMask

Table 51 **Specification for CanTrcvPnFrameDataMask**

Name	`CanTrcvPnFrameDataMask`		
Description	This parameter defines the n byte (Byte0 = LSB) of the data payload mask to be used on the received payload in order to determine if the transceiver must be woken up by the received Wake-up Frame (WUF). *Note: Since CAN transceiver TLE9251V does not support partial networking, this configuration parameter is not supported and made non-editable.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	0		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.10.2 CanTrcvPnFrameDataMaskIndex

Table 52 **Specification for CanTrcvPnFrameDataMaskIndex**

Name	`CanTrcvPnFrameDataMaskIndex`
Description	This parameter holds the position n in frame of the data mask-part.

Table 52 Specification for CanTrcvPnFrameDataMaskIndex (continued)

	Note: Since CAN transceiver TLE9251V does not support partial networking, this configuration parameter is not supported and made non-editable.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 7		
Default value	0		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.11 Container: CanTrcvSpiAccess

This container gives CAN transceiver driver information about accessing SPI. If CAN transceiver hardware has no SPI interface, there is no instance of this container.

Note: Since CAN transceiver TLE9251V hardware supports only DIO interface, this container is not supported.

Note: This configuration container and its sub-containers and parameters are not used in the code but is listed for AUTOSAR compatibility.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: Pre-Compile

1.3.1.12 Container: CanTrcvSpiSequence

This container gives CAN transceiver driver information about one SPI sequence.

Note: Since CAN transceiver TLE9251V hardware supports only DIO interface, this container is not supported.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: -

1.3.1.12.1 CanTrcvSpiAccessSynchronous

Table 53 Specification for CanTrcvSpiAccessSynchronous

Name	`CanTrcvSpiAccessSynchronous`
Description	This parameter is used to define whether the access to the SPI sequence is synchronous or asynchronous. TRUE: SPI access is synchronous. FALSE: SPI access is asynchronous.

Table 53 Specification for CanTrcvSpiAccessSynchronous (continued)

	Note: Since CAN transceiver TLE9251V supports only DIO interface, this parameter is not supported and made non-editable.		
Multiplicity	0..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	FALSE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.12.2 CanTrcvSpiSequenceName

Table 54 Specification for CanTrcvSpiSequenceName

Name	`CanTrcvSpiSequenceName`		
Description	This parameter specifies the reference to an SPI sequence configuration container. *Note: Since CAN transceiver TLE9251V hardware supports only DIO interface, this parameter is not supported and made non-editable.*		
Multiplicity	0..*	**Type**	EcucSymbolicNameReferenceDef
Range	Reference to Node: SpiSequence		
Default value	NULL		
Post-build variant value	FALSE	**Post-build variant multiplicity**	FALSE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	SpiSequence		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.2 Functions - Type definitions

This section lists all the Datatype of the CanTrcv_17_V9251 driver.

1.3.3 Functions - APIs

This section lists all the APIs of the CanTrcv_17_V9251 driver.

1.3.3.1 CanTrcv_17_V9251_Init

Table 55 Specification for `CanTrcv_17_V9251_Init` API

Syntax	`void  CanTrcv_17_V9251_Init`
	`(`
	`    const CanTrcv_17_V9251_ConfigType * const ConfigPtr`
	`)`	

Service ID	0x00	
Sync/Async	Synchronous	
ASIL Level	QM	
Re-entrancy	Non Reentrant	
Parameters (in)	ConfigPtr	Pointer to driver configuration. Note: Since CanTrcv_17_V9251 is pre-compile module, null pointer must be passed as the parameter by the caller of this API.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	void	-
Description	This API initializes all the connected CAN transceivers by setting CAN transceiver hardware to the mode configured by the configuration parameter CanTrcvInitState. The CAN Transceiver driver initialization status is set at the end of the initialization function execution. *Note: Since CanTrcv_17_V9251 module is a pre-compile module, NULL_PTR must be passed as the parameter for CanTrcv_17_V9251_Init API.*	
Source	AUTOSAR	
Error handling	CANTRCV_17_V9251_E_INIT_FAILED	
Configuration dependencies	-	
User hints	None	
SFR accessed	-	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.2 CanTrcv_17_V9251_SetOpMode

Table 56 Specification for CanTrcv_17_V9251_SetOpMode API

Syntax	Std_ReturnType CanTrcv_17_V9251_SetOpMode (const uint8 Transceiver, const CanTrcv_TrcvModeType OpMode)	
Service ID	0x01	
Sync/Async	Synchronous	
ASIL Level	QM	
Re-entrancy	Reentrant for different transceivers	
Parameters (in)	Transceiver	CAN transceiver to which API call has to be made. Note: CanTrcv_17_V9251 driver supports 12 channels, so the range of this parameter must be 0 to 11.
	OpMode	This parameter contains the desired operating mode. Note: CANTRCV_TRCVMODE_NORMAL and CANTRCV_TRCVMODE_STANDBY modes are supported by CanTrcv_17_V9251 driver.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK: CAN Transceiver state has been changed to the requested mode. E_NOT_OK: CAN Transceiver state change has failed or the parameter is out of the allowed range. The previous state has not been changed.
Description	This API sets the mode of the CAN transceiver to the value given by OpMode.	
Source	AUTOSAR	
Error handling	CANTRCV_17_V9251_E_UNINIT, CANTRCV_17_V9251_E_INVALID_TRANSCEIVER, CANTRCV_17_V9251_E_PARAM_TRCV_OPMODE	
Configuration dependencies	-	
User hints	None	
SFR accessed	-	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.3 CanTrcv_17_V9251_GetOpMode

Table 57 Specification for `CanTrcv_17_V9251_GetOpMode` API

Syntax	`Std_ReturnType  CanTrcv_17_V9251_GetOpMode`
	`(`
	`    const uint8 Transceiver,`
	`    CanTrcv_TrcvModeType * const OpMode`
	`)`	

Service ID	0x02	
Sync/Async	Synchronous	
ASIL Level	QM	
Re-entrancy	Reentrant	
Parameters (in)	Transceiver	CAN transceiver to which API call has to be made. Note: CanTrcv_17_V9251 driver supports 12 channels, so the range of this parameter must be 0 to 11.
Parameters (out)	OpMode	Pointer to operation mode of the CAN transceiver.
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK: Operation mode read successfully. E_NOT_OK: Operation mode was not detected or DET error.
Description	This API reads the mode of the CAN transceiver and returns it in the parameter OpMode.	
Source	AUTOSAR	
Error handling	CANTRCV_17_V9251_E_UNINIT, CANTRCV_17_V9251_E_INVALID_TRANSCEIVER, CANTRCV_17_V9251_E_PARAM_POINTER	
Configuration dependencies	-	
User hints	None	
SFR accessed	-	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.4 CanTrcv_17_V9251_GetBusWuReason

Table 58 Specification for `CanTrcv_17_V9251_GetBusWuReason` API

Syntax	`Std_ReturnType  CanTrcv_17_V9251_GetBusWuReason`
	`(`
	`    const uint8 Transceiver,`
	`    CanTrcv_TrcvWakeupReasonType * const reason`
	`)`

Service ID	0x03

Table 58 **Specification for `CanTrcv_17_V9251_GetBusWuReason` API (continued)**

Sync/Async	Synchronous	
ASIL Level	QM	
Re-entrancy	Reentrant	
Parameters (in)	Transceiver	CAN transceiver to which API call has to be made. Note: CanTrcv_17_V9251 driver supports 12 channels, so the range of this parameter must be 0 to 11.
Parameters (out)	reason	Pointer to wake up reason of the CAN transceiver. Note: Only CANTRCV_WU_POWER_ON, CANTRCV_WU_BY_BUS and CANTRCV_WU_INTERNALLY values are supported by the transceiver hardware.
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK: Transceiver wakeup reason was provided successfully. E_NOT_OK: If no wake up reason is available or if the service request failed due to development errors.
Description	This API reads the wakeup reason of the CAN transceiver and returns it in the parameter reason.	
Source	AUTOSAR	
Error handling	CANTRCV_17_V9251_E_UNINIT, CANTRCV_17_V9251_E_INVALID_TRANSCEIVER, CANTRCV_17_V9251_E_PARAM_POINTER	
Configuration dependencies	-	
User hints	None	
SFR accessed	-	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.5 CanTrcv_17_V9251_GetVersionInfo

Table 59 **Specification for `CanTrcv_17_V9251_GetVersionInfo` API**

Syntax	`void  CanTrcv_17_V9251_GetVersionInfo`
	`(`
	`    Std_VersionInfoType * const versioninfo`
	`)`

Service ID	0x04
Sync/Async	Synchronous
ASIL Level	QM
Re-entrancy	Reentrant

Table 59 Specification for `CanTrcv_17_V9251_GetVersionInfo` API (continued)

Parameters (in)	-	-
Parameters (out)	versioninfo	Pointer to version information of the CanTrcv_17_V9251 module.
Parameters (in - out)	-	-
Return	void	-
Description	This API reads the version of the CanTrcv_17_V9251 module and returns it in the parameter versionInfo. *Note: For AUTOSAR 4.2.2, this API depends on the configuration parameter CanTrcvGetVersionInfo.* *Note: For AUTOSAR 4.4.0, this API depends on the configuration parameter CanTrcvVersionInfoApi.*	
Source	AUTOSAR	
Error handling	CANTRCV_17_V9251_E_PARAM_POINTER	
Configuration dependencies	CanTrcvVersionInfoApi,CanTrcvGetVersionInfo	
User hints	None	
SFR accessed	-	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.6 CanTrcv_17_V9251_SetWakeupMode

Table 60 Specification for `CanTrcv_17_V9251_SetWakeupMode` API

Syntax	`Std_ReturnType CanTrcv_17_V9251_SetWakeupMode`
	`(`
	`    const uint8 Transceiver,`
	`    const CanTrcv_TrcvWakeupModeType TrcvWakeupMode`
	`)`	

Service ID	0x05	
Sync/Async	Synchronous	
ASIL Level	QM	
Re-entrancy	Reentrant for different transceivers	
Parameters (in)	Transceiver TrcvWakeupMode	CAN transceiver to which API call has to be made. Note: CanTrcv_17_V9251 driver supports 12 channels, so the range of this parameter must be 0 to 11. Requested CAN transceiver wakeup mode. Note: The supported wake up modes are

Table 60 **Specification for `CanTrcv_17_V9251_SetWakeupMode` API (continued)**

		CANTRCV_WUMODE_ENABLE, CANTRCV_WUMODE_DISABLE and CANTRCV_WUMODE_CLEAR.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK: Wakeup state has changed to the requested mode E_NOT_OK: Wakeup state change has failed or the parameter is out of the allowed range. The previous state has not been changed.
Description	This API enables, disables or clears the wake-up events of the CAN transceiver according to parameter TrcvWakeupMode. - If parameter TrcvWakeupMode is CANTRCV_WUMODE_ENABLE: wake up event is informed to EcuM. - If parameter TrcvWakeupMode is CANTRCV_WUMODE_DISABLE: wake up event is not informed to EcuM and it is stored. - If parameter TrcvWakeupMode is CANTRCV_WUMODE_CLEAR: stored pending wake up will be cleared.	
Source	AUTOSAR	
Error handling	CANTRCV_17_V9251_E_INVALID_TRANSCEIVER, CANTRCV_17_V9251_E_UNINIT, CANTRCV_17_V9251_E_PARAM_TRCV_WAKEUP_MODE	
Configuration dependencies	-	
User hints	None	
SFR accessed	-	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.7 CanTrcv_17_V9251_CheckWakeup

Table 61 **Specification for `CanTrcv_17_V9251_CheckWakeup` API**

Syntax	`Std_ReturnType CanTrcv_17_V9251_CheckWakeup`
	`(`
	`    const uint8 Transceiver`
	`)`

Service ID	0x07
Sync/Async	Synchronous
ASIL Level	QM

Table 61 **Specification for `CanTrcv_17_V9251_CheckWakeup` API (continued)**

Re-entrancy	Reentrant	
Parameters (in)	Transceiver	CAN transceiver to which API call has to be made. Note: CanTrcv_17_V9251 driver supports 12 channels, so the range of this parameter must be 0 to 11.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : when a valid wake up interrupt is detected E_NOT_OK: when false wake up interrupt is detected or due to DET errors.
Description	This API service is called by the underlying CANIF module in case a wake up interrupt is detected. This API validates the wake up by checking the current mode of CAN Transceiver and CanTrcvWakeupByBusUsed configuration parameter. Once it gets valid wake up interrupt, the mode change of the transceiver from standby to normal is taken care in this API. *Note: Since it is not possible to trace CAN transceiver wake up indication on RxD pin, the wake up validation from hardware is not done.*	
Source	AUTOSAR	
Error handling	CANTRCV_17_V9251_E_UNINIT, CANTRCV_17_V9251_E_INVALID_TRANSCEIVER	
Configuration dependencies	-	
User hints	None	
SFR accessed	-	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.4 Notifications and Callbacks

The CanTrcv_17_V9251 driver does not provide any notification or callback.

1.3.5 Scheduled functions

The CanTrcv_17_V9251 driver does not provide any scheduled functions. *Note: Since the CAN transceiver TLE9251V does not support polling mode, therefore no scheduled functions are available.*

1.3.6 Interrupt service routines

The CanTrcv_17_V9251 driver does not provide any interrupt handlers. *Note: The CAN transceiver TLE9251V wake up interrupts are handled by the ICU driver.*

1.3.7 Callout

The CanTrcv_17_V9251 driver does not provide any callout.

1.3.8 Errors Handling

Error Name: Description	Source	Error ID (AS422)	Type (AS422)	Error ID (AS440)	Type (AS440)
CANTRCV_17_V9251_E_INIT_FAILED: This error is reported when CanTrcv_17_V9251_Init API is called without NULL_PTR as the parameter.	AUTOSAR	0x27	DET	0x27	DET
CANTRCV_17_V9251_E_INVALID_TRANSCEIVER: This error will occur when API is called with wrong transceiver parameter for the CanTrcv_17_V9251 driver.	AUTOSAR	0x01	DET	0x01	DET
CANTRCV_17_V9251_E_PARAM_POINTER: This error will occur when API is called with invalid pointer parameter for the CanTrcv_17_V9251 driver.	AUTOSAR	0x02	DET	0x02	DET
CANTRCV_17_V9251_E_PARAM_TRCV_OPMODE: This error will occur when API service is called with invalid parameter for OpMode.	AUTOSAR	0x24	DET	0x24	DET
CANTRCV_17_V9251_E_PARAM_TRCV_WAKEUP_MODE: This error will occur when API service is called with invalid parameter for TrcvWakeupMode.	AUTOSAR	0x23	DET	0x23	DET
CANTRCV_17_V9251_E_UNINIT: This error will occur when module API service is called without CanTrcv_17_V9251 module initialization.	AUTOSAR	0x11	DET	0x11	DET

1.3.9 Deviations and limitations

This section describes the deviations and limitations of the CanTrcv_17_V9251 driver.

1.3.9.1 Deviations

This section describes the deviations of the CanTrcv_17_V9251 driver.

1.3.9.1.1 Software specification deviations

This section describes the deviations from software specification.

Table 62 Known deviations

Reference	Deviation
AUTOSAR CAN Transceiver requirement[SWS_CanTrcv_00090]	Since the TLE9251V hardware supports the wake up functionality, NOT_SUPPORTED mode is not available from the CanTrcv_17_V9251 driver.
AUTOSAR CAN Transceiver requirement[SWS_CanTrcv_00091]	Wake-up by polling mode is not supported by the CanTrcv_17_V9251 driver due to hardware limitations. Instead wake-up is supported by the interrupt mode.
AUTOSAR CAN Transceiver requirement[SWS_CanTrcv_00171],SWS_CanTrcv_00172],SWS_CanTrcv_00173]	Since the ICU driver does not depend on Icu_EnableNotification and Icu_DisableNotification for reporting a wake up, these interfaces are not used in the CanTrcv_17_V9251 driver.
AUTOSAR CAN Transceiver requirement[SWS_CanTrcv_00067]	AUTOSAR-specified file structure is modified to avoid the compilation errors and repeated file inclusions.
AUTOSAR CAN Transceiver requirement[SWS_CanTrcv_00148]	CANTRCV_TRCVMODE_SLEEP mode from AUTOSAR SWS is not supported due to hardware limitations.
AUTOSAR CAN Transceiver requirement[SWS_CanTrcv_00228]	The DEM error CANTRCV_E_BUS_ERROR is not supported due to hardware limitations.
AUTOSAR CAN Transceiver requirement[SWS_CanTrcv_00174],[SWS_CanTrcv_00175],[SWS_CanTrcv_00177],[SWS_CanTrcv_00178]	Since the CAN transceiver hardware does not support partial networking, all these requirements are not supported by the driver.
AUTOSAR CAN Transceiver requirement[SWS_CanTrcv_00084].	The datatypes related for DEM are availed via Dem.h instead of Rte_Dem_Types.h. *Note: Applicable for Autosar version 4.4.0 only.*

1.3.9.1.2 AMDC Violations

This section describes the violations reported by the Vector AMDC checker tool with respect to AUTOSAR.

Table 63 Violations reported by AMDC checker tool for A207

AMDC Rule	A207
Description	TLE9251V hardware supports HS CAN Standard data rates up to 1MBit/s and CAN FD data rates up to 5 Mbit/s. Hence, the maximum value of CanTrcvMaxBaudrate is set to 5000. The unit is kbps.

1.3.9.1.3 VSMD Violations

This section describes the violations reported by the EB VSMD checker tool with respect to AUTOSAR.

Table 64 *Violations reported by VSMD checker tool for EB03*

Rule ID:	EB03
VSMD Node(s):	/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess/CanTrcvSpiSequence/CanTrcvSpiAccessSynchronous /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvChannelEcucPartitionRef /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvDemEventParameterRefs /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvDemEventParameterRefs/CANTRCV_E_BUS_ERROR /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvIcuChannelRef /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPorWakeupSourceRef /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvSyserrWakeupSourceRef /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvWakeupByBusUsed /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvWakeupSourceRef /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvMainFunctionDiagnosticsPeriod /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvMainFunctionPeriod /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvTimerType /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvWaitTime
Description:	The StMD node has LOWER-MULTIPLICITY=0 and UPPER-MULTIPLICITY=1. The VSMD-node shall get the OPTIONAL-attribute instead of creating a list!
Additional Information:	

Table 65 ***Violations reported by VSMD checker tool for EB09***

Rule ID:	EB09
VSMD Node(s):	/AURIX2G_V9251/EcucDefs/CanTrcv
Description:	EB specific rule to check consistency of parameter postBuildVariantUsed.
Additional Information:	

Table 66 ***Violations reported by VSMD checker tool for EcucSws_1007***

Rule ID:	EcucSws_1007
VSMD Node(s):	/AURIX2G_V9251/EcucDefs/CanTrcv/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess/CanTrcvDioChannelAccess/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess/CanTrcvDioChannelAccess/CanTrcvDioSymNameRef/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess/CanTrcvDioChannelAccess/CanTrcvHardwareInterfaceName/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess/CanTrcvSpiSequence/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess/CanTrcvSpiSequence/CanTrcvSpiAccessSynchronous/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess/CanTrcvSpiSequence/CanTrcvSpiSequenceName/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvChannelEcucPartitionRef/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvChannelId/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvChannelUsed/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvControlsPowerSupply/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvDemEventParameterRefs/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/

Table 66 *Violations reported by VSMD checker tool for EcucSws_1007* **(continued)**

	CanTrcvChannel/CanTrcvDemEventParameterRefs/CANTRCV_E_BUS_ERROR/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvHwPnSupport/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvIcuChannelRef/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvInitState/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvMaxBaudrate/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvBaudRate/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvBusErrFlag/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnCanIdIsExtended/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnEnabled/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameCanId/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameCanIdMask/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameDataMaskSpec/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameDataMaskSpec/CanTrcvPnFrameDataMask/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameDataMaskSpec/CanTrcvPnFrameDataMaskIndex/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameDlc/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPowerOnFlag/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPorWakeupSourceRef/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvSyserrWakeupSourceRef/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvWakeupByBusUsed/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/

Table 66 ***Violations reported by VSMD checker tool for EcucSws_1007*** **(continued)**

	CanTrcvChannel/CanTrcvWakeupSourceRef/ AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/ CanTrcvSPICommRetries/AURIX2G_V9251/EcucDefs/ CanTrcv/CanTrcvConfigSet/ CanTrcvSPICommTimeout/AURIX2G_V9251/EcucDefs/ CanTrcv/CanTrcvGeneral/AURIX2G_V9251/EcucDefs/ CanTrcv/CanTrcvGeneral/CanTrcvDevErrorDetect/ AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/ CanTrcvEcucPartitionRef/AURIX2G_V9251/EcucDefs/ CanTrcv/CanTrcvGeneral/CanTrcvGetVersionInfo/ AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/ CanTrcvIndex/AURIX2G_V9251/EcucDefs/CanTrcv/ CanTrcvGeneral/ CanTrcvMainFunctionDiagnosticsPeriod/ AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/ CanTrcvMainFunctionPeriod/AURIX2G_V9251/ EcucDefs/CanTrcv/CanTrcvGeneral/ CanTrcvTimerType/AURIX2G_V9251/EcucDefs/ CanTrcv/CanTrcvGeneral/CanTrcvVersionInfoApi/ AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/ CanTrcvWaitTime/AURIX2G_V9251/EcucDefs/CanTrcv/ CanTrcvGeneral/CanTrcvWakeUpSupport
Description:	For Integer and Float Parameters the MIN values must be >= and the MAX values <= as in the StMD.
Additional Information:	

Table 67 ***Violations reported by VSMD checker tool for EcucSws_1014***

Rule ID:	EcucSws_1014
VSMD Node(s):	/AURIX2G_V9251/EcucDefs/CanTrcv/ CanTrcvConfigSet/CanTrcvChannel
Description:	Additional vendor specific parameter definitions (using ParameterTypes), container definitions and references shall be added to the VSMD according to the alphabetical order.
Additional Information:	

Table 68 ***Violations reported by VSMD checker tool for EcucSws_1035***

Rule ID:	EcucSws_1035
VSMD Node(s):	/AURIX2G_V9251/EcucDefs/CanTrcv /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet /AURIX2G_V9251/EcucDefs/CanTrcv/ CanTrcvConfigSet/CanTrcvChannel /AURIX2G_V9251/EcucDefs/CanTrcv/ CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess

Table 68 *Violations reported by VSMD checker tool for EcucSws_1035* **(continued)**

	/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess/CanTrcvDioChannelAccess /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess/CanTrcvDioChannelAccess/CanTrcvDioSymNameRef /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess/CanTrcvDioChannelAccess/CanTrcvHardwareInterfaceName /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess/CanTrcvSpiSequence /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess/CanTrcvSpiSequence/CanTrcvSpiAccessSynchronous /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess/CanTrcvSpiSequence/CanTrcvSpiSequenceName /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvChannelEcucPartitionRef /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvChannelId /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvChannelUsed /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvControlsPowerSupply /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvDemEventParameterRefs /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvDemEventParameterRefs/CANTRCV_E_BUS_ERROR

Table 68 *Violations reported by VSMD checker tool for EcucSws_1035* **(continued)**

	/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvHwPnSupport /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvIcuChannelRef /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvInitState /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvMaxBaudrate /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvBaudRate /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvBusErrFlag /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnCanIdIsExtended /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnEnabled /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameCanId /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameCanIdMask /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameDataMaskSpec /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameDataMaskSpec/CanTrcvPnFrameDataMask /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameDataMaskSpec/CanTrcvPnFrameDataMaskIndex

Table 68 ***Violations reported by VSMD checker tool for EcucSws_1035*** **(continued)**

	/AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameDlc /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPowerOnFlag /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPorWakeupSourceRef /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvSyserrWakeupSourceRef /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvWakeupByBusUsed /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvWakeupSourceRef /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvSPICommRetries /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvSPICommTimeout /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvDevErrorDetect /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvEcucPartitionRef /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvGetVersionInfo /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvIndex /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvMainFunctionDiagnosticsPeriod /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvMainFunctionPeriod /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvTimerType /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvVersionInfoApi /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvWaitTime /AURIX2G_V9251/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvWakeUpSupport
Description:	For Containers, Parameters and References elements UUID must be unique (also between StMD and VSMD).

Table 68 ***Violations reported by VSMD checker tool for EcucSws_1035*** **(continued)**

Additional Information:	

Table 69 ***Violations reported by VSMD checker tool for EcucSws_2101***

Rule ID:	EcucSws_2101
VSMD Node(s):	/AURIX2G_V9251/EcucDefs/CanTrcv/ POST_BUILD_VARIANT_USED
Description:	For each ConfigurationVariant supported by the ModuleDef, there must be one ImplementationConfigClass element. In VSMD, the ImplementationConfigClass is mandatory.
Additional Information:	

Table 70 ***Violations reported by VSMD checker tool for EcucSws_6003***

Rule ID:	EcucSws_6003
VSMD Node(s):	The SHORT-NAME of the AR-PACKAGEs of StMD and VSMD must be different to ensure a unique SHORT-NAME-path.
Description:	The SHORT-NAME of the AR-PACKAGEs of StMD and VSMD must be different to ensure a unique SHORT-NAME-path.
Additional Information:	

Table 71 ***Violations reported by VSMD checker tool for TpsEcuc_06051_ASR41***

Rule ID:	TpsEcuc_06051_ASR41
VSMD Node(s):	/AURIX2G_V9251/EcucDefs/CanTrcv/ POST_BUILD_VARIANT_USED
Description:	The implementationConfigClass of an EcucParameterDef or EcucAbstractReferenceDef in VSMD shall be the same or higher (where PreCompile configuration class is considered to be the lowest and PostBuild the highest) as in StMD with respect to the selected subset defined by the actually implemented supportedConfigVariant.
Additional Information:	

1.3.9.2 Limitations

The CanTrcv_17_V9251 driver does not have any limitations.

Revision history

Table 72 Major changes since last version

Date	Version	Description
2021-03-08	3.0	Document is released.
2021-02-25	2.1	SWS ID corrected for Rte_Dem_Types.h in Software specification deviations.
2020-11-20	2.0	Document is released.
2020-11-12	1.1	• Unwanted text removed in Error handling section. • SFR access fields added for APIs (Since CanTrcv driver is external driver SFR accessed is not applicable).
2020-08-13	1.0	Document is released.
2020-08-06	0.1	• Initial version • CanTrcv_17_W9255 chapter moved from MCISAR_TC3xx_UM_Basic to this document

Trademarks

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

Edition 2021-03-08

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-ocr1484806431059

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.

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.