MCAL User Manual for CanTrcv_17_W9255

32-bit TriCore™ AURIX™ 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™ AURIX™ 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_W9255 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™ 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

Table of contents

Table of contents

Table of contents

Table of contents

1 CanTrcv_17_W9255 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 the microcontroller. The CAN transceiver driver supports the Infineon TLE9255W hardware. The CAN transceiver driver provides the services for:

- Driver initialization
- Switching of operation modes
- Standard bus wake-up functionality
- CAN partial networking with selective wake-up functionality

The communication between the microcontroller and the CAN transceiver is implemented through the Serial Peripheral Interface (SPI). This communication is synchronous and is configured as full duplex. Multiple CAN transceivers can be connected to the same SPI kernel.

1.1.2 Hardware-software mapping

This section describes the system view of the CanTrcv_17_W9255 driver and peripherals administered by it.

1 CanTrcv_17_W9255 driver

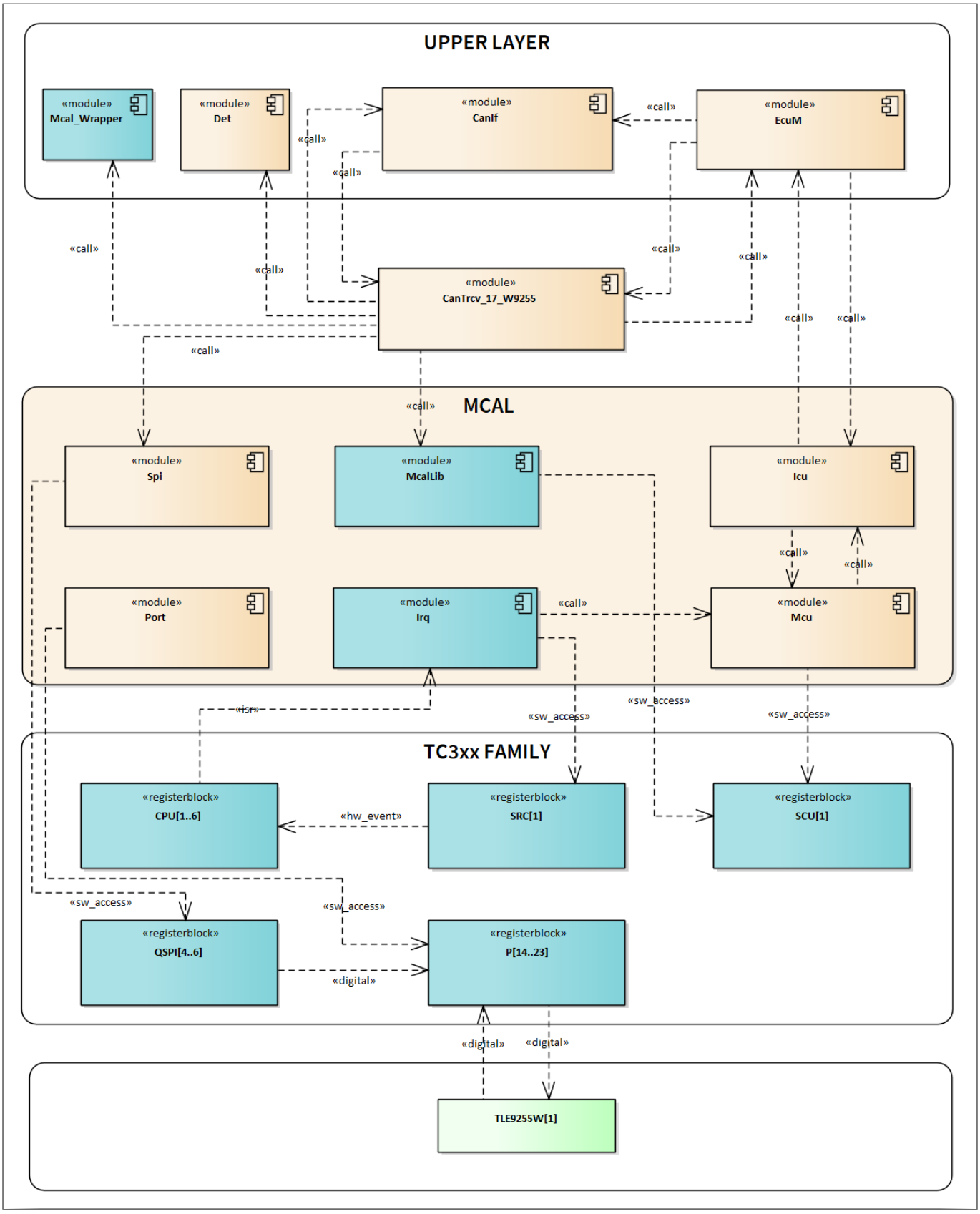

Figure 1 Mapping of hardware-software interfaces

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 CAN transceiver driver depends on PORT driver for configuring the RxD, TxD, MOSI, MISO, CSN, SCLK and WAKE 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

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 TLE9255W: primary hardware peripheral

Hardware functional features

The CAN transceiver driver uses the TLE9255W 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
- CAN Flexible data rate (CAN FD) transmission up to 5 MBit/s
- Supports selective wakeup functionality where the transceiver is woken up by selective wake frames called as wake-up frames (WUFs) when the transceiver is in low power modes.
- Wake-up pattern (WUP) detection in all low-power modes
- Local wake-up input
- Wake-up source recognition

The unsupported features of the TLE9255W are:

- Receive-only mode

Users of the hardware

The CAN transceiver driver exclusively utilizes the TLE9255W module.

Hardware diagnostic features

The hardware diagnostic features used by the driver are:

- The error status register records if any SPI failure is detected or if an invalid SPI command is passed. Both these error scenarios are signaled on the MISO pin. The SPI indicates failures, error counter overflow and synchronization failures to the microcontroller. An invalid SPI command is ignored and the CMD_ERR bit is set and signaled on the MISO pin. Only the microcontroller can reset the CMD_ERR bit. On SPI failure, SPI commands are ignored.
- The SysErr flag in the selective wake status register indicates an error condition in the selective wake unit of the TLE9255W.
- Error counter status register tracks error counter overflow that can occur upon receiving invalid CAN frames.

The unsupported diagnostic features of the TLE9255W are:

- Short-circuit protection
- Undervoltage detection
- Overtemperature warning
- TxD timeout function
- CSN Timeout

Hardware events

The CAN transceiver driver uses the following hardware events from the TLE9255W IP:

Wake-up event: Indication of a valid wake-up event is signaled on the RxD pin and this triggers a mode change.

1.1.3 File structure

1.1.3.1 C file structure

This section provides details of the C files of the CanTrcv_17_W9255 driver.

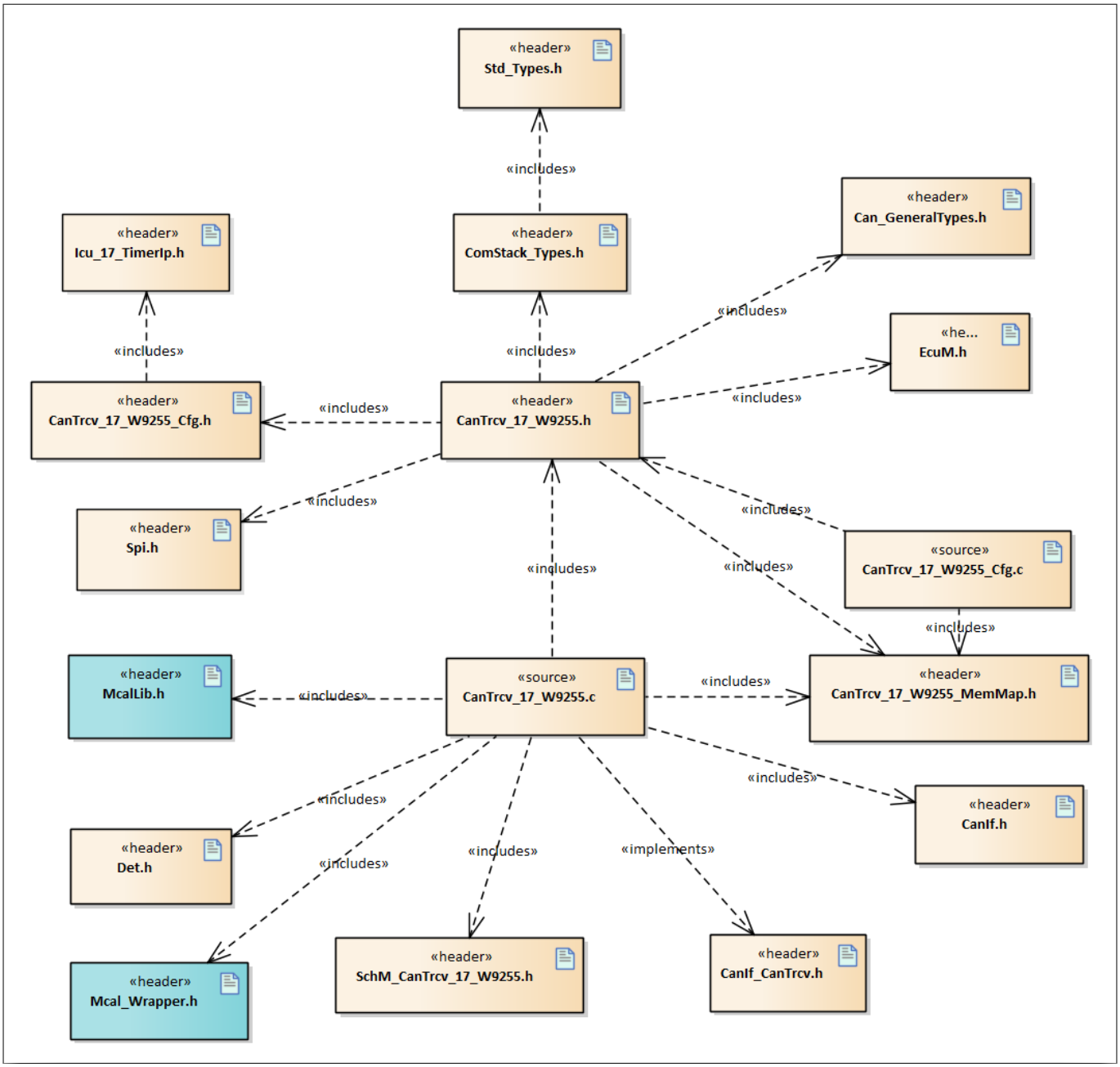

Figure 2 CanTrcv_17_W9255_C_File_Structure-1.png

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_W9255.c`	File (Static) containing implementation of APIs
`CanTrcv_17_W9255.h`	Header file (Static) defining prototypes of data structures and APIs
`CanTrcv_17_W9255_Cfg.c`	File (Generated) containing definition of the configuration data structures

(table continues...)

Table 2 (continued) C file structure

File name	Description
CanTrcv_17_W9255_Cfg.h	Header file (Generated) containing constants and pre-processor macros as #defines
CanTrcv_17_W9255_MemMap.h	File (Static) containing the memory section definitions used by the CAN transceiver 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
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.
Mcal_Wrapper.h	Provides the exported interfaces for Production Error and Runtime Development Errors. Implemented by default to include functions of Dem.h and Det.h files. This file can be modified by the user but function prototype is not user modifiable.
SchM_CanTrcv_17_W9255.h	Export header for SchM functions of the CAN transceiver driver
Spi.h	Header file (Static) defining prototypes of data structures and APIs
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_W9255 driver.

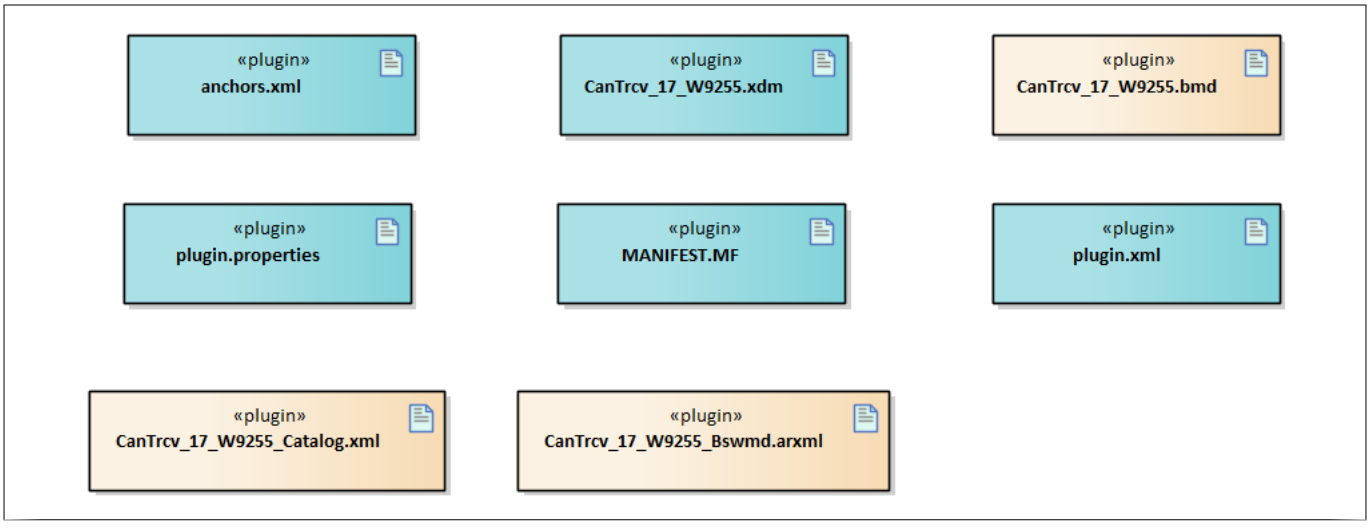

Figure 3 CanTrcv_17_W9255_Code_Generator_Plugin_Files-1.png

Table 3 Code generator plugin files

File name	Description
CanTrcv_17_W9255.bmd	AUTOSAR format XML data model schema file
CanTrcv_17_W9255.xdm	Tresos format XML data model schema file
CanTrcv_17_W9255_Bswmd.arxml	AUTOSAR format module description file
CanTrcv_17_W9255_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 CAN transceiver driver
anchors.xml	Tresos anchors support file for the CAN transceiver Driver
plugin.properties	Tresos plugin support file for the CAN transceiver driver
plugin.xml	Tresos plugin support file for the CAN Transceiver driver

1.1.4 Integration hints

This section lists the key points that an integrator or user of the CanTrcv_17_W9255 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 CAN transceiver 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. Refer to the Notifications and call-backs section for the notification functions called by the transceiver to EcuM.

- **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. Refer to the Notifications and call-backs section for the notification functions called by the transceiver to CanIf.

- **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_W9255_MemMap.h file. The CanTrcv_17_W9255_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_W9255_START_SEC_VAR_CLEARED_QM_LOCAL_32
/* User Pragma here */
#undef CANTRCV_17_W9255_START_SEC_VAR_CLEARED_QM_LOCAL_32
#undef MEMMAP_ERROR
#elif defined CANTRCV_17_W9255_STOP_SEC_VAR_CLEARED_QM_LOCAL_32
/* User Pragma here */
#undef CANTRCV_17_W9255_STOP_SEC_VAR_CLEARED_QM_LOCAL_32
#undef MEMMAP_ERROR
#elif defined CANTRCV_17_W9255_START_SEC_VAR_CLEARED_QM_LOCAL_16
/* User Pragma here */
#undef CANTRCV_17_W9255_START_SEC_VAR_CLEARED_QM_LOCAL_16
#undef MEMMAP_ERROR
#elif defined CANTRCV_17_W9255_STOP_SEC_VAR_CLEARED_QM_LOCAL_16
/* User Pragma here */
#undef CANTRCV_17_W9255_STOP_SEC_VAR_CLEARED_QM_LOCAL_16
#undef MEMMAP_ERROR
#elif defined CANTRCV_17_W9255_START_SEC_VAR_CLEARED_QM_LOCAL_8
/* User Pragma here */
#undef CANTRCV_17_W9255_START_SEC_VAR_CLEARED_QM_LOCAL_8
#undef MEMMAP_ERROR
#elif defined CANTRCV_17_W9255_STOP_SEC_VAR_CLEARED_QM_LOCAL_8
/* User Pragma here */
#undef CANTRCV_17_W9255_STOP_SEC_VAR_CLEARED_QM_LOCAL_8
#undef MEMMAP_ERROR
#elif defined CANTRCV_17_W9255_START_SEC_VAR_INIT_QM_LOCAL_8
/* User Pragma here */
#undef CANTRCV_17_W9255_START_SEC_VAR_INIT_QM_LOCAL_8
#undef MEMMAP_ERROR
#elif defined CANTRCV_17_W9255_STOP_SEC_VAR_INIT_QM_LOCAL_8
/* User Pragma here */
#undef CANTRCV_17_W9255_STOP_SEC_VAR_INIT_QM_LOCAL_8
#undef MEMMAP_ERROR
#elif defined CANTRCV_17_W9255_START_SEC_VAR_INIT_QM_LOCAL_32
/* User Pragma here */
#undef CANTRCV_17_W9255_START_SEC_VAR_INIT_QM_LOCAL_32
#undef MEMMAP_ERROR
#elif defined CANTRCV_17_W9255_STOP_SEC_VAR_INIT_QM_LOCAL_32
/* User Pragma here */
#undef CANTRCV_17_W9255_STOP_SEC_VAR_INIT_QM_LOCAL_32
#undef MEMMAP_ERROR

/**** CANTRCV_17_W9255 MODULE CONFIG DATA ****/

#elif defined CANTRCV_17_W9255_START_SEC_CONFIG_DATA_QM_LOCAL_UNSPECIFIED
/* User Pragma here */
#undef CANTRCV_17_W9255_START_SEC_CONFIG_DATA_QM_LOCAL_UNSPECIFIED
#undef MEMMAP_ERROR
#elif defined CANTRCV_17_W9255_STOP_SEC_CONFIG_DATA_QM_LOCAL_UNSPECIFIED
```

```
/* User Pragma here */
#undef CANTRCV_17_W9255_STOP_SEC_CONFIG_DATA_QM_LOCAL_UNSPECIFIED
#undef MEMMAP_ERROR
#elif defined CANTRCV_17_W9255_START_SEC_CONFIG_DATA_QM_LOCAL_8
/* User Pragma here */
#undef CANTRCV_17_W9255_START_SEC_CONFIG_DATA_QM_LOCAL_8
#undef MEMMAP_ERROR
#elif defined CANTRCV_17_W9255_STOP_SEC_CONFIG_DATA_QM_LOCAL_8
/* User Pragma here */
#undef CANTRCV_17_W9255_STOP_SEC_CONFIG_DATA_QM_LOCAL_8
#undef MEMMAP_ERROR
#elif defined CANTRCV_17_W9255_START_SEC_CONFIG_DATA_QM_LOCAL_16
/* User Pragma here */
#undef CANTRCV_17_W9255_START_SEC_CONFIG_DATA_QM_LOCAL_16
#undef MEMMAP_ERROR
#elif defined CANTRCV_17_W9255_STOP_SEC_CONFIG_DATA_QM_LOCAL_16
/* User Pragma here */
#undef CANTRCV_17_W9255_STOP_SEC_CONFIG_DATA_QM_LOCAL_16
#undef MEMMAP_ERROR

/**** CANTRCV_17_W9255 MODULE CODE SECTION ****/

#elif defined CANTRCV_17_W9255_START_SEC_CODE_QM_LOCAL
/* User Pragma here */
#undef CANTRCV_17_W9255_START_SEC_CODE_QM_LOCAL
#undef MEMMAP_ERROR
#elif defined CANTRCV_17_W9255_STOP_SEC_CODE_QM_LOCAL
/* User Pragma here */
#undef CANTRCV_17_W9255_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 errors reported by the BSW modules. The CAN transceiver driver reports all the development errors to the DET module through the `Det_ReportError()` API. The user of the CAN transceiver 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.

- **Mcal_Wrapper**

  This Driver performs reporting of the Production and Runtime errors. The Handling of the reported errors shall be done by the user. The Mcal_Wrapper_Det_ReportRuntimeError() API, Mcal_Wrapper_Dem_SetEventStatus() API and Mcal_Wrapper_Dem_ReportErrorStatus() API are provided in the Mcal_Wrapper.c and Mcal_Wrapper.h files as a stub code, and can be updated by the integrator to handle the reported errors. The files Mcal_Wrapper.c and Mcal_Wrapper.h are user modifiable, Where the function prototype is not user modifiable and by default the Mcal Wrapper function shall calls AUTOSAR DEM and DET Modules.

  The user of the CAN transceiver driver shall process Runtime errors reported to the Mcal_Wrapper module. Production errors are not applicable for Can Transceiver. The interface used for reporting Runtime error in AUTOSAR version 4.4.0 is Mcal_Wrapper_Det_ReportRuntimeError() API. The Mcal_Wrapper.c and

Mcal_Wrapper.h files are provided in the MCAL package as a stub code and can be replaced with a user specific Runtime error handling module/s during the integration phase.

- **SchM:**

  The SchM module is a part of the RTE that manages the BSW Scheduler. The CAN transceiver driver uses the exclusive areas defined in the `SchM_CanTrcv_17_W9255.c` file to protect the SFRs and variables from concurrent accesses from different threads. The SchM identified for the CanTrcv_17_W9255 driver is:

  SpiStatusUpdate

  The `SchM_CanTrcv_17_W9255.h` and `SchM_CanTrcv_17_W9255.c` files are provided in the MCAL package as an example code and needs to be updated by the integrator. The user must implement the SchM functions defined by the CanTrcv_17_W9255 driver as **suspend / resume** of interrupts for the CPU on which the API is invoked. A sample implementation of the SchM function is shown as follows:

```
/**** Sample implementation of SchM_CanTrcv_17_W9255.c ****/

void SchM_Enter_CanTrcv_17_W9255_SpiStatusUpdate (void)
{
 SuspendAllInterrupts(); /* Suspend CPU core interrupt */
}
void SchM_Exit_CanTrcv_17_W9255_SpiStatusUpdate (void)
{
 ResumeAllInterrupts(); /* Resume CPU core interrupt */
}
```

- **Safety error**

  The CAN transceiver driver does not report any safety errors.

- **Notifications and callbacks**

  The CAN transceiver driver does not implement any notifications. However, the driver reports mode change confirmation, partial networking availability, confirmation of wake-up flags check and clearing of WUF flag indication through notification functions of the CanIf module. The driver also reports wake-up detection through notification functions of the EcuM module.

  The driver reports the following notification functions.

  `EcuM_SetWakeupEvent(EcuM_WakeupSource)`: notification that a wake-up event is detected

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

  `CanIf_CheckTrcvWakeFlagIndication()`: notification for successful check of wake-up flags that was triggered for a transceiver

  `CanIf_ClearTrcvWufFlagIndication()`: notification that the WUF flag is cleared successfully for the triggered transceiver

  `CanIf_ConfirmPnAvailability()`: notification that indicates the triggered transceiver is running in the PN communication mode

- **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 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_W9255 driver does not support execution on multiple cores simultaneously.

1.1.4.3 MCU support

The CanTrcv_17_W9255 driver is dependent on the MCU driver for the ERU channel allocation and system clock configuration. The initialization of the CanTrcv_17_W9255 driver must be started only after completing 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 CAN transceiver driver through the PORT configuration and initialize the port pins prior to invoking of CAN transceiver driver initialization. The MISO, MOSI, SCLK, CSN and WAKE pins of CAN transceiver TLE9255W must be configured in the PORT driver configuration.

1.1.4.5 DMA support

The CanTrcv_17_W9255 driver does not use any services provided by the DMA driver.

1.1.4.6 Interrupt connections

The CanTrcv_17_W9255 driver does not provide any interrupt handlers.

1.1.4.7 Example usage

This chapter describes how the CAN transceiver 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 (Development Error Tracer) should be enabled. The behaviour of the APIs is undefined if DET is disabled and wrong parameters are passed.

Configuration of the driver

1. In the MCU driver, configure the system clock, input clock source for the QSPI peripheral and the QSPI peripheral frequency.
2. In the PORT driver, configure port pins referred by the CAN transceiver TLE9255W. For each configured transceiver channel, MISO, MOSI, SCLK, CSN and WAKE pins have to be configured.
3. In the SPI driver, configure the required number of sequences according to the number of channels (external devices) configured in the CAN transceiver. Each transceiver channel must be configured to have one independent sequence with a job and a channel exclusively configured for a transceiver channel.
4. The MCALLIB driver configuration is required for timing services used by the CAN transceiver driver.
5. In the EcuM, configure the wake-up source reference, wake-up source reference for POR and SYSERR.
6. In the CanTrcv_17_W9255 driver, configure the required number of channels with Normal, Standby or Sleep modes. The `CanTrcvWakeupByBusUsed` parameter must be enabled for wake-up support for the corresponding channel.
7. In the CanTrcv_17_W9255 driver, the `CanTrcvSpiSequenceName` parameter must be referenced to SPI channel to access the TLE9255W hardware.

In the wake-up by interrupt mode, the following additional configurations are required.

1. In the ICU driver, configure the ICU wake-up capable channel to detect the FALLING EDGE of the CAN transceiver TLE9255W RXD pin, this needs the ERU channel configuration.
2. In the MCU driver, allocate the ERU channels for the ICU driver.
3. The IRQ driver configuration is required to configure the interrupt priorities for the interrupts used by the ICU.
4. In the EcuM, configure the wake-up source and same wake-up source must be configured in the CanTrcv_17_W9255 and the ICU configuration.

Refer to the following sample configurations of SPI channel and SPI external device.

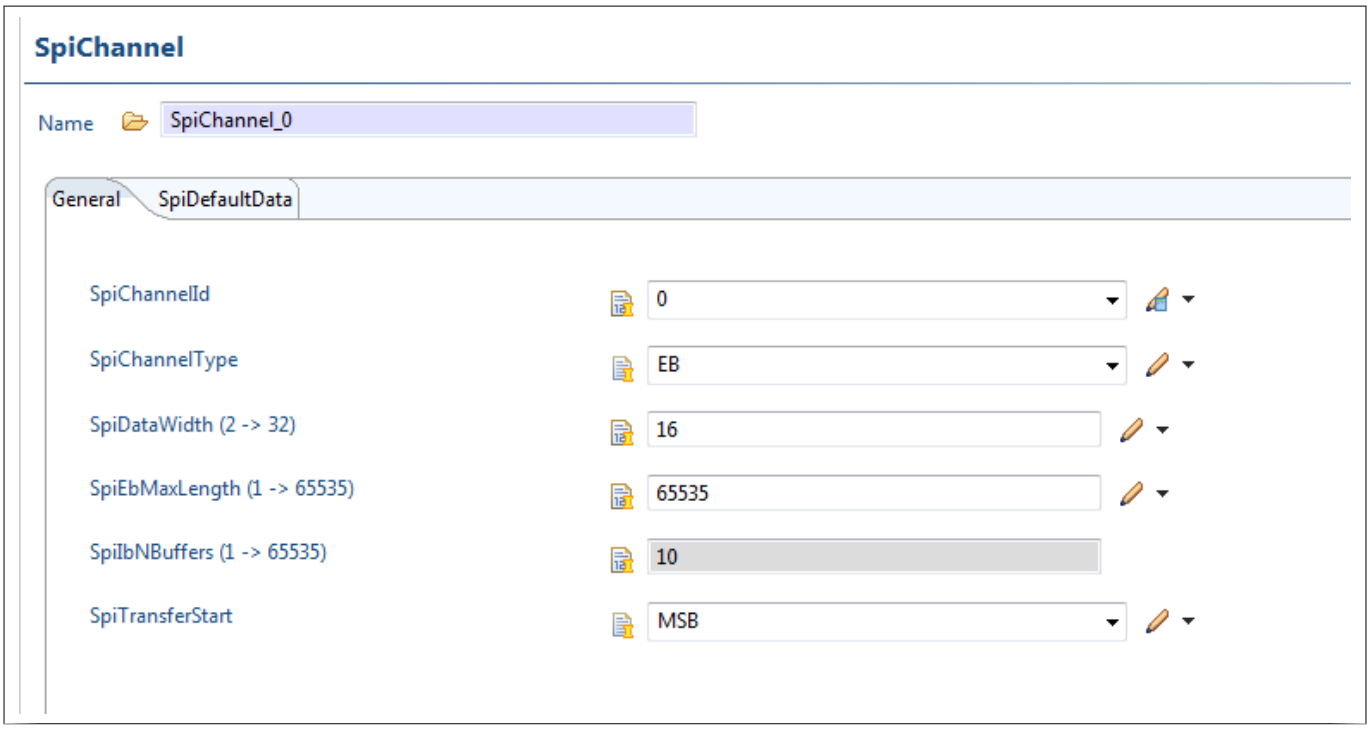

Figure 4 SPI channel configuration

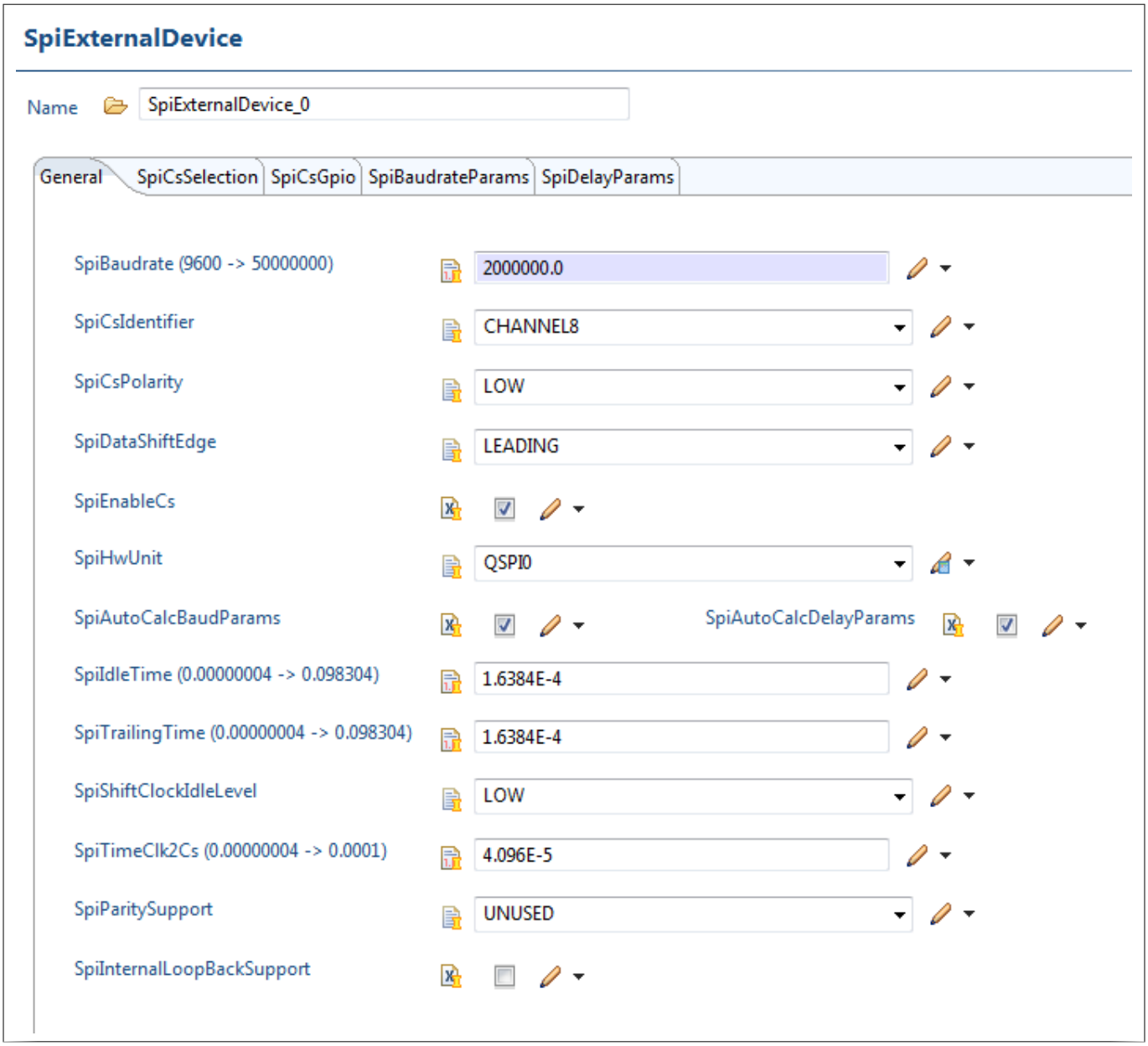

Figure 5 SPI external device configuration

Wake-up by interrupt mode:

The CanTrcv_17_W9255 driver is dependent on the ICU driver for edge detection. The initialization of the CanTrcv_17_W9255 driver must be started only after completion of the ICU initialization. The ICU must be put to sleep mode, and wake-up for the corresponding channel has to be enabled to support the wake-up functionality.

Initialization sequence of CanTrcv_17_W9255 driver:

The Initialization sequence of the CanTrcv_17_W9255 driver is as follows.

```
/*MCU Initialization */
Mcu_Init(&Mcu_Config);
Mcu_InitClock(0U);
while(Mcu_GetPllStatus() != MCU_PLL_LOCKED);
Mcu_DistributePllClock ();
/* Port Initialization */
Port_Init(&Port_Config);
/* SPI Initialization */
Spi_Init(&Spi_Config);
/*ICU Initialization */
Icu_17_TimerIp_Init(&Icu_17_TimerIp_Config);
/* CanTrcv_17_W9255 Initialization */
CanTrcv_17_W9255_Init(NULL_PTR);
/* Further APIs of CanTrcv_17_W9255 driver can be called now */
```

CAN Transceiver operation mode change:

After the initialization of the CanTrcv_17_W9255 driver, the following sequence can be followed for the mode change operation.

```
/* CanTrcv_17_W9255 mode change operation */
CanTrcv_17_W9255_SetOpMode(0,CANTRCV_TRCVMODE_NORMAL);
```

CAN Transceiver wakeup mode change:

After the initialization of the CanTrcv_17_W9255 driver, the following sequence can be followed for changing the wake-up mode.

```
/* CanTrcv_17_W9255 wake-up mode change */
CanTrcv_17_W9255_SetWakeupMode(0,CANTRCV_WUMODE_ENABLE);
```

1.1.5 Key architectural considerations

1.1.5.1 Wake-up by interrupt mode

In addition to the wake-up support by the polling mode, the CAN transceiver driver supports the detection of wake-up by the interrupt mode. This can be configured using the `CanTrcvWakeUpSupport` configuration parameter. In this mode, the RxD pin of the CAN Transceiver hardware is connected to the ERU. Indication of a valid wake-up event is signalled on the RxD pin by the CAN Transceiver. The ICU driver monitors the RxD pin transitions and notifies the EcuM after wake-up detection.

1.1.5.2 User mode support

Since the CAN transceiver driver does not access any AURIX SFRs, the driver does not support the user mode configuration for any of its APIs. Therefore, all APIs of the driver can be executed in the User1 or Supervisor Mode. [cover parentID CANTRCVW9255 = {D0984ABF-D8D3-49bf-9AF1-22CD9DF62F4B}]

1.1.5.3 CanTrcv_17_W9255_SetOpMode and CanTrcv_17_W9255_CheckWakeFlag APIs implemented as synchronous

Since AUTOSAR recommends that the used APIs of the underlying driver (SPI) should be synchronous, therefore synchronous implementation is used for these APIs.

1.2 Assumptions of Use (AoU)

There are no AoUs for the CanTrcv_17_W9255 driver.

1.3 Reference information

1.3.1 Configuration interfaces

Supported configuration variant: Pre-Compile

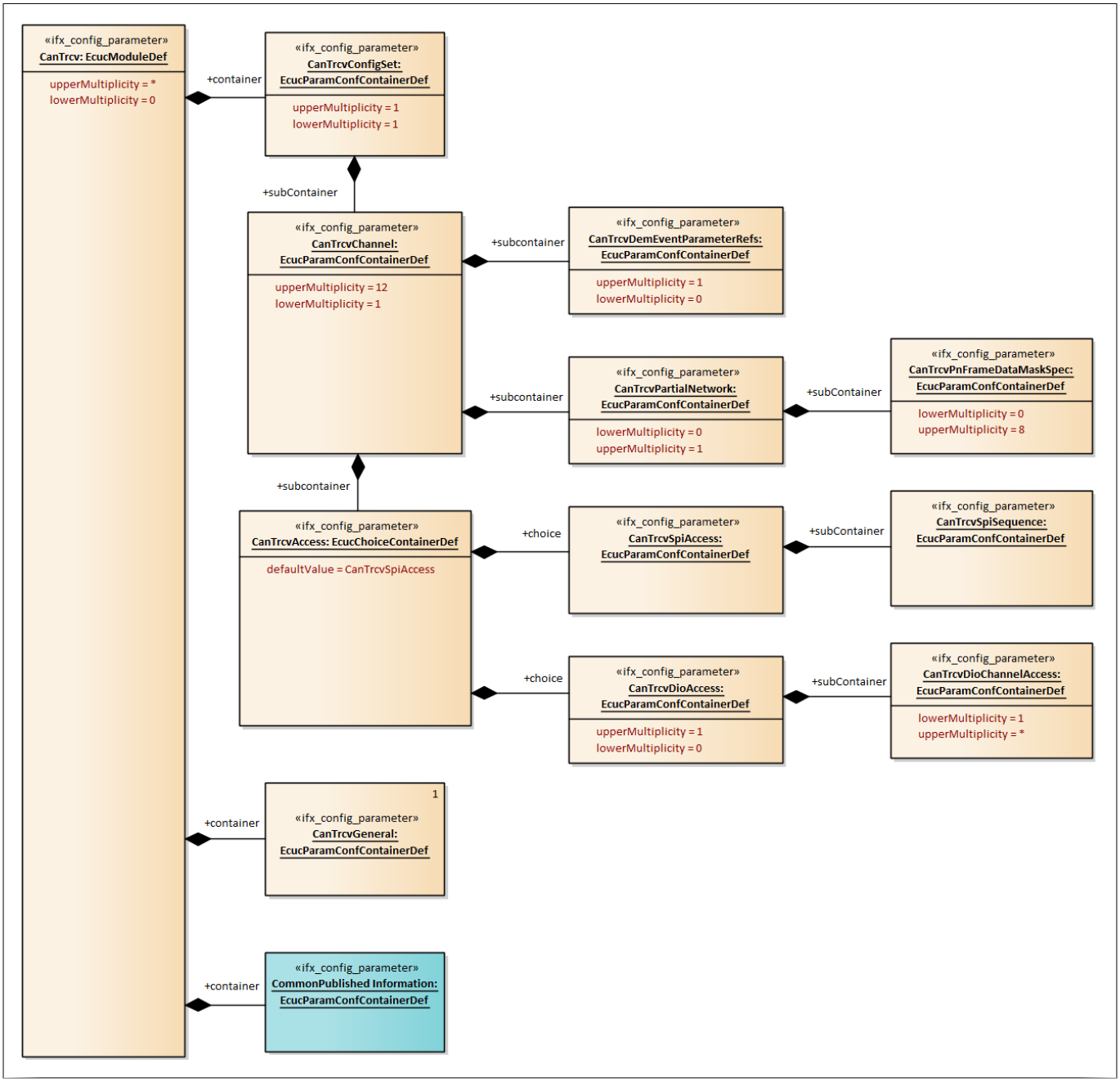

Figure 6 Container hierarchy along with their configuration parameters

1.3.1.1 Container: CanTrcvDemEventParameterRefs

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

Note: Since TLE9255W hardware cannot detect bus failure, this container is not applicable and made non-editable. This configuration container is not used in the code but it is listed for AUTOSAR compatibility.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: Pre-Compile

1.3.1.1.1 CANTRCV_E_BUS_ERROR

Table 4 Specification for CANTRCV_E_BUS_ERROR

Name	CANTRCV_E_BUS_ERROR		
Description	Reference to the DemEventParameter which shall be issued when bus error has occurred. *Note: Since TLE9255W hardware cannot detect bus failure, the module does not raise any Production errors. Therefore, this parameter is not applicable and made non-editable. This configuration parameter is not used in the code but it is listed for AUTOSAR compatibility.* 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: DemEventParameter		
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.		

1.3.1.2 Container: CommonPublished Information

This container contains the common published information of the TLE9255W CAN Transceiver driver.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.2.1 ArMajorVersion

Table 5 Specification for ArMajorVersion

Name	ArMajorVersion		
Description	Parameter provides the major version of the AUTOSAR Specification.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	4		

(table continues...)

Table 5 (continued) Specification for ArMajorVersion

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.		

1.3.1.2.2 ArMinorVersion

Table 6 Specification for ArMinorVersion

Name	`ArMinorVersion`		
Description	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.		

1.3.1.2.3 ArPatchVersion

Table 7 Specification for ArPatchVersion

Name	`ArPatchVersion`		
Description	Parameter provides the patch version of the AUTOSAR Specification.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per AUTOSAR patch version		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-

(table continues...)

Table 7 (continued) Specification for ArPatchVersion

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.		

1.3.1.2.4 ModuleId

Table 8 Specification for ModuleId

Name	`ModuleId`		
Description	Parameter provides the Module Id. *Note: Default value is set to 70, as this is the CAN Transceiver driver module ID.*		
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.		

1.3.1.2.5 Release

Table 9 Specification for Release

Name	`Release`		
Description	Specifies the derivate for which the configuration project is created. *Note: Default value is derived from the property file and represents the hardware derivative of the micro controller for which the CAN Transceiver driver is being configured.*		
Multiplicity	1..1	**Type**	EcucStringParamDef
Range	String		
Default value	As per hardware derivative		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-

(table continues...)

Table 9 (continued) Specification for Release

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.		

1.3.1.2.6 SwMajorVersion

Table 10 Specification for SwMajorVersion

Name	`SwMajorVersion`		
Description	Parameter provides the major version of the Software.		
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.		

1.3.1.2.7 SwMinorVersion

Table 11 Specification for SwMinorVersion

Name	`SwMinorVersion`		
Description	Parameter provides the minor version of the Software.		
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

(table continues...)

Table 11	(continued) Specification for SwMinorVersion
Dependency	-
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.1.2.8 SwPatchVersion

Table 12	Specification for SwPatchVersion		
Name	`SwPatchVersion`		
Description	Parameter provides the patch version of the Software.		
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.		

1.3.1.2.9 VendorApiInfix

Table 13	Specification for VendorApiInfix		
Name	`VendorApiInfix`		
Description	The parameter is used to specify the vendor specific name. *Note: Default value is set to W9255, as this is the unique name of the CAN Transceiver driver module provided by IFX.*		
Multiplicity	1..1	**Type**	EcucStringParamDef
Range	String		
Default value	W9255		
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.		

1.3.1.2.10 VendorId

Table 14 Specification for VendorId

Name	`VendorId`		
Description	Parameter provides the Vendor Id. *Note: Default value is set to 17, as this is the IFX vendor ID.*		
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.		

1.3.1.3 Container: CanTrcv

Configuration of the CAN Transceiver driver module.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: -

1.3.1.4 Container: CanTrcvChannel

This container gives CAN transceiver driver information about a single CAN transceiver (channel).

This container has lower multiplicity of 1 and upper multiplicity of 12 since number of CAN nodes supported in TC3xx is limited to 12.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: Pre-Compile

1.3.1.4.1 CanTrcvAccess

Table 15 Specification for CanTrcvAccess

Name	`CanTrcvAccess`		
Description	This container gives CAN Transceiver Driver information about access to a single CAN transceiver. *Note: Since TLE9255W hardware supports only SPI interface, the container CanTrcvSpiAccess is set as the default choice.*		
Multiplicity	1..1	**Type**	EcucChoiceContainer Def

(table continues...)

Table 15 (continued) Specification for CanTrcvAccess

Range	None		
Default value	CanTrcvSpiAccess		
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 16 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 17 Specification for CanTrcvChannelId

Name	`CanTrcvChannelId`

(table continues...)

Table 17 (continued) Specification for CanTrcvChannelId

Description	Unique identifier of the CAN Transceiver channel. *Note: The channel Id should be less than the number of channels configured. Minimum channel Id is selected as the default value. If channel Id's are not unique then configuration error will be reported.* *Note: Range of channel Id is modified as 0-11 since number of CAN nodes supported in TC3xx is limited to 12.*		
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 18 Specification for CanTrcvChannelUsed

Name	`CanTrcvChannelUsed`		
Description	This parameter specifies if the respective CAN transceiver channel is enabled or not.		
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 19 Specification for CanTrcvControlsPowerSupply

Name	`CanTrcvControlsPowerSupply`		
Description	Indicates if the ECU power supply is controlled by the transceiver. TRUE = Controlled by the transceiver FALSE = Not controlled by the transceiver *Note: Since TLE9255W hardware does not control the ECU power supply, this parameter is set FALSE by default 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 20 Specification for CanTrcvHwPnSupport

Name	`CanTrcvHwPnSupport`		
Description	Indicates whether TLE9255W hardware supports the selective wake-up feature. TRUE = Selective wakeup feature is supported by the transceiver FALSE = Selective wakeup feature is not supported by the transceiver *Note: Since the wakeup is always supported either by polling or by interrupt, this parameter is not dependent on CanTrcvWakeUpSupport .* *Note: Since TLE9255W hardware supports PN, this parameter is set TRUE by default and made non-editable.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	TRUE		

(table continues...)

Table 20 (continued) Specification for CanTrcvHwPnSupport

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.		

1.3.1.4.7 CanTrcvIcuChannelRef

Table 21 Specification for CanTrcvIcuChannelRef

Name	`CanTrcvIcuChannelRef`		
Description	Reference to the ICU channel for detecting the wakeups. 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	CanTrcvWakeUpSupport		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.8 CanTrcvInitState

Table 22 Specification for CanTrcvInitState

Name	`CanTrcvInitState`		
Description	State of CAN transceiver after call to CanTrcv_17_W9255_Init. *Note: Normal mode is set as default mode since the CAN messages can be transmitted and received after driver initialization.*		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef

(table continues...)

Table 22 (continued) Specification for CanTrcvInitState			
Range	CANTRCV_17_W9255_OP_MODE_NORMAL: Normal operation mode CANTRCV_17_W9255_OP_MODE_SLEEP: Sleep operation mode CANTRCV_17_W9255_OP_MODE_STANDBY: Standby operation mode		
Default value	CANTRCV_17_W9255_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 23 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 422, the range of this parameter is modified. Range is extended to 5Mbps since the hardware supports CAN FD data rates upto 5Mbps.* *Note: For AUTOSAR 440, 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 24 Specification for CanTrcvPorWakeupSourceRef

Name	`CanTrcvPorWakeupSourceRef`		
Description	This parameter contains symbolic name reference to specify the wakeup source for this channel that should be used in the calls to EcuM_SetWakeupEvent if POR flag is set in the TLE9255W hardware. Since the name of the dependent parameter is user configurable, the default value is set to NULL. *Note: Multiplicity is modified from 0-1 to 1-1 since TLE9255W hardware supports PN.*		
Multiplicity	1..1	**Type**	EcucSymbolicNameReferenceDef
Range	Reference to Node: EcuMWakeupSource		
Default value	NULL		
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.		

1.3.1.4.11 CanTrcvSyserrWakeupSourceRef

Table 25 Specification for CanTrcvSyserrWakeupSourceRef

Name	`CanTrcvSyserrWakeupSourceRef`		
Description	This parameter contains symbolic name reference to specify the wakeup source for this channel that should be used in the calls to EcuM_SetWakeupEvent if SYSERR flag is set in the TLE9255W hardware. Since the name of the dependent parameter is user configurable, the default value is set to NULL. *Note: Multiplicity is modified from 0-1 to 1-1 since TLE9255W hardware supports PN.*		
Multiplicity	1..1	**Type**	EcucSymbolicNameReferenceDef
Range	Reference to Node: EcuMWakeupSource		
Default value	NULL		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-

(table continues...)

Table 25 (continued) Specification for CanTrcvSyserrWakeupSourceRef

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.		

1.3.1.4.12 CanTrcvWakeupByBusUsed

Table 26 Specification for CanTrcvWakeupByBusUsed

Name	`CanTrcvWakeupByBusUsed`		
Description	Indicates whether wake up by bus functionality is enabled or not. This parameter does not depend on CanTrcvWakeUpSupport since the wake-up by bus functionality is always supported by the transceiver and can be enabled/disabled at channel level. *Note: WUP, WUF and LWU events are not reported if this parameter is FALSE.* *Note: Multiplicity is modified from 0-1 to 1-1 since TLE9255W hardware supports wake-up by bus functionality.*		
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.		

1.3.1.4.13 CanTrcvWakeupSourceRef

Table 27 Specification for CanTrcvWakeupSourceRef

Name	`CanTrcvWakeupSourceRef`

(table continues...)

Table 27 (continued) Specification for CanTrcvWakeupSourceRef

Description	This parameter contains a reference to the wakeup source for this channel in the EcuM configuration. Implementation Type: reference to EcuM_WakeupSourceType This reference is only needed if CanTrcvWakeupByBusUsed is true. 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: 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: CanTrcvConfigSet

This container contains the configuration parameters and sub containers of the AUTOSAR CanTrcv_17_W9255 module.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.5.1 CanTrcvSPICommRetries

Table 28 Specification for CanTrcvSPICommRetries

Name	`CanTrcvSPICommRetries`		
Description	Indicates the maximum number of communication retries in case of a failed SPI communication. If configured value is '0', no retry is allowed.		
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**	-

(table continues...)

Table 28 (continued) Specification for CanTrcvSPICommRetries

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

1.3.1.5.2 CanTrcvSPICommTimeout

Table 29 Specification for CanTrcvSPICommTimeout

Name	`CanTrcvSPICommTimeout`		
Description	Indicates the maximum time allowed to the CAN Transceiver for replying to an 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 hardware capacity. *Note: This parameter is made non-editable as synchronous implementation of SPI driver is used.* *Note: Since this parameter is non-editable, there is no dependency on CanTrcvSpiSequence parameter.*		
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.		

1.3.1.6 Container: CanTrcvDioAccess

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.

If CAN transceiver hardware has no DIO interface, there is no instance of this container.

Note: Since TLE9255W transceiver hardware has no DIO interface, there is no instance of this container and its parameters.

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

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: Pre-Compile

1.3.1.7 Container: CanTrcvDioChannelAccess

Container gives DIO channel access by single CAN transceiver channel.

Note: Since TLE9255W transceiver hardware has no DIO interface, there is no instance of this container.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: -

1.3.1.7.1 CanTrcvDioSymNameRef

Table 30 Specification for CanTrcvDioSymNameRef

Name	`CanTrcvDioSymNameRef`		
Description	This parameter gives the reference to a configured DIO channel. *Note: This configuration parameter is not used in the code but it is added only for AUTOSAR compatibility.*		
Multiplicity	1..1	**Type**	EcucChoiceReferenceDef
Range	Reference to Node: DioChannel		
Default value	NULL		
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 31 Specification for CanTrcvHardwareInterfaceName

Name	`CanTrcvHardwareInterfaceName`		
Description	This parameter specifies CAN transceiver hardware interface name. It is typically the name of a CAN transceiver pin.		
Multiplicity	1..1	**Type**	EcucStringParamDef
Range	String		
Default value	NULL		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-

(table continues...)

Table 31 (continued) Specification for CanTrcvHardwareInterfaceName

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 basic information about CAN transceiver driver.
Post-Build Variant Multiplicity: -
Multiplicity Configuration Class: -

1.3.1.8.1 CanTrcvDevErrorDetect

Table 32 Specification for CanTrcvDevErrorDetect

Name	`CanTrcvDevErrorDetect`		
Description	Parameter enables or disables the Default Error Tracer (DET) detection and reporting. *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 33 Specification for CanTrcvEcucPartitionRef

Name	`CanTrcvEcucPartitionRef`
Description	Maps the CanTrcv driver to zero or multiple ECUC partitions to make the modules API available in this partition. The CanTrcv driver 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.*

(table continues...)

Table 33 (continued) Specification for CanTrcvEcucPartitionRef

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 34 Specification for CanTrcvGetVersionInfo

Name	`CanTrcvGetVersionInfo`		
Description	Parameter adds or removes the API CanTrcv_17_W9255_GetVersionInfo() 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 35 Specification for CanTrcvIndex

Name	`CanTrcvIndex`

(table continues...)

Table 35 (continued) Specification for CanTrcvIndex

Description	Specifies the Instance Id of this module instance. If only one instance is present it shall have the Id 0. *Note: Default value is set to 0 as it is the minimum value supported.*		
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 36 Specification for CanTrcvMainFunctionDiagnosticsPeriod

Name	`CanTrcvMainFunctionDiagnosticsPeriod`		
Description	This parameter describes the period for cyclic call to CanTrcv_17_W9255_MainFunctionDiagnostics. Unit of the parameter is seconds. *Note:* *- In AUTOSAR 422, this parameter range is 0 to 65.535 seconds. The upper range of the parameter is restricted to 65.535 seconds in AUTOSAR 440 as well.* *- Since CanTrcv_17_W9255_MainFunctionDiagnostics API is not provided by the driver, this parameter is not applicable and made non-editable.*		
Multiplicity	0..1	**Type**	EcucFloatParamDef
Range	0.001 - 65.535		
Default value	0.001		
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 37 Specification for CanTrcvMainFunctionPeriod

Name	`CanTrcvMainFunctionPeriod`		
Description	This parameter describes the period for cyclic call to CanTrcv_17_W9255_MainFunction. Unit of the parameter is seconds. It is advisory for all the communication modules to set the default value of this parameter to 0.005 seconds. *Note: In AUTOSAR 422, this parameter range is 0 to 65.535 seconds. The upper range of the parameter is restricted to 65.535 seconds in AUTOSAR 440 as well.*		
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.		

1.3.1.8.7 CanTrcvRunTimeErrorDetect

Table 38 Specification for CanTrcvRunTimeErrorDetect

Name	`CanTrcvRunTimeErrorDetect`		
Description	Switches the Runtime Error detection and notification ON or OFF. - true: enabled (ON). - false: disabled (OFF). *Note: The default value of this parameter is set to TRUE to ensure the runtime error detection during the product lifecycle.*		
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**	-

(table continues...)

Table 38 (continued) Specification for CanTrcvRunTimeErrorDetect

Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar version 4.4.0.		

1.3.1.8.8 CanTrcvTimerType

Table 39 Specification for CanTrcvTimerType

Name	`CanTrcvTimerType`		
Description	Type of the Time Service Predefined Timer. *Note: Default value of this parameter is set to 'None' since McalLib APIs are used to realize the wait time. The parameter is made non-editable.*		
Multiplicity	0..1	**Type**	EcucEnumerationParamDef
Range	None: No timer configured. Timer_1us16bit: 16 bit 1us timer		
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.		

1.3.1.8.9 CanTrcvVersionInfoApi

Table 40 Specification for CanTrcvVersionInfoApi

Name	`CanTrcvVersionInfoApi`		
Description	Parameter adds or removes the API CanTrcv_17_W9255_GetVersionInfo() 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 continues...)

Table 40 (continued) Specification for CanTrcvVersionInfoApi

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

1.3.1.8.10 CanTrcvWaitTime

Table 41 Specification for CanTrcvWaitTime

Name	`CanTrcvWaitTime`		
Description	Wait time for transceiver mode changes. Unit of the parameter is seconds. The minimum and default values are set to 20 micro seconds as it is the worst case wait time needed for a mode change. The parameter is made non-editable. *Note: The lower multiplicity of this parameter is set to 1 as the transceiver needs time for mode change.*		
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.		

1.3.1.8.11 CanTrcvWakeUpSupport

Table 42 Specification for CanTrcvWakeUpSupport

Name	`CanTrcvWakeUpSupport`

(table continues...)

Table 42 (continued) Specification for CanTrcvWakeUpSupport

Description	Informs whether wake up is supported by polling or interrupt. *Note: CANTRCV_17_W9255_WAKEUP_NOT_SUPPORTED is not provided, since the TLE9255W hardware supports wake up functionality.* *Note: A new option CANTRCV_17_W9255_WAKE_UP_BY_INTERRUPT is added which supports wake-up by interrupt functionality.* *Note: CanTrcv_17_W9255_MainFunction is available only in the case of wakeup support by polling.*		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	CANTRCV_17_W9255_WAKE_UP_BY_INTERRUPT: Wake up by interrupt CANTRCV_17_W9255_WAKE_UP_BY_POLLING: Wake up by polling		
Default value	CANTRCV_17_W9255_WAKE_UP_BY_POLLING		
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.

This configuration container always exists for every channel, since parameter CanTrcvHwPnSupport is always set TRUE and made non-editable.

This container have a lower multiplicity of 0 and upper multiplicity of 1.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: Pre-Compile

1.3.1.9.1 CanTrcvBaudRate

Table 43 Specification for CanTrcvBaudRate

Name	`CanTrcvBaudRate`

(table continues...)

Table 43 (continued) Specification for CanTrcvBaudRate

Description	Baud rate to be set for PN frame wake-up. Unit of the parameter is kbps. According to AUTOSAR 422, this parameter range is 0 to 1000 kbit/s. According to AUTOSAR 440, this parameter range is 0 to 12000 kbit/s.TLE9255W hardware supports the following baud rates for the Selective Wake unit: 125 kbit/s, 250 kbit/s, 500 kbit/s and 1Mbit/s. Hence, the range is restricted to 125 kbit/s -1000 kbit/s for both the AUTOSAR versions. *Note :Default value of this parameter is set to 500 kbit/s since CAN standard messages are usually of the same baudrate value.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	125 - 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
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 TLE9255W hardware cannot detect bus error, this parameter is not applicable. Hence, this parameter is set FALSE by default and made non-editable. This configuration parameter is not used in the code but it is listed for AUTOSAR compatibility.*		
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 continues...)

Table 44 (continued) Specification for CanTrcvBusErrFlag

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	Indicates whether extended or standard CAN Id is used. TRUE = Extended CAN identifier is used FALSE = Standard CAN identifier is used		
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	CanTrcvHwPnSupport		
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	Indicates whether the selective wake-up functionality is enabled or disabled in CAN Transceiver hardware. TRUE = Selective wakeup feature is enabled in the transceiver hardware. FALSE = Selective wakeup feature is disabled in the transceiver hardware.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		

(table continues...)

Table 46 (continued) Specification for CanTrcvPnEnabled

Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	CanTrcvHwPnSupport		
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	CAN ID of the Wake-up Frame. *Note: Default value is set to 0 as it is the minimum value supported.*		
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	CanTrcvHwPnSupport		
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	ID Mask for the selective activation of the CAN transceiver. It is used to enable WUF on a group of IDs. *Note: Default value is set to 4294967295 as it activates WUF only on one ID.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 4294967295		
Default value	4294967295		

(table continues...)

Table 48 (continued) Specification for CanTrcvPnFrameCanIdMask

Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	CanTrcvHwPnSupport		
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	Indicates the Data Length of the WUF. Default value of this parameter is set to 1 as recommended by AUTOSAR. *Note: Minimum value of the range is deviated from AUTOSAR requirement and changed to 1, since AUTOSAR SWS states "Although WUF with DLC=0 is technically possible, it is explicitly not wanted" in CanTrcvBaudRate parameter dependency.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	1 - 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	CanTrcvHwPnSupport		
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`

(table continues...)

Table 50 (continued) Specification for CanTrcvPowerOnFlag

Description	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 Power On Reset (POR) flag is available and is managed by the transceiver, this parameter is set TRUE by default and made non-editable.*		
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.		

1.3.1.10 Container: CanTrcvPnFrameDataMaskSpec

Defines data mask to be used on the received CAN frames in order to determine if the transceiver must be woken up by the received Wake-up Frame.

Note: Since the minimum value of CanTrcvPnFrameDlc is 1, at least one data mask needs to be configured if PN is enabled.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: Pre-Compile

1.3.1.10.1 CanTrcvPnFrameDataMask

Table 51 Specification for CanTrcvPnFrameDataMask

Name	`CanTrcvPnFrameDataMask`		
Description	Defines the data mask of the WUF at the configured index. *Note: Default value is set to 255 as this allows a wide range of data.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	255		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-

(table continues...)

Table 51 (continued) Specification for CanTrcvPnFrameDataMask

Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	CanTrcvHwPnSupport		
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	Holds the index of the data mask in the configured WUF. *Note: Default value is set to 0 as it is the minimum value supported.*		
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	CanTrcvHwPnSupport		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.11 Container: CanTrcvSpiAccess

Container gives CAN transceiver driver information about accessing SPI.

Note: Multiplicity is modified from 0-1 to 1-1 since TLE9255W hardware transceiver hardware uses SPI interface for communication.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.12 Container: CanTrcvSpiSequence

Container gives CAN transceiver driver information about one SPI sequence.

Note: Multiplicity is modified from 1- to 1-1 since one sequence is enough for one transceiver channel for communication.*

Post-Build Variant Multiplicity: -

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 This parameter is set to true by default and made non-editable as the CAN Transceiver driver always accesses SPI synchronously. *Note: Multiplicity is modified from 0-1 to 1-1 since TLE9255W hardware transceiver hardware uses SPI interface for communication.*		
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.		

1.3.1.12.2 CanTrcvSpiSequenceName

Table 54 Specification for CanTrcvSpiSequenceName

Name	`CanTrcvSpiSequenceName`		
Description	Reference to a SPI sequence configuration container. Since the name of the dependent parameter is user configurable, the default value is set to NULL. *Note: The lower multiplicity of this parameter is set to 1 since TLE9255W transceiver hardware uses SPI interface for communication and needs at least one sequence per channel. The upper multiplicity is set to 1 since one sequence is enough for one transceiver channel for communication.*		
Multiplicity	1..1	**Type**	EcucSymbolicNameReferenceDef
Range	Reference to Node: SpiSequence		
Default value	NULL		

(table continues...)

Table 54 (continued) Specification for CanTrcvSpiSequenceName

Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
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_W9255 driver.

1.3.2.1 CanTrcv_17_W9255_ConfigType

Table 55 Specification for CanTrcv_17_W9255_ConfigType

Syntax	`CanTrcv_17_W9255_ConfigType`	
Type	void	
File	`CanTrcv_17_W9255.h`	
Range	None	
Description	This is the type of the external data structure containing the overall initialization data for the CAN transceiver driver and settings affecting all transceivers. *Note: Since CanTrcv_17_W9255 driver is implemented as a Pre-Compile module, this type is implemented as of type void as the module supports single configuration variant.*	
Source	IFX	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.2 CanTrcv_17_W9255_PNActivationType

Table 56 Specification for CanTrcv_17_W9255_PNActivationType

Syntax	`CanTrcv_17_W9255_PNActivationType`	
Type	Enumeration	
File	`CanTrcv_17_W9255.h`	
Range	0 - CANTRCV_17_W9255_PN_DISABLED	PN wakeup functionality in CAN Transceiver is disabled.
	1 - CANTRCV_17_W9255_PN_ENABLED	PN wakeup functionality in CAN Transceiver is enabled.
Description	Datatype used for describing whether PN wakeup functionality in the CAN Transceiver is enabled or disabled.	

(table continues...)

Table 56 (continued) Specification for CanTrcv_17_W9255_PNActivationType

Source	AUTOSAR
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.2.3 CanTrcv_17_W9255_TrcvFlagStateType

Table 57 Specification for CanTrcv_17_W9255_TrcvFlagStateType

Syntax	`CanTrcv_17_W9255_TrcvFlagStateType`	
Type	Enumeration	
File	`CanTrcv_17_W9255.h`	
Range	0 - CANTRCV_17_W9255_FLAG_CLEARED	The flag is cleared in the transceiver hardware.
	1 - CANTRCV_17_W9255_FLAG_SET	The flag is set in the transceiver hardware.
Description	Provides the state of a flag in the transceiver hardware.	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3 Functions - APIs

This section lists all the APIs of the CanTrcv_17_W9255 driver.

1.3.3.1 CanTrcv_17_W9255_Init

Table 58 Specification for `CanTrcv_17_W9255_Init` API

Syntax	`void CanTrcv_17_W9255_Init ( const CanTrcv_17_W9255_ConfigType * const ConfigPtr )`	
Service ID	0x00	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant	
Parameters (in)	ConfigPtr	Pointer to driver configuration. Note: Since CAN Transceiver is implemented as a pre-compile module, a null pointer shall be passed as the parameter by the caller of this API.
Parameters (out)	-	-

(table continues...)

Table 58 (continued) Specification for `CanTrcv_17_W9255_Init` API

Parameters (in - out)	-	-
Return	void	-
Description	This API initializes all the connected CAN Transceivers. The registers of the TLE9255W hardware are initialized as per the configuration. The CAN Transceiver initialization status is set at the end of the initialization function execution.	
Source	AUTOSAR	
Error handling	CANTRCV_17_W9255_E_NO_TRCV_CONTROL, CANTRCV_17_W9255_E_INIT_FAILED	
Configuration dependencies	-	
User hints	-	
SFR accessed	CPU_CORE_ID(r), DMA_CH_ADICR(rw), DMA_CH_CHCFGR(w), DMA_CH_CHCSR(w), DMA_CH_DADR(w), DMA_CH_RDCRCR(w), DMA_CH_SADR(w), DMA_CH_SDCRCR(w), DMA_CH_SHADR(rw), DMA_TSR(rw), P_OMR(w), QSPI_BACONENTRY(w), QSPI_DATAENTRY(w), QSPI_ECON(w), QSPI_FLAGSCLEAR(w), QSPI_GLOBALCON(rw), QSPI_GLOBALCON1(w), QSPI_MC(w), QSPI_MCCON(w), QSPI_RXEXIT(r), QSPI_STATUS(r), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.2 CanTrcv_17_W9255_SetOpMode

Table 59 Specification for `CanTrcv_17_W9255_SetOpMode` API

Syntax	`Std_ReturnType CanTrcv_17_W9255_SetOpMode ( const uint8 Transceiver, const CanTrcv_TrcvModeType OpMode )`	
Service ID	0x01	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Reentrant for different transceivers	
Parameters (in)	Transceiver OpMode	CAN transceiver to which API call has to be applied. This parameter has a valid range of 0-11. This parameter contains the desired operating mode
Parameters (out)	-	-

(table continues...)

Table 59 (continued) Specification for `CanTrcv_17_W9255_SetOpMode` API

Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK: If the mode is changed successfully E_NOT_OK: If there is SPI communication failure or a development error occurs.
Description	This API sets the mode of the requested Transceiver to the value OpMode. If PN is enabled, the API checks for POR and SYSERR flags. If POR flag is set, transceiver is reinitialized and if SYSERR flag is set, transceiver is reinitialized for PN functionality.	
Source	AUTOSAR	
Error handling	CANTRCV_17_W9255_E_PARAM_TRCV_OPMODE, CANTRCV_17_W9255_E_NO_TRCV_CONTROL, CANTRCV_17_W9255_E_INVALID_TRANSCEIVER, CANTRCV_17_W9255_E_TRCV_NOT_STANDBY, CANTRCV_17_W9255_E_UNINIT, CANTRCV_17_W9255_E_TRCV_NOT_NORMAL	
Configuration dependencies	-	
User hints	-	
SFR accessed	CPU_CORE_ID(r), DMA_CH_ADICR(rw), DMA_CH_CHCFGR(w), DMA_CH_CHCSR(w), DMA_CH_DADR(w), DMA_CH_RDCRCR(w), DMA_CH_SADR(w), DMA_CH_SDCRCR(w), DMA_CH_SHADR(rw), DMA_TSR(rw), P_OMR(w), QSPI_BACONENTRY(w), QSPI_DATAENTRY(w), QSPI_ECON(w), QSPI_FLAGSCLEAR(w), QSPI_GLOBALCON(rw), QSPI_GLOBALCON1(w), QSPI_MC(w), QSPI_MCCON(w), QSPI_RXEXIT(r), QSPI_STATUS(r), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.3 CanTrcv_17_W9255_GetOpMode

Table 60 Specification for `CanTrcv_17_W9255_GetOpMode` API

Syntax	`Std_ReturnType CanTrcv_17_W9255_GetOpMode ( const uint8 Transceiver, CanTrcv_TrcvModeType * const OpMode )`
Service ID	0x02
Sync/Async	Synchronous
Safety Level	Refer to the release notes for the safety related info
Re-entrancy	Reentrant

(table continues...)

Table 60 (continued) Specification for `CanTrcv_17_W9255_GetOpMode` API

Parameters (in)	Transceiver	CAN transceiver to which API call has to be applied. This parameter has a valid range of 0-11.
Parameters (out)	OpMode	Pointer to operation mode of the transceiver the API is applied to.
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK: If the operation mode was detected. E_NOT_OK: If SPI communication failure or a development error occurs.
Description	This API gets the mode of the Transceiver and returns it in parameter OpMode.	
Source	AUTOSAR	
Error handling	CANTRCV_17_W9255_E_UNINIT, CANTRCV_17_W9255_E_NO_TRCV_CONTROL, CANTRCV_17_W9255_E_INVALID_TRANSCEIVER, CANTRCV_17_W9255_E_PARAM_POINTER	
Configuration dependencies	-	
User hints	-	
SFR accessed	CPU_CORE_ID(r), DMA_CH_ADICR(rw), DMA_CH_CHCFGR(w), DMA_CH_CHCSR(w), DMA_CH_DADR(w), DMA_CH_RDCRCR(w), DMA_CH_SADR(w), DMA_CH_SDCRCR(w), DMA_CH_SHADR(rw), DMA_TSR(rw), P_OMR(w), QSPI_BACONENTRY(w), QSPI_DATAENTRY(w), QSPI_ECON(w), QSPI_FLAGSCLEAR(w), QSPI_GLOBALCON(rw), QSPI_GLOBALCON1(w), QSPI_MC(w), QSPI_MCCON(w), QSPI_RXEXIT(r), QSPI_STATUS(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.4 CanTrcv_17_W9255_GetBusWuReason

Table 61 Specification for `CanTrcv_17_W9255_GetBusWuReason` API

Syntax	`Std_ReturnType CanTrcv_17_W9255_GetBusWuReason ( const uint8 Transceiver, CanTrcv_TrcvWakeupReasonType * const Reason )`
Service ID	0x03
Sync/Async	Synchronous
Safety Level	Refer to the release notes for the safety related info
Re-entrancy	Reentrant

(table continues...)

Table 61 (continued) Specification for `CanTrcv_17_W9255_GetBusWuReason` API

Parameters (in)	Transceiver	CAN transceiver to which API call has to be applied. This parameter has a valid range of 0-11.
Parameters (out)	Reason	Pointer to wake up reason of the transceiver the API is applied to. This parameter can hold the following valid enum values as per CanTrcv_TrcvWakeupReasonType requirement: - CANTRCV_WU_ERROR - CANTRCV_WU_NOT_SUPPORTED - CANTRCV_WU_BY_BUS - CANTRCV_WU_INTERNALLY - CANTRCV_WU_RESET - CANTRCV_WU_POWER_ON - CANTRCV_WU_BY_PIN - CANTRCV_WU_BY_SYSERR
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : If wake up reason was detected E_NOT_OK: If a development error occurs
Description	This API gets the wake-up reason for the requested Transceiver and returns it in the parameter reason. The driver supports the following wake-up reasons: - Wake-up by bus - CANTRCV_WU_BY_BUS - Wake-up by pin - CANTRCV_WU_BY_PIN - Wake-up due to an ECU reset after power on - CANTRCV_WU_POWER_ON - Wake up due to transition to normal mode - CANTRCV_WU_INTERNALLY - Wake-up due to hardware related device failure (SYSERR) - CANTRCV_WU_BY_SYSERR - Wake-up due to an ECU reset - CANTRCV_WU_RESET The driver does not support the following wake-up reasons due to hardware limitations: - Wake up reason not detected due to an error - CANTRCV_WU_ERROR - Information for the wake-up reason not supported - CANTRCV_WU_NOT_SUPPORTED	
Source	AUTOSAR	
Error handling	CANTRCV_17_W9255_E_UNINIT, CANTRCV_17_W9255_E_INVALID_TRANSCEIVER, CANTRCV_17_W9255_E_PARAM_POINTER	
Configuration dependencies	-	
User hints	-	
SFR accessed	-	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.5 CanTrcv_17_W9255_GetVersionInfo

Table 62 Specification for `CanTrcv_17_W9255_GetVersionInfo` API

Syntax	`void CanTrcv_17_W9255_GetVersionInfo ( Std_VersionInfoType * const versioninfo )`	
Service ID	0x04	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Reentrant	
Parameters (in)	-	-
Parameters (out)	versioninfo	Pointer to where to store the version information of the CanTrcv_17_W9255 driver.
Parameters (in - out)	-	-
Return	void	-
Description	This API gets the version of the module and returns it in versionInfo. *Note : In AUTOSAR 422, enabling and disabling of this API depends on CanTrcvGetVersionInfo parameter. In AUTOSAR 440, enabling and disabling of this API depends on CanTrcvVersionInfoApi parameter.*	
Source	AUTOSAR	
Error handling	CANTRCV_17_W9255_E_PARAM_POINTER	
Configuration dependencies	CanTrcvVersionInfoApi,CanTrcvGetVersionInfo	
User hints	-	
SFR accessed	-	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.6 CanTrcv_17_W9255_SetWakeupMode

Table 63 Specification for `CanTrcv_17_W9255_SetWakeupMode` API

Syntax	`Std_ReturnType CanTrcv_17_W9255_SetWakeupMode ( const uint8 Transceiver, const CanTrcv_TrcvWakeupModeType TrcvWakeupMode )`
Service ID	0x05
Sync/Async	Synchronous

(table continues...)

Table 63 (continued) Specification for `CanTrcv_17_W9255_SetWakeupMode` API

Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Reentrant for different transceivers	
Parameters (in)	Transceiver TrcvWakeupMode	CAN transceiver to which API call has to be applied. This parameter has a valid range of 0-11. Requested transceiver wakeup mode type
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK: Wakeup state changed to the requested mode. E_NOT_OK: If SPI communication fails, wake-up by bus is disabled or a development error occurs. The previous state has not been changed.
Description	This API enables, disables or clears wake-up events of the Transceiver according to TrcvWakeupMode. Enable mode: The CAN Transceiver driver reports wake-up event during wake-up event detection. Besides, the driver also reports wake-up event if it has a stored wake-up event pending for the addressed transceiver. Disable mode: The wake-up events are disabled on the addressed transceiver. Any wake-up event occurred during this time is stored internally. Clear mode: Stored wake-up event and hardware wake flags are cleared on the addressed transceiver. CANTRCV_17_W9255_E_NO_TRCV_CONTROL DET will be checked only in Clear mode since no communication with the hardware happens in the other two modes.	
Source	AUTOSAR	
Error handling	CANTRCV_17_W9255_E_PARAM_TRCV_WAKEUP_MODE, CANTRCV_17_W9255_E_INVALID_TRANSCEIVER, CANTRCV_17_W9255_E_UNINIT, CANTRCV_17_W9255_E_NO_TRCV_CONTROL	
Configuration dependencies	-	
User hints	-	
SFR accessed	CPU_CORE_ID(r), DMA_CH_ADICR(rw), DMA_CH_CHCFGR(w), DMA_CH_CHCSR(w), DMA_CH_DADR(w), DMA_CH_RDCRCR(w), DMA_CH_SADR(w), DMA_CH_SDCRCR(w), DMA_CH_SHADR(rw), DMA_TSR(rw), P_OMR(w), QSPI_BACONENTRY(w), QSPI_DATAENTRY(w), QSPI_ECON(w), QSPI_FLAGSCLEAR(w), QSPI_GLOBALCON(rw), QSPI_GLOBALCON1(w), QSPI_MC(w), QSPI_MCCON(w), QSPI_RXEXIT(r), QSPI_STATUS(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.7 CanTrcv_17_W9255_CheckWakeup

Table 64 Specification for `CanTrcv_17_W9255_CheckWakeup` API

Syntax	`Std_ReturnType CanTrcv_17_W9255_CheckWakeup ( const uint8 Transceiver )`	
Service ID	0x07	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Reentrant	
Parameters (in)	Transceiver	CAN transceiver to which API call has to be applied. This parameter has a valid range of 0-11.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : If a valid interrupt is detected. E_NOT_OK : If SPI communication fails, development error is detected, wake up by bus is disabled for the channel called, or a false interrupt is detected.
Description	This service is called by the underlying CanIf, in cases of polling or interrupt mode. This API validates wake-up event on the requested transceiver channel and if true, reports it to EcuM if the wakeup mode is enabled, clears the wake flags on the hardware and changes the mode of the respective channel to Normal.	
Source	AUTOSAR	
Error handling	CANTRCV_17_W9255_E_UNINIT, CANTRCV_17_W9255_E_INVALID_TRANSCEIVER	
Configuration dependencies	-	
User hints	-	
SFR accessed	CPU_CORE_ID(r), DMA_CH_ADICR(rw), DMA_CH_CHCFGR(w), DMA_CH_CHCSR(w), DMA_CH_DADR(w), DMA_CH_RDCRCR(w), DMA_CH_SADR(w), DMA_CH_SDCRCR(w), DMA_CH_SHADR(rw), DMA_TSR(rw), P_OMR(w), QSPI_BACONENTRY(w), QSPI_DATAENTRY(w), QSPI_ECON(w), QSPI_FLAGSCLEAR(w), QSPI_GLOBALCON(rw), QSPI_GLOBALCON1(w), QSPI_MC(w), QSPI_MCCON(w), QSPI_RXEXIT(r), QSPI_STATUS(r), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.8 CanTrcv_17_W9255_CheckWakeFlag

Table 65 Specification for `CanTrcv_17_W9255_CheckWakeFlag` API

Syntax	`Std_ReturnType CanTrcv_17_W9255_CheckWakeFlag ( const uint8 Transceiver )`	
Service ID	0x0e	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant	
Parameters (in)	Transceiver	CAN transceiver to which API call has to be applied. This parameter has a valid range of 0-11.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK: If the request for checking the wakeup flags has been accepted. E_NOT_OK: If the request for checking the wakeup flags has not been accepted, wake up by bus is disabled, development error occurred or if SPI communication fails.
Description	This API checks the status of the wake-up flags from the transceiver hardware and informs the CanIf with the callback notification CanIf_CheckTrcvWakeFlagIndication, that the wake flags of the CAN transceiver with the corresponding Transceiver ID have been checked.	
Source	AUTOSAR	
Error handling	CANTRCV_17_W9255_E_INVALID_TRANSCEIVER	
Configuration dependencies	-	
User hints	-	
SFR accessed	CPU_CORE_ID(r), DMA_CH_ADICR(rw), DMA_CH_CHCFGR(w), DMA_CH_CHCSR(w), DMA_CH_DADR(w), DMA_CH_RDCRCR(w), DMA_CH_SADR(w), DMA_CH_SDCRCR(w), DMA_CH_SHADR(rw), DMA_TSR(rw), P_OMR(w), QSPI_BACONENTRY(w), QSPI_DATAENTRY(w), QSPI_ECON(w), QSPI_FLAGSCLEAR(w), QSPI_GLOBALCON(rw), QSPI_GLOBALCON1(w), QSPI_MC(w), QSPI_MCCON(w), QSPI_RXEXIT(r), QSPI_STATUS(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.9 CanTrcv_17_W9255_ClearTrcvTimeoutFlag

Table 66 Specification for `CanTrcv_17_W9255_ClearTrcvTimeoutFlag` API

Syntax	`Std_ReturnType CanTrcv_17_W9255_ClearTrcvTimeoutFlag ( const uint8 Transceiver )`	
Service ID	0x0c	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant	
Parameters (in)	Transceiver	CAN transceiver to which API call has to be applied. This parameter has a valid range of 0-11.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK: If the timeout flag is successfully cleared. E_NOT_OK: If SPI communication failure or a development error occurs.
Description	This API clears the status of the timeout flag in the transceiver hardware. Since the configuration parameter CanTrcvHwPnSupport is always TRUE, this API is always available. The timeout flag indicates whether or not the TLE9255W hardware has entered the Selective Sleep Sub-Mode at least once.	
Source	AUTOSAR	
Error handling	CANTRCV_17_W9255_E_INVALID_TRANSCEIVER	
Configuration dependencies	CanTrcvHwPnSupport	
User hints	-	
SFR accessed	CPU_CORE_ID(r), DMA_CH_ADICR(rw), DMA_CH_CHCFGR(w), DMA_CH_CHCSR(w), DMA_CH_DADR(w), DMA_CH_RDCRCR(w), DMA_CH_SADR(w), DMA_CH_SDCRCR(w), DMA_CH_SHADR(rw), DMA_TSR(rw), P_OMR(w), QSPI_BACONENTRY(w), QSPI_DATAENTRY(w), QSPI_ECON(w), QSPI_FLAGSCLEAR(w), QSPI_GLOBALCON(rw), QSPI_GLOBALCON1(w), QSPI_MC(w), QSPI_MCCON(w), QSPI_RXEXIT(r), QSPI_STATUS(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.10 CanTrcv_17_W9255_ClearTrcvWufFlag

Table 67 Specification for `CanTrcv_17_W9255_ClearTrcvWufFlag` API

Syntax	`Std_ReturnType CanTrcv_17_W9255_ClearTrcvWufFlag ( const uint8 Transceiver )`	
Service ID	0x0a	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Reentrant for different transceivers	
Parameters (in)	Transceiver	CAN transceiver to which API call has to be applied. This parameter has a valid range of 0-11.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK: If the WUF flag has been cleared. E_NOT_OK: If SPI communication failure or a development error occurs.
Description	This API clears the WUF flag in the transceiver hardware. Since the configuration parameter CanTrcvHwPnSupport is always TRUE, this API is always available.	
Source	AUTOSAR	
Error handling	CANTRCV_17_W9255_E_NO_TRCV_CONTROL, CANTRCV_17_W9255_E_INVALID_TRANSCEIVER, CANTRCV_17_W9255_E_UNINIT	
Configuration dependencies	CanTrcvHwPnSupport	
User hints	-	
SFR accessed	CPU_CORE_ID(r), DMA_CH_ADICR(rw), DMA_CH_CHCFGR(w), DMA_CH_CHCSR(w), DMA_CH_DADR(w), DMA_CH_RDCRCR(w), DMA_CH_SADR(w), DMA_CH_SDCRCR(w), DMA_CH_SHADR(rw), DMA_TSR(rw), P_OMR(w), QSPI_BACONENTRY(w), QSPI_DATAENTRY(w), QSPI_ECON(w), QSPI_FLAGSCLEAR(w), QSPI_GLOBALCON(rw), QSPI_GLOBALCON1(w), QSPI_MC(w), QSPI_MCCON(w), QSPI_RXEXIT(r), QSPI_STATUS(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.11 CanTrcv_17_W9255_GetTrcvSystemData

Table 68 Specification for `CanTrcv_17_W9255_GetTrcvSystemData` API

Syntax	`Std_ReturnType CanTrcv_17_W9255_GetTrcvSystemData ( const uint8 Transceiver, uint32 * const TrcvSysData )`	
Service ID	0x09	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant	
Parameters (in)	Transceiver	CAN transceiver to which API call has to be applied. This parameter has a valid range of 0-11.
Parameters (out)	TrcvSysData	This parameter holds the selective wake status, error status, transceiver status and wake-up event status information. The first 8 bits of LSB contain the data stored in TRANS_STAT register, the next 8 bits contain the data in SWK_ECNT_STAT register, the next 8 bits depict the data stored in WAKE_STAT register and the last 8 bits contain the data stored in SWK_STAT register.
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK: If the transceiver status is successfully read. E_NOT_OK: If SPI communication failure or a development error occurs.
Description	This API reads the transceiver status data and returns it through parameter TrcvSysData. Since the configuration parameter CanTrcvHwPnSupport is always TRUE, this API is always available.	
Source	AUTOSAR	
Error handling	CANTRCV_17_W9255_E_UNINIT, CANTRCV_17_W9255_E_NO_TRCV_CONTROL, CANTRCV_17_W9255_E_INVALID_TRANSCEIVER, CANTRCV_17_W9255_E_PARAM_POINTER	
Configuration dependencies	CanTrcvHwPnSupport	
User hints	-	
SFR accessed	CPU_CORE_ID(r), DMA_CH_ADICR(rw), DMA_CH_CHCFGR(w), DMA_CH_CHCSR(w), DMA_CH_DADR(w), DMA_CH_RDCRCR(w), DMA_CH_SADR(w), DMA_CH_SDCRCR(w), DMA_CH_SHADR(rw), DMA_TSR(rw), P_OMR(w), QSPI_BACONENTRY(w), QSPI_DATAENTRY(w), QSPI_ECON(w), QSPI_FLAGSCLEAR(w), QSPI_GLOBALCON(rw), QSPI_GLOBALCON1(w), QSPI_MC(w), QSPI_MCCON(w), QSPI_RXEXIT(r), QSPI_STATUS(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	

(table continues...)

Table 68 (continued) Specification for `CanTrcv_17_W9255_GetTrcvSystemData` API

Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.12 CanTrcv_17_W9255_ReadTrcvSilenceFlag

Table 69 Specification for `CanTrcv_17_W9255_ReadTrcvSilenceFlag` API

Syntax	`Std_ReturnType CanTrcv_17_W9255_ReadTrcvSilenceFlag ( const uint8 Transceiver, CanTrcv_17_W9255_TrcvFlagStateType * const FlagState )`	
Service ID	0x0d	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant	
Parameters (in)	Transceiver	Pointer to operation mode of the transceiver the API is applied to. This parameter has a valid range of 0-11.
Parameters (out)	FlagState	State of the silence flag
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK: If status of the silence flag is successfully read. E_NOT_OK: If status of the silence flag could not be read or a development error occurs.
Description	This API reads the status of the silence flag from the transceiver hardware. Since the configuration parameter CanTrcvHwPnSupport is always TRUE, this API is always available. The silence flag, if set, indicates that there is no communication on the CAN bus for a specified period of time (0.6 - 1.2 seconds). It helps to identify whether or not the TLE9255W hardware is in the Selective Sleep Sub-Mode.	
Source	AUTOSAR	
Error handling	CANTRCV_17_W9255_E_INVALID_TRANSCEIVER, CANTRCV_17_W9255_E_PARAM_POINTER	
Configuration dependencies	CanTrcvHwPnSupport	
User hints	-	

(table continues...)

Table 69 (continued) Specification for `CanTrcv_17_W9255_ReadTrcvSilenceFlag` API

SFR accessed	CPU_CORE_ID(r), DMA_CH_ADICR(rw), DMA_CH_CHCFGR(w), DMA_CH_CHCSR(w), DMA_CH_DADR(w), DMA_CH_RDCRCR(w), DMA_CH_SADR(w), DMA_CH_SDCRCR(w), DMA_CH_SHADR(rw), DMA_TSR(rw), P_OMR(w), QSPI_BACONENTRY(w), QSPI_DATAENTRY(w), QSPI_ECON(w), QSPI_FLAGSCLEAR(w), QSPI_GLOBALCON(rw), QSPI_GLOBALCON1(w), QSPI_MC(w), QSPI_MCCON(w), QSPI_RXEXIT(r), QSPI_STATUS(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.13 CanTrcv_17_W9255_ReadTrcvTimeoutFlag

Table 70 Specification for `CanTrcv_17_W9255_ReadTrcvTimeoutFlag` API

Syntax	`Std_ReturnType CanTrcv_17_W9255_ReadTrcvTimeoutFlag`
	`(`
	`  const uint8 Transceiver,`
	`  CanTrcv_17_W9255_TrcvFlagStateType * const FlagState`
	`)`	

Service ID	0x0b	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant	
Parameters (in)	Transceiver	CAN transceiver to which API call has to be applied. This parameter has a valid range of 0-11.
Parameters (out)	FlagState	State of the timeout flag
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK: If status of the timeout flag is successfully read. E_NOT_OK: If status of the timeout flag could not be read or a development error occurs.
Description	This API reads the status of the timeout flag from the transceiver hardware. Since the configuration parameter CanTrcvHwPnSupport is always TRUE, this API is always available. The timeout flag indicates whether or not the TLE9255W hardware has entered the Selective Sleep Sub-Mode at least once.	
Source	AUTOSAR	
Error handling	CANTRCV_17_W9255_E_INVALID_TRANSCEIVER, CANTRCV_17_W9255_E_PARAM_POINTER	
Configuration dependencies	CanTrcvHwPnSupport	
User hints	-	

(table continues...)

Table 70 (continued) Specification for `CanTrcv_17_W9255_ReadTrcvTimeoutFlag` API

SFR accessed	CPU_CORE_ID(r), DMA_CH_ADICR(rw), DMA_CH_CHCFGR(w), DMA_CH_CHCSR(w), DMA_CH_DADR(w), DMA_CH_RDCRCR(w), DMA_CH_SADR(w), DMA_CH_SDCRCR(w), DMA_CH_SHADR(rw), DMA_TSR(rw), P_OMR(w), QSPI_BACONENTRY(w), QSPI_DATAENTRY(w), QSPI_ECON(w), QSPI_FLAGSCLEAR(w), QSPI_GLOBALCON(rw), QSPI_GLOBALCON1(w), QSPI_MC(w), QSPI_MCCON(w), QSPI_RXEXIT(r), QSPI_STATUS(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.14 CanTrcv_17_W9255_SetPNActivationState

Table 71 Specification for `CanTrcv_17_W9255_SetPNActivationState` API

Syntax	`Std_ReturnType CanTrcv_17_W9255_SetPNActivationState ( const CanTrcv_17_W9255_PNActivationType ActivationState )`	
Service ID	0x0f	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant	
Parameters (in)	ActivationState	PN_ENABLED: PN wakeup functionality in CAN Transceiver shall be enabled. PN_DISABLED: PN wakeup functionality in CAN Transceiver shall be disabled.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK: If PN has been changed to the requested configuration. E_NOT_OK: If the PN configuration change has failed or if an invalid enum value is passed as parameter.
Description	This API enables/disables selective wake-up functionality of all those channels which have enabled PN in their configuration.	
Source	AUTOSAR	
Error handling	CANTRCV_17_W9255_E_UNINIT	
Configuration dependencies	-	
User hints	-	

(table continues...)

Table 71 (continued) Specification for `CanTrcv_17_W9255_SetPNActivationState` API

SFR accessed	CPU_CORE_ID(r), DMA_CH_ADICR(rw), DMA_CH_CHCFGR(w), DMA_CH_CHCSR(w), DMA_CH_DADR(w), DMA_CH_RDCRCR(w), DMA_CH_SADR(w), DMA_CH_SDCRCR(w), DMA_CH_SHADR(rw), DMA_TSR(rw), P_OMR(w), QSPI_BACONENTRY(w), QSPI_DATAENTRY(w), QSPI_ECON(w), QSPI_FLAGSCLEAR(w), QSPI_GLOBALCON(rw), QSPI_GLOBALCON1(w), QSPI_MC(w), QSPI_MCCON(w), QSPI_RXEXIT(r), QSPI_STATUS(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.4 Notifications and Callbacks

The CANTRCV_17_W9255 driver does not provide any notification or callbacks.

1.3.5 Scheduled functions

This section lists all the scheduled functions of the CanTrcv_17_W9255 driver.

1.3.5.1 CanTrcv_17_W9255_MainFunction

Table 72 Specification for `CanTrcv_17_W9255_MainFunction` API

Syntax	`void CanTrcv_17_W9255_MainFunction ( void )`	
Service ID	0x06	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant	
Parameters (in)	-	-
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	void	-
Description	This API scans all transceiver channels in Stand-by and Sleep modes for wake up events and sets a wake-up event flag to perform these events. *Note: The wake-up event flag is polled by CanTrcv_17_W9255_CheckWakeup API in polling mode.*	
Source	AUTOSAR	

(table continues...)

Table 72 (continued) Specification for `CanTrcv_17_W9255_MainFunction` API

Error handling	CANTRCV_17_W9255_E_UNINIT
Configuration dependencies	CanTrcvWakeUpSupport
User hints	-
SFR accessed	CPU_CORE_ID(r), DMA_CH_ADICR(rw), DMA_CH_CHCFGR(w), DMA_CH_CHCSR(w), DMA_CH_DADR(w), DMA_CH_RDCRCR(w), DMA_CH_SADR(w), DMA_CH_SDCRCR(w), DMA_CH_SHADR(rw), DMA_TSR(rw), P_OMR(w), QSPI_BACONENTRY(w), QSPI_DATAENTRY(w), QSPI_ECON(w), QSPI_FLAGSCLEAR(w), QSPI_GLOBALCON(rw), QSPI_GLOBALCON1(w), QSPI_MC(w), QSPI_MCCON(w), QSPI_RXEXIT(r), QSPI_STATUS(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar version 4.2.2.

1.3.5.2 CanTrcv_17_W9255_MainFunction

Table 73 Specification for `CanTrcv_17_W9255_MainFunction` API

Syntax	`void CanTrcv_17_W9255_MainFunction ( void )`	
Service ID	0x06	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant	
Parameters (in)	-	-
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	void	-
Description	This API scans all transceiver channels in Stand-by and Sleep modes for wake up events and sets a wake-up event flag to perform these events. *Note: The wake-up event flag is polled by CanTrcv_17_W9255_CheckWakeup API in polling mode.*	
Source	AUTOSAR	
Error handling	-	

(table continues...)

Table 73 (continued) Specification for `CanTrcv_17_W9255_MainFunction` API

Configuration dependencies	CanTrcvWakeUpSupport
User hints	-
SFR accessed	CPU_CORE_ID(r), DMA_CH_ADICR(rw), DMA_CH_CHCFGR(w), DMA_CH_CHCSR(w), DMA_CH_DADR(w), DMA_CH_RDCRCR(w), DMA_CH_SADR(w), DMA_CH_SDCRCR(w), DMA_CH_SHADR(rw), DMA_TSR(rw), P_OMR(w), QSPI_BACONENTRY(w), QSPI_DATAENTRY(w), QSPI_ECON(w), QSPI_FLAGSCLEAR(w), QSPI_GLOBALCON(rw), QSPI_GLOBALCON1(w), QSPI_MC(w), QSPI_MCCON(w), QSPI_RXEXIT(r), QSPI_STATUS(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar version 4.4.0.

1.3.6 Interrupt service routines

The CanTrcv_17_W9255 driver does not provide any interrupt handlers. *Note: CAN transceiver TLE9255W wake up interrupts are handled by ICU driver.*

1.3.7 Callout

The CanTrcv_17_W9255 driver does not provide any callout.

1.3.8 Errors Handling

This section describes the various error types reported by the CanTrcv_17_W9255 driver.

Error Name: Description	Source	Error ID (AS422)	Type (AS422)	Error ID (AS440)	Type (AS440)
CANTRCV_17_W9255_E_INIT_FAILED: Error is reported if initialization of the driver has failed. Since it is a Pre-compile module, the init function expects a NULL pointer to be passed as parameter. This DET is reported if a non-null pointer is passed as a parameter during init.	AUTOSAR	0x27	DET	0x27	DET
CANTRCV_17_W9255_E_INVALID_TRANSCEIVER: Error is reported if the API is called with invalid transceiver channel Id.	AUTOSAR	0x01	DET	0x01	DET

Error Name: Description	Source	Error ID (AS422)	Type (AS422)	Error ID (AS440)	Type (AS440)
CANTRCV_17_W9255_E_NO_TRCV_CONTROL: Error is reported when there is no/ incorrect communication to transceiver.	AUTOSAR	0x26	DET	0x26	RUNTIME
CANTRCV_17_W9255_E_PARAM_POINTER: Error is reported if API is invoked with null-pointer as a parameter.	AUTOSAR	0x02	DET	0x02	DET
CANTRCV_17_W9255_E_PARAM_TRCV_OPMODE: Error is reported when the API service is called with invalid parameter for OpMode.	AUTOSAR	0x24	DET	0x24	DET
CANTRCV_17_W9255_E_PARAM_TRCV_WAKEUP_MODE: Error is reported when the API service is called with invalid parameter for TrcvWakeupMode.	AUTOSAR	0x23	DET	0x23	DET
CANTRCV_17_W9255_E_TRCV_NOT_NORMAL: Error is reported when the CAN Transceiver is not in Normal mode or Stand-by mode and has got a request to transit to Stand-by mode.	AUTOSAR	0x22	DET	0x22	DET
CANTRCV_17_W9255_E_TRCV_NOT_STANDBY: Error is reported when the CAN Transceiver is not in Stand-by mode or Sleep mode and has got a request to transit to Sleep mode.	AUTOSAR	0x21	DET	0x21	DET
CANTRCV_17_W9255_E_UNINIT: Error is reported when the API service is used without initialization.	AUTOSAR	0x11	DET	0x11	DET

1.3.9 Deviations and limitations

This section describes the deviations and limitations of the CanTrcv_17_W9255 driver.

1.3.9.1 Deviations

This section describes the deviations of the CanTrcv_17_W9255 driver.

1.3.9.1.1 Software specification deviations

This section describes the deviations from software specification.

Table 74 Known deviations

Reference	Deviation
AUTOSAR CAN Transceiver requirement[SWS_CanTrcv_00090].	Since the hardware supports the wake-up functionality, the NOT_SUPPORTED mode is not applicable for the CAN transceiver driver.
AUTOSAR CAN Transceiver requirements[SWS_CanTrcv_00171], [SWS_CanTrcv_00172],[SWS_CanTrcv_00173].	Since the ICU driver does not depend on the Icu_EnableNotification and Icu_DisableNotification APIs for reporting wake-up, these interfaces are not used in the CAN transceiver driver.
AUTOSAR CAN Transceiver requirement[SWS_CanTrcv_00067].	In order to avoid compilation errors and repeated inclusion of files, AUTOSAR specified file structure is modified.
AUTOSAR CAN Transceiver requirements[SWS_CanTrcv_00228], [SWS_CanTrcv_00218].	Since TLE9255W hardware cannot detect bus failure, the CanTrcv_MainFunctionDiagnostics API and DEM error CANTRCV__E_BUS_ERROR is not available by the CAN transceiver driver.
For all requirements related to Runtime errors	Det_ReportRuntimeError is done through Mcal_Wrapper_Det_ReportRuntimeError interface. This is applicable for only AUTOSAR 4.4.0. All runtime related datatypes and modified interfaces inclusion shall be done via Mcal_Wrapper.h

1.3.9.1.2 AMDC Violations

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

Table 75 Violations reported by AMDC checker tool for A207

AMDC Rule	A207
Description	Maximum value of parameter 'CanTrcv/CanTrcvConfigSet/ CanTrcvChannel/CanTrcvMaxBaudrate' in VSMD (5000) may not be larger than maximum value defined in StMD (1000). [CanTrcv_17_W9255.bmd]

1.3.9.1.3 VSMD Violations

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

Table 76 Violations reported by VSMD checker tool for EB03

Rule ID:	EB03

(table continues...)

Table 76 (continued) Violations reported by VSMD checker tool for EB03

VSMD Node(s):	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess/CanTrcvSpiSequence/CanTrcvSpiAccessSynchronous /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvChannelEcucPartitionRef /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvDemEventParameterRefs /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvDemEventParameterRefs/CANTRCV_E_BUS_ERROR /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvIcuChannelRef /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPorWakeupSourceRef /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvSyserrWakeupSourceRef /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvWakeupSourceRef /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvMainFunctionDiagnosticsPeriod /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvMainFunctionPeriod /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvTimerType /AURIX2G_W9255/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 77 Violations reported by VSMD checker tool for EB09

Rule ID:	EB09
VSMD Node(s):	/AURIX2G_W9255/EcucDefs/CanTrcv

(table continues...)

Table 77 (continued) Violations reported by VSMD checker tool for EB09

Description:	EB specific rule to check consistency of parameter postBuildVariantUsed.
Additional Information:	

Table 78 Violations reported by VSMD checker tool for EcucSws_1007

Rule ID:	EcucSws_1007
VSMD Node(s):	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvMaxBaudrate
Description:	For Integer and Float Parameters the MIN values must be >= and the MAX values <= as in the StMD.
Additional Information:	

Table 79 Violations reported by VSMD checker tool for EcucSws_1014

Rule ID:	EcucSws_1014
VSMD Node(s):	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvGeneral /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvWakeUpSupport
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 80 Violations reported by VSMD checker tool for EcucSws_1035

Rule ID:	EcucSws_1035

(table continues...)

Table 80 (continued) Violations reported by VSMD checker tool for EcucSws_1035

VSMD Node(s):	/AURIX2G_W9255/EcucDefs/CanTrcv /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess/CanTrcvDioChannelAccess /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess/CanTrcvDioChannelAccess/CanTrcvDioSymNameRef /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess/CanTrcvDioChannelAccess/CanTrcvHardwareInterfaceName /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess/CanTrcvSpiSequence /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess/CanTrcvSpiSequence/CanTrcvSpiAccessSynchronous /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess/CanTrcvSpiSequence/CanTrcvSpiSequenceName /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvChannelEcucPartitionRef /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvChannelId /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvChannelUsed /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvControlsPowerSupply

(table continues...)

Table 80 (continued) Violations reported by VSMD checker tool for EcucSws_1035

	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvDemEventParameterRefs
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvDemEventParameterRefs/CANTRCV_E_BUS_ERROR
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvHwPnSupport
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvIcuChannelRef
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvInitState
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvMaxBaudrate
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvBaudRate
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvBusErrFlag
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnCanIdIsExtended
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnEnabled
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameCanId
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameCanIdMask
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameDataMaskSpec
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/

(table continues...)

Table 80 (continued) Violations reported by VSMD checker tool for EcucSws_1035

	CanTrcvPnFrameDataMaskSpec/ CanTrcvPnFrameDataMask
	/AURIX2G_W9255/EcucDefs/ CanTrcv/CanTrcvConfigSet/ CanTrcvChannel/CanTrcvPartialNetwork/ CanTrcvPnFrameDataMaskSpec/ CanTrcvPnFrameDataMaskIndex
	/AURIX2G_W9255/EcucDefs/CanTrcv/ CanTrcvConfigSet/CanTrcvChannel/ CanTrcvPartialNetwork/CanTrcvPnFrameDlc
	/AURIX2G_W9255/EcucDefs/CanTrcv/ CanTrcvConfigSet/CanTrcvChannel/ CanTrcvPartialNetwork/CanTrcvPowerOnFlag
	/AURIX2G_W9255/EcucDefs/CanTrcv/ CanTrcvConfigSet/CanTrcvChannel/ CanTrcvPorWakeupSourceRef
	/AURIX2G_W9255/EcucDefs/CanTrcv/ CanTrcvConfigSet/CanTrcvChannel/ CanTrcvSyserrWakeupSourceRef
	/AURIX2G_W9255/EcucDefs/CanTrcv/ CanTrcvConfigSet/CanTrcvChannel/ CanTrcvWakeupByBusUsed
	/AURIX2G_W9255/EcucDefs/CanTrcv/ CanTrcvConfigSet/CanTrcvChannel/ CanTrcvWakeupSourceRef
	/AURIX2G_W9255/EcucDefs/CanTrcv/ CanTrcvConfigSet/CanTrcvSPICommRetries
	/AURIX2G_W9255/EcucDefs/CanTrcv/ CanTrcvConfigSet/CanTrcvSPICommTimeout
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvGeneral
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvGeneral/ CanTrcvDevErrorDetect
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvGeneral/ CanTrcvEcucPartitionRef
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvGeneral/ CanTrcvGetVersionInfo
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvGeneral/ CanTrcvIndex
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvGeneral/ CanTrcvMainFunctionDiagnosticsPeriod
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvGeneral/ CanTrcvMainFunctionPeriod
	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvGeneral/ CanTrcvTimerType

(table continues...)

Table 80 (continued) Violations reported by VSMD checker tool for EcucSws_1035

	/AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvVersionInfoApi /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvWaitTime /AURIX2G_W9255/EcucDefs/CanTrcv/CanTrcvGeneral/CanTrcvWakeUpSupport
Description:	For Containers, Parameters and References elements UUID must be unique (also between StMD and VSMD).
Additional Information:	

Table 81 Violations reported by VSMD checker tool for EcucSws_2101

Rule ID:	EcucSws_2101
VSMD Node(s):	/AURIX2G_W9255/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 82 Violations reported by VSMD checker tool for EcucSws_6003

Rule ID:	EcucSws_6003
VSMD Node(s):	/AURIX2G_W9255/EcucDefs/CanTrcv
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 83 Violations reported by VSMD checker tool for TpsEcuc_06051_ASR41

Rule ID:	TpsEcuc_06051_ASR41
VSMD Node(s):	/AURIX2G_W9255/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

This section describes the limitations of the CanTrcv_17_W9255 driver.

Revision history

Table 84 Major changes since last version

Date	Version	Description
2023-06-19	5.0	Document is released.
2023-05-25	4.1	• ASIL level field changed to Safety level with description as "refer to release notes" for all APIs under 1.3.3 Functions - APIs and 1.3.5 Scheduled functions. • In 1.1.4 Integration hints section, the following points are modified - DEM module section has been removed. - Mcal_wrapper module section has been added. - Updated DET section to remove runtime error from the description. • Updated Figure 1 under 1.1.2 Hardware-software mapping, DEM Module is removed and Mcal_wrapper Module is added. • Updated section 1.1.3.1 C file structure to remove Dem.h and include Mcal_wrapper.h. • Updated the section 1.3.9.1.1: Software Specification Deviations for Autosar requirements. - Updated Reference from "SWS_CanTrcv_00084: Rte_Dem_Types.h" to "For all requirements related to Runtime errors". - Updated Description of "SWS_CanTrcv_00084: Rte_Dem_Types.h" to add Mcal_Wrapper Module Information.
2021-11-09	4.0	Document is released.
2021-11-08	3.1	Config variant attribute table information is removed and added this information in 'Configuration interfaces' section.
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.

(table continues...)

Table 84 (continued) Major changes since last version

2020-11-12	2.0	SFR access fields added for APIs (Since CanTrcv driver is external driver all SFR fields updated as None).
2020-08-13	1.0	Document is released.
2020-08-03	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 2023-06-19

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.