Elektrobit

CanTrcv_1_T03 documentation

Module release 1.3.9

Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen, Germany
Phone: +49 9131 7701 0
Fax: +49 9131 7701 6333
Email: info.automotive@elektrobit.com

Technical support

Europe	Japan	USA
Phone: +49 9131 7701 6060	Phone: +81 3 5577 6110	Phone: +1 888 346 3813

Support URL

https://www.elektrobit.com/support

Legal notice

Confidential and proprietary information

Table of Contents

1. Overview

Welcome to the CanTrcv_1_T03 product release notes and documentation.

This document provides:

- Chapter 2, “CanTrcv_1_T03 module release notes“: details of changes and new features in the current release
- Chapter 3, “CanTrcv user's guide“: concept information and configuration instructions
- Chapter 4, “CanTrcv_1_T03 module references“: configuration parameters and the application programming interface

2. CanTrcv_1_T03 module release notes

- AUTOSAR R4.0 Rev 3
- AUTOSAR SWS document version: 3.0.0
- Module version: 1.3.9.B331203
- Supplier: Elektrobit Automotive GmbH

2.1. Change log

This chapter lists the changes between different versions.

Module version 1.3.9

2020-05-27

- ASCCANTRCV-429 Fixed known issue: Missing infix of CANTRCV_WAKEUP_BY_POLLING and CANTRCV_WAKEUP_NOT_SUPPORTED

Module version 1.3.8

2018-10-26

- Internal module improvement. This module version update does not affect module functionality

Module version 1.3.7

2018-06-22

- Internal module improvement. This module version update does not affect module functionality

Module version 1.3.6

2018-05-25

- Internal module improvement. This module version update does not affect module functionality

Module version 1.3.5

2018-04-05

- Internal module improvement. This module version update does not affect module functionality

Module version 1.3.4

2018-03-20

- Internal module improvement. This module version update does not affect module functionality

Module version 1.3.3

2017-11-17

- Wakeup-By-Pin is independent of Wakeup-By-Bus
- Internal module improvement. This module version update does not affect module functionality

Module version 1.3.2

2017-09-22

- Internal module improvement. This module version update does not affect module functionality

Module version 1.3.1

2017-07-28

- Internal module improvement. This module version update does not affect module functionality
- Internal module improvement. This module version update does not affect module functionality

Module version 1.3.0

2017-05-05

- Usage of selective wakeup functionality on transceivers without partial networking

Module version 1.2.7

2017-03-31

- Internal module improvement. This module version update does not affect module functionality

Module version 1.2.6

2017-03-10

- Internal module improvement. This module version update does not affect module functionality

Module version 1.2.5

2017-02-03

- Internal module improvement. This module version update does not affect module functionality
- Internal module improvement. This module version update does not affect module functionality
- Internal module improvement. This module version update does not affect module functionality

Module version 1.2.4

2017-01-05

- Add support for asynchronous SPI handling

Module version 1.2.3

2016-12-02

- Added CAN-FD passive support, configurable via parameter `CanTrcvCanFdTolerance`

Module version 1.2.2

2016-02-05

- Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeaderFile`

Module version 1.2.1

2015-04-28

- ASCCANTRCV-288 Fixed known issue: CanTrcv generates wrong symbolic names for CanTrcv channels (namely CanTrcvConf_1_T03_<shortName of EcucContainerValue container>). According to [ecuc_sws_2108] however these symbolic names shall adhere to the following name pattern: CanTrcvConf_1_T03_CanTrcvChannel_<shortName of EcucContainerValue container>.
- Extended license checks for partial networking feature to cover new license string EB_CANTRCV_PARTIAL_NETWORK as well
- Adapted generated XGEN file for compatibility with ACG-7.3 Base plugin
- Modified memory class used in compiler abstraction of `CanTrcv_1_T03_SpiCommRead/WriteIB()` function from `SPI_VAR` to `AUTOMATIC` (since the used buffer is allocated on the stack)

Module version 1.2.0

2014-08-21

- Added usage of abstracted transceiver IDs from CanIf configuration when invoking CanIf callback APIs
- Added support for running parallel multiple variants of a CanTrcv plugin

Module version 1.1.2

2014-05-21

- ASCCANTRCV-259 Fixed known issue: Build error due to missing file CanTrcv_1_T03_PBcfg.c if code generation for CanTrcv is disabled and only post-build configuration is compiled
- ASCCANTRCV-261 Fixed known issue: CanTrcv_1_T03: Selective wakeup registers may not be initialized properly
- ASCCANTRCV-265 Fixed known issue: CanTrcv_1_T03: CanTrcv does not correctly disable/enable ICU interrupts
- ASCCANTRCV-262 Fixed known issue: CanTrcv_1_T03: CanTrcv does not store wakeup events during initialization if partial networking support is disabled
- ASCCANTRCV-264 Fixed known issue: CanTrcv_1_T03: `CanTrcv_GetBusWuReason()` always returns `E_NOT_OK` if wakeup mode is disabled

- ASCCANTRCV-264 Fixed known issue: CanTrcv_1_T03: `CanTrcv_GetBusWuReason()` always returns `E_NOT_OK` if wakeup mode is disabled
- ASCCANTRCV-263 Fixed known issue: CanTrcv_1_T03: `CanTrcv_GetBusWuReason()` may report `CANTRCV_WU_ERROR` even if no hardware error occurs
- Added license protection for feature CANTRCV_FEATURE_PARTIAL_NETWORKING

Module version 1.1.1

2014-02-28

- Implemented small source code improvements to increase module maintainability

Module version 1.1.0

2013-11-15

- Added support for relocatable Post-build configuration
- Improved MCG to generate XML code for Binary Code Generation
- Added ICU driver notification support
- Changed configuration of wakeup source references from mandatory to optional

Module version 1.0.1

2013-02-14

- Changed wakeup indication during de-initialization of partial network to match with CanSM
- ASCCANTRCV-154 Fixed known issue: CanTrcv fails transition to mode SLEEP for a specific wakeup configuration (CanTrcv_1_T03)
- ASCCANTRCV-157 Fixed known issue: Mode transition might fail if Power-on-Reset event is pending (CanTrcv_1_T03)

Module version 1.0.0

2012-11-16

- Initial AUTOSAR 4.0 revision 3

2.2. New features

- Added support for selective wakeup functionality on transceivers without partial networking. It should be possible to enable the selective wakeup support in CanTrcv without having to enable the partial networking support in CanIf, CanSM, ComM, Nm, and CanNm.

2.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

- Operation Mode API can be switched on and off

  Description:

  The configuration parameter `CanTrcvOperationModeApi` allows to disable the Operation Mode API `CanTrcv_SetOpMode` and `CanTrcv_GetOpMode` for a CAN transceiver which only task is to operate in normal state. In this case the CAN transceiver enters normal operation after call of function `CanTrcv_Init`.

- Wakeup API can be switched on and off

  Description:

  The configuration parameter `CanTrcvWakeupModeApi` allows to disable the Wakeup Mode API `CanTrcv_GetBusWuReason` and `CanTrcv_SetWakeupMode` and `CanTrcv_CheckWakeup` if wakeup is disabled by configuration.

- ICU driver notification support can be switched on and off

  Description:

  The ICU driver notification support API functions can be disabled by the configuration parameter `CanTrcvIcuDriverSupportApiEnable` if no ICU driver support is required.

- Parameter `CanTrcvSubModeCMC` added

  Description:

  The parameter `CanTrcvSubModeCMC` defines the behavior of the transceiver in the mode Normal. This configuration parameter enables to configure feature CMC of CAN control and Transceiver status registers. For details consult the data sheet of the transceiver device.

- Parameter `CanTrcvSilentDetection` added

  Description:

The parameter `CanTrcvSilentDetection` enables/disables CAN bus silence detection. This configuration parameter enables to configure feature CBSE of Transceiver event capture enable register. For details consult the data sheet of the transceiver device.

- Parameter `CanTrcvCanFailureDetection` added

  Description:

  The parameter `CanTrcvCanFailureDetection` enables/disables CAN failure detection. This configuration parameter enables to configure feature CFE of Transceiver event capture enable register. For details consult the data sheet of the transceiver device.

- Parameter `CanTrcvRisingEdgeDetection` added

  Description:

  The parameter `CanTrcvRisingEdgeDetection` enables/disables the detection of a rising-edge on `WAKE` pin. This configuration parameter enables to configure feature WPR of `WAKE` pin event status register. For details consult the data sheet of the transceiver device.

- Parameter `CanTrcvFallingEdgeDetection` added

  Description:

  The parameter `CanTrcvFallingEdgeDetection` enables/disables the detection of a falling-edge on `WAKE` pin. This configuration parameter enables to configure feature WPF of `WAKE` pin event status register. For details consult the data sheet of the transceiver device.

- Support for asynchronous SPI handling

  Description:

  This new behavior shall be configurable via the `CanTrcvSpiAccessSynchronous` switch available on each CanTrcv channel. The parameter `CanTrcvSpiAccessAsynchronousTimeout` has been added and it is enabled when `CanTrcvSpiAccessSynchronous` is FALSE. This parameter describes the maximum number of loop iterations used from polling the completion of the asynchronous transfer.

2.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

- Usage of `CanTrcv` channel cannot be switched off

  Description:

  The precompile time parameter `CanTrcvChannelUsed` has no influence on the behavior of the `CanTrcv` driver. A configured channel is automatically enabled and cannot be switched off.

Rationale:

Disabling parts of the configuration makes less sense for precompile or link-time configuration variants.

- The function `CanTrcv_MainFunction` does not exist

  Description:

  The module `CanTrcv` has no function `CanTrcv_MainFunction`. It is implemented as empty macro.

  Rationale:

  The specification of the function `CanTrcv_MainFunction` is a mismatch to the AUTOSAR SWS of the upper level modules `CanIf` and `EcuM`. The detection of wakeups works different as mentioned. The function `CanTrcv_CheckWakeup` is responsible for polling wakeup events.

  Requirements:

  CanTrcv013, CanTrcv112, CanTrcv128

- The function `CanTrcv_MainFunctionDiagnostics` does not exist

  Description:

  The API function `CanTrcv_MainFunctionDiagnostics` is not implemented.

  Rationale:

  `CanTrcv_MainFunctionDiagnostics` shall detect transceiver errors and perform error reporting. However DEM reportings are not supported by CanTrcv, thus the API function `CanTrcv_MainFunctionDiagnostics` is useless.

  Requirements:

  CanTrcv178, CanTrcv204, CanTrcv205, CanTrcv206, CanTrcv207, CanTrcv171_Conf

- The symbol *CANIF_TRCV_WU_POWERON* does not exist in `AUTOSAR_SWS_CANInterface.pdf`.

  Description:

  The `AUTOSAR_SWS_CANTransceiverDriver.pdf` defines the symbol *CANIF_TRCV_WU_POWERON* which is not included in the `AUTOSAR_SWS_CANInterface.pdf` specification.

  Rationale:

  The specification of *CANIF_TRCV_WU_POWERON* is a mismatch to the AUTOSAR SWS of the upper level module CanIf. The symbol *CANIF_TRCV_WU_POWERON* has been replaced by *CANTRCV_WU_POWER_ON*.

  Requirements:

CanTrcv183

- The symbol *CANIF_TRCV_WU_BY_SYSERR* does not exist in `AUTOSAR_SWS_CANInterface.pdf`

Description:

The `AUTOSAR_SWS_CANTransceiverDriver.pdf` defines the symbol *CANIF_TRCV_WU_BY_SYSERR* which is not included in the `AUTOSAR_SWS_CANInterface.pdf` specification.

Rationale:

The specification of *CANIF_TRCV_WU_BY_SYSERR* is a mismatch to the AUTOSAR SWS of the upper level module CanIf. The symbol *CANIF_TRCV_WU_BY_SYSERR* has been replaced by *CANTRCV_WU_BY_SYSERR*.

Requirements:

CanTrcv184

- The function `CanTrcv_GetTrcvSystemData` has no functionality

Description:

The function `CanTrcv_GetTrcvSystemData` only performs the specified DET checks. `CanTrcv_GetTrcvSystemData` returns always the return value `E_NOT_OK`.

Rationale:

This function is only a placeholder. Currently, there is no agreement for the transceiver hardware specification. Refer to the note in the `AUTOSAR_SWS_CANTransceiverDriver.pdf` in section 8.3.7. In addition, the output parameter `TrcvSysData` is defined as a pointer to a constant and is obsolete.

Requirements:

CanTrcv189, CanTrcv190

- No AUTOSAR Debug and Trace support

Description:

`CanTrcv` does not support AUTOSAR Debug and Trace.

Requirements:

CanTrcv151, CanTrcv152, CanTrcv153, CanTrcv154, CanTrcv155

- Only post-build configuration is supported

Description:

The CanTrcv module only supports configuration variant VARIANT-POST-BUILD. VARIANT-PRE-COMPILE (CanTrcv017) is not supported. The source file CanTrcv_1_T03_Cfg.c (CanTrcv062) is not provided.

Requirements:

CanTrcv017, CanTrcv062

- Ensure compatibility to CanSM state machine

  Description:

  If the CanTrcv accepts the request for checking the wakeup flag (API CanTrcv_CheckWakeFlag) it invokes the notification CanIf_CheckTrcvWakeFlagIndication in any case. This ensures compatibility with the CanSM as defined by the AUTOSAR SWS [CANSM460]. In addition the CanTrcv informs the EcuM about a wakeup via API EcuM_SetWakeupEvent.

  Requirements:

  CanTrcv224

- CanTrcv violates StMD to VSMD derivation rules

  Description:

  The following CanTrcv configuration parameters listed below violate the AUTOSAR StMD to VSMD derivation rules:

  - CanTrcvSPICommTimeout, CanTrcvSPICommRetries:

    These parameters have been removed from container CanTrcvGeneral.

    These parameters are of configuration class PreCompile instead of class PostBuild.

  - CanTrcvDioSymNameRef:

    This parameter is implemented as a reference to a DioChannel container, instead of being a choice reference.

  Rationale:

  - CanTrcvSPICommTimeout and CanTrcvSPICommRetries are duplicated in the AUTOSAR schema. The EB implementation uses the configuration parameters contained in the configuration container CanTrcvConfigSet.
  - The CanTrcv expects a DioChannel for the port pin configuration. Thus CanTrcvDioSymNameRef is limited to a reference to a single DioChannel.

  Requirements:

CanTrcv149_Conf, CanTrcv178_Conf, CanTrcv179_Conf

- Always initialize transceiver registers with `CanTrcv_Init()`

  Description:

  According to the `CanTrcv` SWS the function `CanTrcv_Init()` shall initialize transceiver registers only after detection of a power on or system error. Additionally `CanTrcv_SetopMode()` shall initialize the registers with the same constraints as `CanTrcv_Init()`. Contrary to this the current implementation always initializes the transceiver registers during `CanTrcv_Init()` but not in `CanTrcv_SetopMode()`.

  Rationale:

  In order to allow post-build functionality, changes in the configuration must take effect after calling `CanTrcv_Init()` with the respective configuration pointer. Reinitializing the register configuration during a call to `CanTrcv_SetopMode()` increases execution time, while this is not necessary assuming that this can always be achieved by calling `CanTrcv_Init()` if required by system design ,e.g. loss of battery power.

  Requirements:

  CanTrcv181, CanTrcv182, CanTrcv186, CanTrcv187

- `CanTrcv` does not report CANIF_TRCV_WU_POWERON as wakeup event

  Description:

  `CanTrcv` does not report `CANIF_TRCV_WU_POWERON` as wakeup event to the EcuM API function `EcuM_SetWakeupEvent()`. Instead `CANIF_TRCV_WU_POWERON` is reported by the API function `CanTrcv_1_T03_GetBusWuReason()` if no device specific wakeup has been detected, e.g. pin or network.

  Rationale:

  `EcuM_SetWakeupEvent()` does not accept parameters of the type `CANIF_TRCV_WU_POWERON`, instead they require a configurable value. However, it is not the job of the Can transceiver device to detect power up events.

  Requirements:

  CanTrcv183, CanTrcv170_Conf

2.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- The `CanTrcv` cannot enter sleep mode if no wakeup event is enabled

  Description:

  The `CanTrcv` hardware needs at least one regular wakeup event enabled to enter sleep mode. Therefore the following configuration constraint exists.

  If global wakeup support is enabled (configuration parameter `CanTrcvWakeUpSupport`) at least one of the following wakeup events must be enabled:

  - Wakeup-by-Bus (configuration parameter `CanTrcvWakeupByBusUsed`)
  - CAN bus silence detection (configuration parameter `CanTrcvSilentDetection`)
  - CAN failure detection (configuration parameter `CanTrcvCanFailureDetection`)
  - Rising-edge detection on `WAKE` pin (configuration parameter `CanTrcvRisingEdgeDetection`)
  - Falling-edge detection on `WAKE` pin (configuration parameter `CanTrcvFallingEdgeDetection`)

  If none of these wakeup events is enabled, the `CanTrcv` rejects a transition to sleep mode by returning `E_NOT_OK` for a request to `CanTrcv_SetOpMode()`.

  Rationale:

  To avoid deadlock, at least one regular wake-up event must be enabled and all event capture status bits must be cleared before the transceiver device switches to sleep mode. Otherwise the device switches to standby mode in response to a go-to-sleep command (MC = 001).

- The `CanTrcv` does not support Spi_DataType with 32bit width

  Description:

  The `Spi` provides the type `Spi_DataType` with implementation specific bit width allowing 8, 16 and 32bit. The `CanTrcv` restricts the possible range of `Spi_DataType` to 8 or 16bit. In case that `Spi_DataType` is generated as 32bit variable, correct opereration of `CanTrcv` is not guaranteed.

  Rationale:

  The Spi API `Spi_WriteIB()` and `Spi_ReadIB()` uses a pointer to `Spi_DataType`. The `CanTrcv` uses a buffer width of 16bit. Casting a pointer (either implicit or explicit) of object type uint16 to object type uint32 may lead to incompatible alignment.

- `SPI` asynchronous transfer in `CanTrcv`

  Description:

  If `CanTrcvSpiAccessSynchronous` is set to `False` asynchronous transfer mode will be used.

  Rationale:

To ensure asynchronous transmision the integration shall ensure that the `Spi_MainFunction_Handling()` function is executed in parallel to this polling loop by either mapping the `Spi_MainFunction_Handling()` function to a higher priority task than the task where any of the following functions are called: `CanTrcv_1_T03_GetOpMode()` `CanTrcv_1_T03_Init()` `CanTrcv_1_T03_SetOpMode()` `CanTrcv_1_T03_SetWakeupMode()` `CanTrcv_1_T03_ClearTrcvWufFlag()` `CanTrcv_1_T03_CheckWakeup()` `CanTrcv_1_T03_CheckWakeFlag()`

3. CanTrcv user's guide

3.1. Supported hardware

This CAN Transceiver Driver is designed, implemented and tested for the CAN transceiver device NXP TJA1145.

According to the manufacturer's CAN transceiver datasheets the following devices are compatible and thus also supported by this CAN Transceiver Driver:

- Atmel ATA6570

3.2. Software behavior

The API function `CanTrcv_ReadTrcvSilenceFlag` returns `E_NOT_OK` if configuration parameter `CanTrcvSilentDetection` is set to false.

The API function `CanTrcv_ReadTrcvTimeoutFlag` returns `E_NOT_OK` if configuration parameter `CanTrcvCanFailureDetection` is set to false.

Moreover, the API function `CanTrcv_SetWakeupMode` does not manage the DET `CANTRCV_E_NO_TRCV_CONTROL` in case of `TrcvWakeupMode = CANTRCV_WUMODE_DISABLE`. Indeed, no SPI requests are done.

3.3. Interface to hardware device

The CAN Transceiver Driver controls the hardware device via a SPI network. For minimal functionality at least a configuration for the following pins of the transceiver hardware must be provided:

- `SCK`

  SPI Clock Input. Synchronizes the transceiver device with the microcontroller.

- `SDI`

  SPI Data Input. Input sequence for managing the transceiver.

- `SDO`

SPI Data Output. Output data answering to the input sequence.

- `SCSN`

  SPI Chip Select input. Communication connection to the transceiver.

In addition the SPI module must be configure to manage one SPI sequence:

- `SpiDataWidth`

  The data unit transmitted in the sequence must be configured to 16 bits.

- `SpiTransferStart`

  The starting bit for the transmission must start with the MSB (register address).

3.4. SPI transfer modes

The CAN Transciever driver supports two modes of transfer:

- Synchronous

  Set `CanTrcvSpiAccessSynchronous` parameter to `True` to enable synchronous transfer.

  Additional parameters to configure if synchronous mode is used:

  - `CanTrcvSPICommRetries`

    Indicates the maximum number of communication retries in case of a failed SPI communication. This applies both to timed out communication and to errors/NACK in the response data.

- Asynchronous

  Set `CanTrcvSpiAccessSynchronous` parameter to `False` to enable asynchronous transfer.

  Additional parameters to configure if asynchronous mode is used:

  - `CanTrcvSpiAccessAsynchronousTimeout`

    Indicates the maximum number of loop iterations that are used to poll the completion of the asynchronous transfer. This parameter is related to the `SpiMainFunctionPeriod` parameter.

  - `CanTrcvSPICommRetries`

    Indicates the maximum number of communication retries in case of a failed SPI communication. This applies both to timed out communication and to errors/NACK in the response data.

  To ensure asynchronous transmission, the integration shall ensure that the `Spi_MainFunction_Handling()` function is executed in parallel to this polling loop by mapping the `Spi_MainFunction_Han-`

`dling()` function to a task that has a higher priority than the task where any of the following functions are called:

- `CanTrcv_1_T03_GetOpMode()`
- `CanTrcv_1_T03_Init()`
- `CanTrcv_1_T03_SetOpMode()`
- `CanTrcv_1_T03_SetWakeupMode()`
- `CanTrcv_1_T03_ClearTrcvWufFlag()`
- `CanTrcv_1_T03_CheckWakeup()`
- `CanTrcv_1_T03_CheckWakeFlag()`

4. CanTrcv_1_T03 module references

4.1. Configuration parameters

Containers included		
Container name	**Multiplicity**	**Description**
CanTrcvConfigSet	1..n	This is the multiple configuration set container for CAN Transceiver.
CanTrcvGeneral	1..1	Container gives CAN transceiver driver basic information.
VendorSpecific	1..1	Container contains CAN transceiver optimization parameter.
CommonPublishedInformation	1..1	**Label:** Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
PublishedInformation	1..1	**Label:** EB Published Information Additional published parameters not covered by CommonPublishedInformation container.

Parameters included	
Parameter name	**Multiplicity**
IMPLEMENTATION_CONFIG_VARIANT	1..1

Parameter Name	IMPLEMENTATION_CONFIG_VARIANT
Label	Config Variant
Multiplicity	1..1
Type	ENUMERATION
Default value	VariantPostBuild
Range	VariantPostBuild

4.1.1. CanTrcvConfigSet

Containers included		
Container name	**Multiplicity**	**Description**

Containers included		
CanTrcvChannel	1..n	Container gives CAN transceiver driver information about a single CAN.

Parameters included	
Parameter name	**Multiplicity**
CanTrcvSPICommRetries	1..1
CanTrcvSPICommTimeout	1..1

Parameter Name	**CanTrcvSPICommRetries**	
Description	Indicates the maximum number of communication retries in case of a failed SPI communication (applies both to timed out communication and to errors/NACK in the response data).	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255 >=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**CanTrcvSPICommTimeout**	
Description	Indicates the maximum time allowed to the CanTrcv for replying (either positively or negatively) to a SPI command. This parameter is unused, the CanTrcv supports only synchronous SPI access.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=100 >=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

4.1.2. CanTrcvChannel

Containers included		
Container name	**Multiplicity**	**Description**
VendorSpecific	1..1	Container contains CAN transceiver optimization parameter.
CanTrcvAccess	1..1	Container gives CanTrcv Driver information about access to a single CAN transceiver.
CanTrcvPartialNetwork	0..1	Container gives CAN transceiver driver information about the configuration of Partial Networking functionality.

Parameters included	
Parameter name	**Multiplicity**
CanTrcvChannelId	1..1
CanTrcvChannelUsed	1..1
CanTrcvControlsPowerSupply	1..1
CanTrcvHwPnSupport	1..1
CanTrcvInitState	1..1
CanTrcvWakeupByBusUsed	0..1
CanTrcvWakeupSourceRef	0..1
CanTrcvSyserrWakeupSourceRef	0..1
CanTrcvByBusWakeupSourceRef	0..1
CanTrcvByPinWakeupSourceRef	0..1

Parameter Name	CanTrcvChannelId
Description	Unique identifier of the CAN Transceiver Channel.
Multiplicity	1..1
Type	INTEGER
Range	<=255 >=0

Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**CanTrcvChannelUsed**
Description	Shall the related CAN transceiver channel be used? This parameter is not used.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**CanTrcvControlsPowerSupply**
Description	Is ECU power supply controlled by this transceiver? TRUE = Controlled by transceiver. FALSE = Not controlled by transceiver. This parameter is not used.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**CanTrcvHwPnSupport**
Description	Indicates whether the selective wake-up functionality is enabled or disabled in HW on a per-channel basis.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**CanTrcvInitState**
Description	State of CAN transceiver after call to CanTrcv_Init.
Multiplicity	1..1
Type	ENUMERATION
Default value	CANTRCV_OP_MODE_NORMAL

Range	CANTRCV_OP_MODE_NORMAL CANTRCV_OP_MODE_SLEEP CANTRCV_OP_MODE_STANDBY
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**CanTrcvWakeupByBusUsed**
Description	Is wake up by bus supported? If CAN transceiver hardware does not support wake up by bus value is always FALSE. If CAN transceiver hardware supports wake up by bus value is TRUE or FALSE depending whether it is used or not. TRUE = Is used. FALSE = Is not used
Multiplicity	0..1
Type	BOOLEAN
Default value	false
Configuration class	**PostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**CanTrcvWakeupSourceRef**
Description	Reference to a wakeup source in the EcuM configuration. **This configuration parameter is unused and replaced by the vendor specific parameters CanTrcvSyserrWakeupSourceRef, CanTrcvByBusWakeupSourceRef and CanTrcvByPinWakeupSourceRef.**
Multiplicity	0..1
Type	REFERENCE
Configuration class	**PostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**CanTrcvSyserrWakeupSourceRef**
Description	Reference to a wakeup source in the EcuM configuration for SYSERR detection. This parameter is active only if CanTrcvWakeUpSupport is active. The following events trigger a SYSERR: ▶ Partial networking frame detection error (always active if partial network enabled) ▶ CAN failure (if CanTrcvCanFailureDetection enabled) If all of the listed conditions are false, this reference can be disabled.

	Disabling this parameter does not prevent wakeup, it only prevents EcuM notification (except all listed conditions are also disabled).
Multiplicity	0..1
Type	REFERENCE
Configuration class	**PostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**CanTrcvByBusWakeupSourceRef**
Description	Reference to a wakeup source in the EcuM configuration for a by bus wake-up detection. This parameter is only active if CanTrcvWakeupByBusUsed is active and true. The following events trigger a by-bus wake-up: ▶ Wakeup on CAN bus ▶ Internal ECU wakeup via request to NORMAL mode Disabling this parameter does not prevent wakeup, it only prevents EcuM notification.
Multiplicity	0..1
Type	REFERENCE
Configuration class	**PostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**CanTrcvByPinWakeupSourceRef**
Description	Reference to a wakeup source in the EcuM configuration for a by PIN wake-up detection. This parameter is only active if CanTrcvWakeUpSupport is active. The following events trigger a wakeup-by-pin: ▶ Rising edge on WAKE pin (if CanTrcvRisingEdgeDetection enabled) ▶ Falling edge on WAKE pin (if CanTrcvFallingEdgeDetection enabled) If all of the listed conditions are false, this reference can be disabled. Disabling this parameter does not prevent wakeup, it only prevents EcuM notification (except all listed conditions are also disabled).
Multiplicity	0..1
Type	REFERENCE
Configuration class	**PostBuild:** VariantPostBuild

Origin	AUTOSAR_ECUC

4.1.3. VendorSpecific

Containers included		
Container name	**Multiplicity**	**Description**
CanTrcvTrcvEventRegister	1..1	Contains setting to enable or disable various transceiver event notifications.
CanTrcvSystEventRegister	1..1	Contains setting to enable or disable various system event notifications.
CanTrcvWakeEventRegister	1..1	Sets behavior for the transceiver wake-up pins.

Parameters included	
Parameter name	**Multiplicity**
CanTrcvIcuChannelRef	0..1
CanTrcvSubModeCMC	1..1
CanTrcvPnFramePNDM	1..1

Parameter Name	**CanTrcvIcuChannelRef**	
Description	Reference to the IcuChannel to enable/disable the interrupts for wakeups.	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	**PostBuild:**	VariantPostBuild

Parameter Name	**CanTrcvSubModeCMC**
Description	Select the CAN transceiver operating mode. This mode is only available in Normal mode. The state can be set into Offline mode Active mode Listen-only mode This parameter corresponds to the CMC bit of the TJA1145 hardware
Multiplicity	1..1
Type	ENUMERATION
Default value	Active
Range	Active
	Offline
	Listen_Only

Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**CanTrcvPnFramePNDM**	
Description	Indicates if the data and DLC have to be checked when Partial Networking is activated. This parameter has no effect, if PN is disabled.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

4.1.4. CanTrcvTrcvEventRegister

Parameters included	
Parameter name	**Multiplicity**
CanTrcvSilentDetection	1..1
CanTrcvCanFailureDetection	1..1

Parameter Name	**CanTrcvSilentDetection**	
Description	<para> Switches CAN bus silence detection on or off. </para> <para> In case the silent detection is disabled API service CanTrcv_ReadTrcvSilenceFlag can not obtain the silence status. </para>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**CanTrcvCanFailureDetection**
Description	<para> Switches CAN failure detection on or off. If disabled, a CAN failure does not result in a wakeup of the transceiver device. </para> <para> In case CAN failure detection is enabled the CanTrcv sets the SYSERR flag if the TXD dominant time-out time is exceeded. </para>

Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

4.1.5. CanTrcvSystEventRegister

4.1.6. CanTrcvWakeEventRegister

Parameters included	
Parameter name	**Multiplicity**
CanTrcvRisingEdgeDetection	1..1
CanTrcvFallingEdgeDetection	1..1

Parameter Name	**CanTrcvRisingEdgeDetection**	
Description	<para> Switches rising-edge detection on WAKE pin on or off. If disabled, a rising-edge on the WAKE pin does not result in a wakeup of the transceiver device. </para> <para> In case of a rising edge the CanTrcv reports a wake-up. </para>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**CanTrcvFallingEdgeDetection**
Description	<para> Switches falling-edge detection on WAKE pin on or off. If disabled, a falling-edge on the WAKE pin does not result in a wakeup of the transceiver device. </para> <para> In case of a falling edge the CanTrcv reports a wake-up. </para>
Multiplicity	1..1

Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

4.1.7. CanTrcvAccess

Containers included		
Container name	**Multiplicity**	**Description**
CanTrcvDioAccess	1..1	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 a CAN transceiver hardware has no Dio interface, there is no instance of this container.
CanTrcvSpiSequence	1..1	Container gives CAN transceiver driver information about one SPI se-quence. One SPI sequence used by CAN transceiver driver is in exclusive use for it. No other driver is allowed to access this sequence. CAN trans-ceiver driver may use one sequence to access n CAN transceiver hard-wares chips of the same type or n sequences are used to access one sin-gle CAN transceiver hardware chip. If a CAN transceiver hardware has no SPI interface, there is no instance of this container.

4.1.8. CanTrcvDioAccess

Containers included		
Container name	**Multiplicity**	**Description**
CanTrcvDioChannelAccess	1..n	Container gives DIO channel access by single Can transceiver channel.

4.1.9. CanTrcvDioChannelAccess

Parameters included	
Parameter name	**Multiplicity**

Parameters included	
CanTrcvHardwareInterfaceName	1..1
CanTrcvDioSymNameRef	1..1

Parameter Name	**CanTrcvHardwareInterfaceName**	
Description	CAN transceiver hardware interface name. It is typically the name of a pin. From a Dio point of view it is either a port, a single channel or a channel group. Depending on this fact either. CANTRCV_DIO_PORT_SYMBOLIC_NAME or CANTRCV_DIO_CHANNEL_SYMBOLIC_NAME or CANTRCV_DIO_CHANNEL_GROUP_SYMBOLIC_NAME shall reference a Dio configuration. The CAN transceiver driver implementation description shall list up this name for the appropriate CAN transceiver hardware.	
Multiplicity	1..1	
Type	STRING	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**CanTrcvDioSymNameRef**	
Description	Choice Reference to a DIO Port, DIO Channel or DIO Channel Group. This reference replaces the CANTRCV_DIO_PORT_SYM_NAME, CANTRCV_DIO_CHANNEL_SYM_NAME and CANTRCV_DIO_GROUP_SYM_NAME references in the Can Trcv SWS.	
Multiplicity	1..1	
Type	CHOICE-REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

4.1.10. CanTrcvSpiSequence

Parameters included	
Parameter name	**Multiplicity**
CanTrcvSpiAccessSynchronous	0..1

Parameters included	
CanTrcvSpiSequenceName	1..1

Parameter 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.	
Multiplicity	0..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**PostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	CanTrcvSpiSequenceName	
Description	Reference to a Spi sequence configuration container.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	**PostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

4.1.11. CanTrcvPartialNetwork

Containers included		
Container name	**Multiplicity**	**Description**
CanTrcvPnFrameDataMaskSpec	0..8	Defines data payload mask to be used on the received payload in order to determine if the transceiver must be woken up by the received Wake-up Frame (WUF).

Parameters included	
Parameter name	**Multiplicity**
CanTrcvBaudRate	1..1
CanTrcvBusErrFlag	1..1
CanTrcvPnCanIdIsExtended	1..1

Parameters included	
CanTrcvPnEnabled	1..1
CanTrcvPnFrameCanId	1..1
CanTrcvPnFrameCanIdMask	1..1
CanTrcvPnFrameDlc	1..1
CanTrcvPowerOnFlag	1..1

Parameter Name	CanTrcvBaudRate	
Description	Indicates the CAN Bus communication baud rate in kbps.	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	CanTrcvBusErrFlag	
Description	**This parameter is not used. Detecting of bus failures is currently not implemented.** 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. This parameter is not used.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	CanTrcvPnCanIdIsExtended	
Description	Indicates whether extended or standard ID is used.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild

Origin	AUTOSAR_ECUC

Parameter Name	**CanTrcvPnEnabled**
Description	Enables/disables the call to CanIf_ConfirmPNAvailabity() function. This parameter needs partial network support enabled in the upper layers.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**CanTrcvPnFrameCanId**
Description	CAN ID of the Wake-up Frame (WUF).
Multiplicity	1..1
Type	INTEGER
Range	<=4294967295 >=0
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**CanTrcvPnFrameCanIdMask**
Description	ID Mask for the selective activation of the transceiver. It is used to enableFrame Wake-up (WUF) on a group of IDs.
Multiplicity	1..1
Type	INTEGER
Range	<=4294967295 >=0
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**CanTrcvPnFrameDlc**
Description	Data Length of the Wake-up Frame (WUF).
Multiplicity	1..1
Type	INTEGER

Range	<=8 >=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**CanTrcvPowerOnFlag**	
Description	**This parameter is not used. Power on flag currently not checked on startup.** 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.	
Multiplicity	1..1	
Type	BOOLEAN	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

4.1.12. CanTrcvPnFrameDataMaskSpec

Parameters included	
Parameter name	**Multiplicity**
CanTrcvPnFrameDataMask	1..1
CanTrcvPnFrameDataMaskIndex	1..1

Parameter Name	**CanTrcvPnFrameDataMask**
Description	Defines the n byte (Byte0 = LSB) of the data payload mask to be used on the received payload in order to determine if the transceiver must be woken up by the received Wake-up Frame (WUF).
Multiplicity	1..1
Type	INTEGER
Range	<=255 >=0

Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**CanTrcvPnFrameDataMaskIndex**	
Description	holds the position n in frame of the mask-part	
Multiplicity	1..1	
Type	INTEGER	
Range	<=7 >=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

4.1.13. CanTrcvGeneral

Parameters included	
Parameter name	**Multiplicity**
CanTrcvDevErrorDetect	1..1
CanTrcvGetVersionInfo	1..1
CanTrcvWaitCount	1..1
CanTrcvWakeUpSupport	1..1

Parameter Name	**CanTrcvDevErrorDetect**	
Description	Switches development error detection and notification on and off.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**CanTrcvGetVersionInfo**
Description	Switches version information API on and off. If switched off, function need not be present in compiled code
Multiplicity	1..1

Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**CanTrcvWaitCount**
Description	Indicates the number of wait states to change the transceiver operation mode. Transceiver hardware may need wait states for some transitions.
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<=4294967295 >=0
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**CanTrcvWakeUpSupport**
Description	Informs whether wake up is supported by polling or not supported. In case no wake up is supported by the hardware,setting has to be NOT_SUPPORTED. Only in the case of wake up supported by polling, function CanTrcv_MainFunction has to be present and to be invoked by the scheduler.
Multiplicity	1..1
Type	ENUMERATION
Default value	CANTRCV_WAKEUP_BY_POLLING
Range	CANTRCV_WAKEUP_BY_POLLING CANTRCV_WAKEUP_NOT_SUPPORTED
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

4.1.14. VendorSpecific

Parameters included	
Parameter name	**Multiplicity**

Parameters included	
CanTrcvOperationModeApiEnable	1..1
CanTrcvWakeupModeApiEnable	1..1
CanTrcvPnSupport	1..1
CanTrcvCanFdTolerance	1..1
CanTrcvMaxChannelNumber	1..1
CanTrcvRelocatablePbcfgEnable	1..1
CanTrcvPnCanIfCallbackSupport	1..1
CanTrcvIcuDriverSupportApiEnable	1..1
CanTrcvSpiAccessAsynchronousTimeout	0..1

Parameter Name	CanTrcvOperationModeApiEnable
Description	Switches `CanTrcv_SetOpMode` and `CanTrcv_GetOpMode` API on and off. If set to `false` the Can transceiver is always set to mode 'CANTRCV_OP_MODE_NORMAL' after initalization. ▶ `true`: Enables operation mode API. ▶ `false`: Disables operation mode API. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	CanTrcvWakeupModeApiEnable

Description	Switches `CanTrcv_SetWakeupMode` and `CanTrcv_GetWuReason` API on and off. The API can only be disabled if parameter CanTrcvWakeUpSupport is set to `CANTRCV_WAKEUP_NOT_SUPPORTED`. ▶ `true`: Enables Wakeup mode API. ▶ `false`: Disables Wakeup mode API. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**CanTrcvPnSupport**
Description	Enables the support for the selective wake-up function (partial networking). ▶ `true`: Enables selective wake-up. ▶ `false`: Disables selective wake-up. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code. ▶ **ROM reduction (config):** Disabling this parameter reduces the ROM consumption of the module configuration. ▶ **Execution time reduction (code):** Disabling this parameter reduces the execution time of the module code. **This configuration parameter replaces CanTrcvHwPnSupport.** Please note that this parameter is enabled only if a valid license for the license feature CANTRCV_FEATURE_PARTIAL_NETWORKING is installed.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild

Origin	Elektrobit Automotive GmbH

Parameter Name	**CanTrcvCanFdTolerance**	
Description	Controls the tolerance of CAN-FD frames in combination with partial networking (passive support). ▶ `true`: CFDC bit set ▶ `false`: CFDC bit **not** set This parameter corresponds to the CFDC bit of the TJA1145's control register.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**CanTrcvMaxChannelNumber**	
Description	This parameter defines the size of the reserved RAM during link-time to hold the runtime data of the CanTrcv channel configuration. This parameter has impact on the size of the PostBuild RAM. **Optimization Effect:** ▶ **RAM reduction:** Selecting a small value for this parameter reduces the RAM consumption of the module.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**CanTrcvRelocatablePbcfgEnable**
Description	Enables/disable support for relocatable postbuild configuration. ▶ True: Postbuild configuration relocatable in memory. ▶ False: Postbuild configuration not relocatable in memory.
Multiplicity	1..1
Type	BOOLEAN

Default value	true
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**CanTrcvPnCanIfCallbackSupport**
Description	Switches on/off the call of CanIf_ConfirmPNAvailabity(), CanIf_ClearTrcvWufFlagIndication(), and CanIf_CheckTrcvWakeFlagIndication() functions ▶ True: Enables the call of APIs. ▶ False: Disables the call of APIs. **Important:** This parameter shall be set to true only if PN is enabled in CanIf.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**CanTrcvIcuDriverSupportApiEnable**
Description	Switches ICU driver API on and off. The API can only be disabled if all channels do not support ICU driver. ▶ `true`: Enables ICU driver API. ▶ `false`: Disables ICU driver API. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**CanTrcvSpiAccessAsynchronousTimeout**

Description	Indicates the maximum number of loop iterations used from polling the completion of the asynchronous transfer. This parameter is related to the SpiMainFunctionPeriod parameter. See limitations chapter.
Multiplicity	0..1
Type	INTEGER
Default value	1
Range	<=255 >=1
Configuration class	**PreCompile:** VariantPostBuild
Origin	AUTOSAR_ECUC

4.1.15. CommonPublishedInformation

Parameters included	
Parameter name	**Multiplicity**
ArMajorVersion	1..1
ArMinorVersion	1..1
ArPatchVersion	1..1
SwMajorVersion	1..1
SwMinorVersion	1..1
SwPatchVersion	1..1
ModuleId	1..1
VendorId	1..1
VendorApiInfix	0..1
Release	1..1

Parameter Name	**ArMajorVersion**
Label	AUTOSAR Major Version
Description	Major version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	3

Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**ArMinorVersion**
Label	AUTOSAR Minor Version
Description	Minor version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	0
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**ArPatchVersion**
Label	AUTOSAR Patch Version
Description	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	0
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**SwMajorVersion**
Label	Software Major Version
Description	Major version number of the vendor specific implementation of the module.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**SwMinorVersion**
Label	Software Minor Version

Description	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	3
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**SwPatchVersion**
Label	Software Patch Version
Description	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	9
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**ModuleId**
Label	Numeric Module ID
Description	Module ID of this module from Module List
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	70
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**VendorId**
Label	Vendor ID
Description	Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1

Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**VendorApiInfix**
Label	Vendor API Infix
Multiplicity	0..1
Type	STRING_LABEL
Default value	T03
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**Release**
Label	Release Information
Multiplicity	1..1
Type	STRING_LABEL
Default value	
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

4.1.16. PublishedInformation

Parameters included	
Parameter name	**Multiplicity**
PbcfgMSupport	1..1

Parameter Name	**PbcfgMSupport**
Label	PbcfgM support
Description	Specifies whether or not the CanTrcv_1_T03 can use the PbcfgM module for post-build support.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**PublishedInformation:**

Origin	Elektrobit Automotive GmbH

4.2. Application programming interface (API)

4.2.1. Macro constants

4.2.1.1. CANTRCV_1_T03_CHECKWAKEFLAG_ID

Purpose	CanTrcv_1_T03_CheckWakeFlag () service ID.
Value	0x0EU

4.2.1.2. CANTRCV_1_T03_CHECKWAKEUP_ID

Purpose	CanTrcv_1_T03_CheckWakeup() service ID.
Value	0x07U

4.2.1.3. CANTRCV_1_T03_CLEARTRCVTIMEOUTFLAG_ID

Purpose	CanTrcv_1_T03_CanTrcv_ClearTrcvTimeoutFlag () service ID.
Value	0x0CU

4.2.1.4. CANTRCV_1_T03_CLEARTRCVWUFFLAG_ID

Purpose	CanTrcv_1_T03_CanTrcv_ClearTrcvWufFlag () service ID.
Value	0x0AU

4.2.1.5. CANTRCV_1_T03_E_BAUDRATE_NOT_SUPPORTED

Purpose	API service called with invalid parameter for Baudrate DET error code.

Value	0x25U

4.2.1.6. CANTRCV_1_T03_E_INVALID_TRANSCEIVER

Purpose	API called with wrong parameter for transceiver DET error code.
Value	0x01U

4.2.1.7. CANTRCV_1_T03_E_NO_TRCV_CONTROL

Purpose	API service called with invalid parameter for no/incorrect communication to Transceiver DET error code.
Value	0x26U

4.2.1.8. CANTRCV_1_T03_E_PARAM_POINTER

Purpose	API called with with null pointer parameter DET error code.
Value	0x02U

4.2.1.9. CANTRCV_1_T03_E_PARAM_TRCV_OPMODE

Purpose	API service called with invalid parameter for TrcvOperationMode DET error code.
Value	0x24U

4.2.1.10. CANTRCV_1_T03_E_PARAM_TRCV_WAKEUP_MODE

Purpose	API service called with invalid parameter for TrcvWakeupMode DET error code.
Value	0x23U

4.2.1.11. CANTRCV_1_T03_E_TRCV_NOT_NORMAL

Purpose	API service called in wrong transceiver operation mode (2) DET error code.
Value	0x22U

4.2.1.12. CANTRCV_1_T03_E_TRCV_NOT_STANDBY

Purpose	API service called in wrong transceiver operation mode (1) DET error code.
Value	0x21U

4.2.1.13. CANTRCV_1_T03_E_UNINIT

Purpose	API service used without initialization DET error code.
Value	0x11U

4.2.1.14. CANTRCV_1_T03_GETBUSWUREASON_ID

Purpose	CanTrcv_1_T03_GetBusWuReason() service ID.
Value	0x03U

4.2.1.15. CANTRCV_1_T03_GETOPMODE_ID

Purpose	CanTrcv_1_T03_GetOpMode() service ID.
Value	0x02U

4.2.1.16. CANTRCV_1_T03_GETTRCVSYSTEMDATA_ID

Purpose	CanTrcv_1_T03_GetTrcvSystemData() service ID.
Value	0x09U

4.2.1.17. CANTRCV_1_T03_GETVERSIONINFO_ID

Purpose	CanTrcv_1_T03_GetVersionInfo() service ID.
Value	0x04U

4.2.1.18. CANTRCV_1_T03_INIT_ID

Purpose	CanTrcv_1_T03_Init() service ID.

Value	0x00U

4.2.1.19. CANTRCV_1_T03_ISVALIDCONFIG_SERVICE_ID

Purpose	CanTrcv_1_T03_IsValidConfig() service ID.
Value	0x60U
Description	EB_Automotive_C_Source_File

4.2.1.20. CANTRCV_1_T03_MAINFUNCTIONDIAGNOSTICS_ID

Purpose	CanTrcv_1_T03_MainFunctionDiagnostics () service ID.
Value	0x08U

4.2.1.21. CANTRCV_1_T03_MAINFUNCTION_ID

Purpose	CanTrcv_1_T03_MainFunction() service ID.
Value	0x06U

4.2.1.22. CANTRCV_1_T03_READTRCVSILENCEFLAG_ID

Purpose	CanTrcv_1_T03_CanTrcv_ReadTrcvSilenceFlag () service ID.
Value	0x0DU

4.2.1.23. CANTRCV_1_T03_READTRCVTIMEOUTFLAG_ID

Purpose	CanTrcv_1_T03_CanTrcv_ReadTrcvTimeoutFlag () service ID.
Value	0x0BU

4.2.1.24. CANTRCV_1_T03_SETOPMODE_ID

Purpose	CanTrcv_1_T03_SetOpMode() service ID.
Value	0x01U

4.2.1.25. CANTRCV_1_T03_SETPNACTIVATIONSTATE_ID

Purpose	CanTrcv_1_T03_SetPNActivationState () service ID.
Value	0x0FU

4.2.1.26. CANTRCV_1_T03_SETWAKEUPMODE_ID

Purpose	CanTrcv_1_T03_SetWakeupMode() service ID.
Value	0x05U

4.2.1.27. CanTrcv_1_T03_MainFunction

Purpose	CAN transceiver main function.
Value	
Description	Main function of Can Tranceiver. Note: In this implementation the MainFunction has no functionality (it is possible to satisfy the AUTOSAR 4.0 spec without MainFunction). Therefore and for optimization reason the function is defined as empty macro.

4.2.1.28. CanTrcv_1_T03_MainFunctionDiagnostics

Purpose	Main function Diagnostics.
Value	
Description	Service to read the transceiver diagnostic status periodically and sets products/development accordingly

4.2.2. Functions

4.2.2.1. CanTrcv_1_T03_CheckWakeFlag

Purpose	Check wake flag.

Synopsis	`Std_ReturnType` **`CanTrcv_1_T03_CheckWakeFlag`** `( uint8 Transceiver );`	
Service ID	0x0E	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	`Transceiver`	Index of the transceiver
Return Value	Result of operation	
	`E_OK`	check of the WU flag OK
	`E_NOT_OK`	check WU flag not accepted
Description	Check the status of the wakeup flag from the transceiver hardware	

4.2.2.2. CanTrcv_1_T03_CheckWakeup

Purpose	CAN transceiver check for wakeup function.	
Synopsis	`Std_ReturnType` **`CanTrcv_1_T03_CheckWakeup`** `( uint8 Transceiver );`	
Service ID	0x07	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	`Transceiver`	Index of the transceiver
Return Value	Result of operation	
	`E_OK`	valid interrupt detected
	`E_NOT_OK`	no interrupt detected
Description	This function checks for wake-up events	

4.2.2.3. CanTrcv_1_T03_ClearTrcvTimeoutFlag

Purpose	Clear timeout flag of transceiver.
Synopsis	`Std_ReturnType` **`CanTrcv_1_T03_ClearTrcvTimeoutFlag`** `( uint8 Transceiver );`
Service ID	0x0C
Sync/Async	Synchronous
Reentrancy	Non-Reentrant

Parameters (in)	`Transceiver`	Index of the transceiver
Return Value	Result of operation	
	`E_OK`	flag CF clear
	`E_NOT_OK`	error during clearing CFS flag
Description	This function will clear the flag CF in charge of managing CAN failure anor dominant clamped Tx	

4.2.2.4. CanTrcv_1_T03_ClearTrcvWufFlag

Purpose	CAN transceiver clear WUF flag.	
Synopsis	`Std_ReturnType` **`CanTrcv_1_T03_ClearTrcvWufFlag`** `( uint8 Transceiver );`	
Service ID	0x0A	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	`Transceiver`	Index of the transceiver
Return Value	Result of operation	
	`E_OK`	flag event clear ok
	`E_NOT_OK`	flag even clear not done
Description	This function clear all WU flags and also global register	

4.2.2.5. CanTrcv_1_T03_GetBusWuReason

Purpose	CAN transceiver get wakeup reason function.	
Synopsis	`Std_ReturnType` **`CanTrcv_1_T03_GetBusWuReason`** `( uint8 Transceiver , CanTrcv_TrcvWakeupReasonType * reason );`	
Service ID	0x03	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	`Transceiver`	Index of the transceiver
Parameters (out)	`reason`	Pointer to wake up reason of the bus
Return Value	Result of operation	

	E_OK	wake up reason was detected
	E_NOT_OK	wake up reason was not detected
Description	This function gets the wakeup reason for the channel CanNetwork	

4.2.2.6. CanTrcv_1_T03_GetOpMode

Purpose	CAN transceiver get operation mode function.	
Synopsis	`Std_ReturnType` **`CanTrcv_1_T03_GetOpMode`** `( uint8 Transceiver , CanTrcv_TrcvModeType * OpMode );`	
Service ID	0x02	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	Transceiver	Index of the transceiver
Parameters (out)	OpMode	Pointer to operation mode of the bus
Return Value	Result of operation	
	E_OK	operation mode was detected
	E_NOT_OK	operation mode was not detected
Description	This function gets the mode of the transceiver	

4.2.2.7. CanTrcv_1_T03_GetTrcvSystemData

Purpose	CAN transceiver coniguration/status data function.	
Synopsis	`Std_ReturnType` **`CanTrcv_1_T03_GetTrcvSystemData`** `( uint8 Transceiver , const uint32 * TrcvSysData );`	
Service ID	0x09	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	Transceiver	Index of the transceiver TrcvSysData Configuration/status of the transceiver
Return Value	Result of operation	
	E_OK	Trcv reading status
	E_NOT_OK	Wrong reading, data not available

Description	This function checks the transceive configuration/status of registers: Mode control register (address 01h) CAN control register (address 20h) Transceiver status register (address 22h) Transceiver event capture enable register (address 23h)

4.2.2.8. CanTrcv_1_T03_GetVersionInfo

Purpose	CAN transceiver get version info function.	
Synopsis	`void CanTrcv_1_T03_GetVersionInfo ( Std_VersionInfoType * versioninfo );`	
Service ID	0x04	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (out)	`versioninfo`	Pointer to version information of this module
Description	This function gets the version of the module	

4.2.2.9. CanTrcv_1_T03_Init

Purpose	CAN transceiver initialization function.
Synopsis	`void CanTrcv_1_T03_Init ( const CanTrcv_1_T03_ConfigType * ConfigPtr );`
Service ID	0x00
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Description	This function initializes the CAN transceiver module.

4.2.2.10. CanTrcv_1_T03_IsValidConfig

Purpose	Validate configuration.
Synopsis	`Std_ReturnType CanTrcv_1_T03_IsValidConfig ( const void * voidConfigPtr );`
Service ID	0x60

Sync/Async	Synchronous
Reentrancy	Reentrant
Return Value	E_OK if the given module configurations is valid otherwise E_NOT_OK.
Description	Checks if the post build configuration fits to the link time configuration part.

4.2.2.11. CanTrcv_1_T03_ReadTrcvSilenceFlag

Purpose	Read slicence flag.	
Synopsis	`Std_ReturnType` **`CanTrcv_1_T03_ReadTrcvSilenceFlag`** `( uint8 Transceiver , CanTrcv_TrcvFlagStateType * FlagState );`	
Service ID	0x0D	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	`Transceiver`	Index of the transceiver FlagState Set the status of the network
Return Value	Result of operation	
	`E_OK`	flags read ok
	`E_NOT_OK`	issue for reading flags
Description	This function read the flag CBS and CBSS to check if activities occur on the network	

4.2.2.12. CanTrcv_1_T03_ReadTrcvTimeoutFlag

Purpose	CAN transceiver read timeout flag.	
Synopsis	`Std_ReturnType` **`CanTrcv_1_T03_ReadTrcvTimeoutFlag`** `( uint8 Transceiver , CanTrcv_TrcvFlagStateType * FlagState );`	
Service ID	0x0B	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	`Transceiver`	Index of the transceiver FlagState Set the CFS status
Return Value	Result of operation	
	`E_OK`	flag CFS as been read

	`E_NOT_OK`	error during acces to CFS flag
Description	This function read the flag CFS to know if it has been set and fill in FlagState	

4.2.2.13. CanTrcv_1_T03_SetOpMode

Purpose	CAN transceiver set operation mode function.	
Synopsis	`Std_ReturnType` **`CanTrcv_1_T03_SetOpMode`** `( uint8 Transceiver , CanTrcv_TrcvModeType OpMode );`	
Service ID	0x01	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	`Transceiver`	Index of the transceiver
	`OpMode`	Desired operating mode
Return Value	Result of operation	
	`E_OK`	transceiver state has been changed to the requested mode
	`E_NOT_OK`	transceiver state change has failed or the parameter is out of allowed range
Description	This function sets the mode of the transceiver	

4.2.2.14. CanTrcv_1_T03_SetPNActivationState

Purpose	Manage PN (dis)activation.	
Synopsis	`Std_ReturnType` **`CanTrcv_1_T03_SetPNActivationState`** `( CanTrcv_PNActivationType ActivationState );`	
Service ID	0x0F	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	`ActivationState`	(dis)active PN mode
Return Value	Result of operation	
	`E_OK`	PN has been changed to the requested configuration

	`E_NOT_OK`	PN configuration change has failed
Description	Calling this function allows to activate or not the PN of the transceiver modifying CPNC bit	

4.2.2.15. CanTrcv_1_T03_SetWakeupMode

Purpose	CAN transceiver set wakeup mode function.	
Synopsis	`Std_ReturnType` **`CanTrcv_1_T03_SetWakeupMode`** `( uint8 Transceiver , CanTrcv_TrcvWakeupModeType TrcvWakeupMode );`	
Service ID	0x05	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	`Transceiver`	Index of the transceiver
	`TrcvWakeupMode`	Requested transceiver wakeup reason
Return Value	Result of operation	
	`E_OK`	wakeup state has been changed to the requested mode
	`E_NOT_OK`	wakeup state change has failed or the parameter is out of the allowed range
Description	This function enables, disables or clears wake-up events	

4.3. Integration notes

4.3.1. Integration requirements

WARNING ⚠

Integration requirements list is not exhaustive

The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user's guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

Integration requirements are not listed for the CanTrcv_1_T03 module.