Elektrobit

EB tresos® AutoCore Generic 8

CAN Stack documentation

release notes update for the CanTp module

product release 8.5.1

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

Table of Contents

1. Overview

This document provides you with the release notes to accompany an update to the `CanTp` module. Refer to the changelog Section 2.1, “Change log” for details of changes made for this update.

2. CanTp module release notes

- AUTOSAR R4.0 Rev 3
- AUTOSAR SWS document version: 4.0.0
- Module version: 6.8.15.B210702
- Supplier: Elektrobit Automotive GmbH

2.1. Change log

This chapter lists the changes between different versions.

Module version 6.8.15

2018-08-24

- Added support for CanTp_ChangeTxParameter() for TX N-SDUs.

Module version 6.8.14

2018-07-27

- Improving CanTp_MainFunction Performance.

Module version 6.8.13

2018-06-22

- Allow TxConfirmation() to request another transmission of the same PDU.
- ASCCANTP-1228 Fixed known issue: The dynamic STmin value change during segmented transmission is not allowed.
- ASCCANTP-1253 Fixed known issue: Invalid N_PCI of Single Frame with FD accepted.

Module version 6.8.12

2018-05-25

- ASCCANTP-1209 Fixed known issue: Wrong padding value for small CAN-FD frames.

Module version 6.8.11

2018-04-20

- Add support for uint32 PduLengthType

Module version 6.8.10

2018-03-16

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

Module version 6.8.9

2018-02-16

- ASCCANTP-1206 Fixed known issue: CAN-FD Flow Control length erroneously computed for CAN-FD messages.

Module version 6.8.8

2018-01-19

- ASCCANTP-1174 Fixed known issue: Configuration of CAN 2.0 frames with length less than 8 bytes is not allowed.

Module version 6.8.7

2017-12-15

- Reduce memory consumption by introducing parallel channels.

Module version 6.8.6

2017-09-22

- Improve usage of critical section for idle channels.

Module version 6.8.5

2017-08-25

- ASCCANTP-1157 Fixed known issue: Usage of mixed CAN 2.0 and CAN-FD PDUs not allowed.

Module version 6.8.4

2017-07-28

- ASCCANTP-1150 Fixed known issue: Incorrect compiler abstraction used in CanTp_RxIndication.
- Post-build selectable support.
- ASCCANTP-1148 Fixed known issue: Wrong data type is used for the struture member NumberOfChannels.

Module version 6.8.3

2017-06-30

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

Module version 6.8.2

2017-06-02

- Provide CanTp_ReadParameter() API.

Module version 6.8.1

2017-05-05

- Report Det error when a postponed Rx frame is overwritten.

Module version 6.8.0

2017-03-31

- The number of CanTpRx/TxChannels as well as CanTpRx/TxNSdus that are supported by the implementation is extended.

Module version 6.7.1

2017-03-10

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

Module version 6.7.0

2017-03-03

- Configurable CAN-FD PDUs padding length to 64 bytes.
- Move integration requirements to separate reqm file.

Module version 6.6.1

2017-02-03

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

Module version 6.6.0

2017-01-05

- Different padding byte values for CAN 2.0 and CAN FD PDUs.
- Different configurable timeout values for repeated FC WAIT and other FC PDUs

Module version 6.5.12

2016-12-02

- ASCCANTP-1078 Fixed known issue: Behavior upon reception of unexpected PDUs deviates from ISO/CD 15765-2:2014 and AUTOSAR 4.1.x/4.2.x

Module version 6.5.11

2016-11-04

- Adapted resource file for the scheduling of main functions to the split of `IpduM_MainFunction()` into `IpduM_MainFunctionRx()` and `IpduM_MainFunctionTx()`.
- Improve the description of CanTpNSa and CanTpNTa
- Remove compiler warnings with GHS multi C Compiler v2014.1.6

Module version 6.5.10

2016-07-01

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

Module version 6.5.9

2016-04-29

- ASCCANTP-1059 Fixed known issue: CanTp sends flow control frame with status overflow as response on discarded single frame

Module version 6.5.8

2016-02-05

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

Module version 6.5.7

2015-11-06

- Removed the usage EcuC PduLength as maximum for I-PDUs
- ASCCANTP-1045 Fixed known issue: If there are multiple CanTpRxNPduId or CanTpRxFcNPduId instances configured to value 0, CanTp_SetNSa() fails
- Updated memory section naming

Module version 6.5.6

2015-06-19

- Added support to transmit and receive segmented messages with more than 4095 bytes
- Added CAN FD support to transmit and receive N-PDUs with length up to 64 bytes
- ASCCANTP-1037 Fixed known issue: CanTp_SetNSa()/CanTp_GetNSa() APIs write and read source addresses for wrong N-SDUs

Module version 6.5.5

2015-02-20

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

Module version 6.5.4

2015-01-07

- Added support for configurable mapping of CanTp_IsValidConfig function to dedicate memory section
- Removed AUTOSAR 3.x compliant symbolic name value macros and updated the logic to only provide AUTOSAR 4.0.2 compliant macros if macro CANTP_PROVIDE_LEGACY_SYMBOLIC_NAMES is defined

Module version 6.5.3

2014-10-02

- Improved state machine to allow expected incoming frames (CTS, CF) before outgoing frames (CF, CTS) are confirmed
- Update range check of `CanTpGptChannelResolution` to prevent that it is configured to zero

Module version 6.5.2

2014-04-24

- Updated address space to allow incoming N-PDU and FC N-PDU with same address
- ASCCANTP-964 Fixed known issue: The service needs assistant tries to generate (non existent) CanTp Dem events and reports a warning that shall be ignored
- ASCCANTP-983 Fixed known issue: Compilation aborts and reports an error if memory mapping is used for memory sections `CANTP_START_CONFIG_DATA_UNSPECIFIED` and `CANTP_START_SEC_CODE`
- Introduced memory section for jump table shared variables
- ASCCANTP-985 Fixed known issue: Compilation aborts if Dbg function call tracing is enabled for internal CanTp function `CanTp_RequestTxFrameData()`
- ASCCANTP-986 Fixed known issue: Build error due to missing file CanTp_PBcfg.c if code generation for CanTp is disabled and only post-build configuration is compiled
- Updated block size value for segmented frame reception

Module version 6.5.1

2013-10-10

- ASCCANTP-913 Fixed known issue: CanTp expects the upper layer to provide data sufficient to fill a CF which may lead to a N_Cr timeout

- Updated symbolic name value naming schema according to AUTOSAR 4.0 rev3
- Updated MCG to generate XML code for Binary Code Generation

Module version 6.5.0

2013-06-18

- ASCCANTP-804 Fixed known issue: The functions `CanTp_CancelReceive()` and `CanTp_CancelTransmit()` incorrectly report a DET error
- ASCCANTP-836 Fixed known issue: `CanTp_Transmit` does not check the N-Sdu data size for functional addressing properly
- Added checking of configuration and platform specific signature to prevent loading of incompatible post-build configuration
- Added checking of published information signature to prevent loading of incompatible post-build configuration
- ASCCANTP-862 Fixed known issue: Post-build configuration does not work if jumptables are enabled
- Implemented `CanTpTc`
- Updated handle ID wizard to set the configuration parameters `CanTpRxNPduId` and `CanTpRxFcNPduId` also for extended and mixed addressing format

Module version 6.4.0

2013-02-08

- Add relocatability to post build configuration
- ASCCANTP-729 Fixed known issue: If `CanTpNcs` is equal to `CanTpMainFunctionPeriod`, a timeout always occurs
- ASCCANTP-598 Fixed known issue: If a timeout occurs, CanTp might report `NTFRSLT_E_NOT_OK` instead of the timeout specific error code
- Update block size calculation to AUTOSAR 4.0 rev3

Module version 6.3.0

2012-10-16

- Update BSW to AUTOSAR 4.0 rev3 TP API
- Migration to ASR 4.0 ComStack HandleId Policy

- ASCCANTP-738 Fixed known issue: Automatic assignment of `CanTp_MainFunction()` to a task for periodic execution works only if the `CanTpConfig` is named `CanTpConfig_0`
- The top-level structure of the software-component description in the ARXML files changed from `/AUTOSAR/CanTp` to `/AUTOSAR_CanTp`

Module version 6.2.0

2012-06-20

- Updated config according to AUTOSAR 4.0 rev3
- Introduce post build data structure

Module version 6.1.0

2012-03-16

- ASCCANTP-612 Fixed known issue: CanTp uses default N_Cs value for buffer handling timeouts if no physical Tx N-SDU is configured
- Updated naming scheme for #defines for symbolic name values to AUTOSAR 4.0 rev3 naming scheme

Module version 6.0.1

2011-09-30

- ASCCANTP-597 Fixed known issue: Compilation of CanTp generates a compiler warning or fails with an error due to the use of invalid preprocessor directive in the source code
- Updated Dem handling (except the configuration) to AUTOSAR 4.0

Module version 6.0.0

2011-09-02

- Initial AUTOSAR 4.0 version

2.2. New features

- Allow TxConfirmation() to request another transmission of the same PDU.

- Added support for CanTp_ChangeTxParameter() for TX N-SDUs.

2.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

- The parameter `CanTpGeneral/CanTpDynamicNSaEnabled` was added to configure the handling of N_SA values. It allows to configure the following handling of N_SA values:
  - TRUE: N_SA values can be set and get via API interface functions.
  - FALSE: Use of N_SA values as configured in ROM (default).
- The module can be used as jump table module from several applications.

  In this case, one application must be the jump table server, that implements the jump table and all functionality. The other applications can then configure the CanTp as jump table client which means that the functionality is reduced to wrapper functions or macros, that call the jump table server functions.
- The module can recover from lower layer transmit errors.

  If the call to `CanIf_Transmit()` fails, the module does not return an error immediately. Instead, it tries to transmit the frame until the N_As timeout has elapsed and then it notifies the upper layer upon failure.
- The module tolerates received padded frames even if padding is disabled.

  If the CanTp module of the sender is configured with padding enabled and the CanTp module of the receiver is configured with padding disabled, the CanTp module of the receiver tolerates received padded frames and processes them.
- The module is able to receive and transmit segmented frames with up to 65535 bytes of payload.

  The module is able to transmit and receive segmented frames to transport up to 65535 bytes of data. For a payload greater than 4095 the module will use the first frame format as specified in AUTOSAR 4.2.1.
- The module supports CAN FD to transmit and receive PDUs up to 64 byte.

  The module is also able to transmit and receive frames with PDU length of 12, 16, 20, 24, 32, 48, or 64. For single frames with PDU size greater 8 the module will use the single frame format as specified in AUTOSAR 4.2.1. This feature can be enabled by the configuration parameter `CanTpGeneral/CanTpFlexibleDataRateSupport`. The maximum allowed PDU size can be configured for every EcuC PDU which is referred by `CanTpRxNPduRef` or `CanTpTxNPduRef`.
- The CanTp module supports different configurable padding values for CAN-FD frames.

  The module is able to transmit different padding values for CAN-FD and CAN 2.0 PDUs. This feature can be enabled by the configuration parameter `CanTpGeneral/CanTpPaddingByteCanFD` allowing a maximum padding value up to 255.

- The module supports different configurable timeout values for repeated FC WAIT PDUs.

  The module is able to transmit Flow Control frames with WAIT status using a different timeout value for repeated Flow Control WAIT PDUs. This feature can be enabled by the configuration parameter `CanTpGeneral/CanTpNbrWaitRepeatedSupport`. The Flow Control PDU timeout can be configured for every RxNSdu via `CanTpNbrWaitRepeated`.

- The CanTp module supports mandatory padding of CAN FD PDUs to 64 bytes and CAN 2.0 PDUs to 8 bytes, if CANTP_ON_CAN_CAN_FD is configured.

  The module is able to transmit CAN FD PDUs with the maximum value of 64 bytes. This feature can be enabled by the configuration parameter CanTpTxPaddingActivation (on transmission) and CanTpRxPaddingActivation (on reception). The possible enumeration literals of the existing config parameters CanTpTxPaddingActivation and CanTpRxPaddingActivation are extended by the literal CANTP_ON_CAN_CAN_FD: - CANTP_OFF: No padding needed - CANTP_ON: Enable mandatory padding to 8 bytes for CAN 2.0 PDUs only - CANTP_ON_CAN_CAN_FD: Enable mandatory padding to 8 bytes for CAN 2.0 PDUs and 64 bytes for CAN FD PDUs

- The module supports an enlarged upper limit of NSdus and channels.

  The CanTp module supports up to 65535 half duplex or 32767 full duplex connection channels. The maximum number of TxNSdus, as well as RxNSdus was extended to 32767.

- The module reports an error when a queued CF is overwritten.

  The module reports a Det error when a postpones Rx frame is overwritten.

- The module provides CanTp_ReadParameter() API.

  To get a high performance link between a tester and an ECU during the network, the TP has to be speed up by changing FlowControl parameters like STmin and Blocksize. After changing STmin and Blocksize parameters, CanTp module provides an interface to read the current values for STmin and Blocksize from the CAN-TP. The use case for reading the TP parameters is to have the possibility to check the values of the parameters after writing them.

- The module supports parallel channels.

  Parallel channels are an efficient and fast way to reduce RAM consumption. Information during run-time can be stored using parallel channels. When CanTpMaxParallelChannels are configured the major amount of required global RAM is given by the array CanTp_Channel which dimension is equal with the maximum number or parallel channels.

- CanTp_MainFunction better performance when all channels are idle.

  If all channels are Idle, CanTp_MainFunction shall exit immediately without executing any functionality or entering unwanted critical sections. That improves efficiently the execution time of CanTp_MainFunction API.

- The module provides CanTp_ChangeTxParameter() API.

The module provides the CanTp_ChangeTxParameter() API support to change the transmit parameter STmin.

- The module provides CanTp_ResetTxParameter() API.

  The module provides the CanTp_ResetTxParameter() API support to reset the STmin parameter value.

- The module provides configuration parameter CANTP_CHANGE_TX_PARAMETER_REQ_API.

  This feature can be enabled by the configuration parameter CANTP_CHANGE_TX_PARAMETER_REQ_-API API .

- The module provides CanTp_ChangeRxParameter() and CanTp_ChangeParameter() APIs.

  This two APIs can be used at a time, but never together.

2.4. Deviations

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

- Flow control frames are sent immediately without respecting timeout N_Br.

  Description:

  During reception of a segmented message, ISO 15765-2 chapter 6.7.1 mandates to wait for N_Br to elapse before sending a flow control (FC) frame. For the CanTp implementation, the flow control messages FC(CTS) and FC(OVFLW) are sent immediately when the corresponding conditions (buffer available, buffer request failed permanently) are met. The flow control message FC(WT) is sent after N_Br has elapsed.

  Rationale:

  To improve bus performance, feedback is provided immediately when it is known. FC(WT) is only sent if needed.

- Initialization check in main function

  Description:

  If the main function is called while the module is not yet initialized the main function returns immediately without performing any functionality and without raising any Det error. This initialization check is always performed independent of the development error detection setting.

  Rationale:

The RTE module may schedule the modules main function before the module is initialized. This would result in lots of Det errors during start up. Therefore the module's main function does not throw a Det error if the module is not yet initialized and simply returns in this case.

- CanTp does not report `CANTP_E_INVALID_TX_BUFFER` and `CANTP_E_INVALID_RX_BUFFER`.

  Description:

  CanTp does not provide any DET checks which reports the error `CANTP_E_INVALID_TX_BUFFER` or `CANTP_E_INVALID_RX_BUFFER`.

  Rationale:

  With the change of the AUTOSAR Tp API in AUTOSAR 4.0, the CanTp DET errors `CANTP_E_INVALID_TX_BUFFER` and `CANTP_E_INVALID_RX_BUFFER` are obsolete. See Bugzilla http://www.autosar.org/bugzilla/show_bug.cgi?id=56264.

  Requirements:

  CANTP293

- CanTp does not provide the API function `CanTp_Shutdown()` (reference to product description: ASCPD-96).

  Description:

  The API function `CanTp_Shutdown()` is not implemented in the CanTp module.

  Rationale:

  There is no AUTOSAR internal user for the API function `CanTp_Shutdown()` and the behavior and operating constraints are not clearly specified in the AUTOSAR SWS. Using the function might by risky since expectations and actual behavior might differ, so it was decided to skip the function implementation.

  Requirements:

  CANTP010, CANTP211, CANTP202, CANTP200

- `PduR_CanTpChangeParameterConfirmation()` must not be called.

  Description:

  The callback function `PduR_CanTpChangeParameterConfirmation()` is not used to notify the upper layer about the result of the `CanTp_ChangeParameter()` function call.

  Rationale:

Since `CanTp_ChangeParameter()` is specified as synchronous, there is no need to use the callback function `PduR_CanTpChangeParameterConfirmation()` to notify the upper layer. The return value is sufficient. Also see Bugzilla http://www.autosar.org/bugzilla/show_bug.cgi?id=46227.

Requirements:

CANTP304, CANTP305, CANTP306

- Notification result `NTFRSLT_E_CANCELATION_OK` and `NTFRSLT_E_CANCELATION_NOT_OK` not used

Description:

CanTp reports a successful cancellation with `PduR_CanTpRxIndication()`/`PduR_CanTpTxConfirmation()` using the notification result `NTFRSLT_E_NOT_OK`. In case that the cancellation was not successful, `PduR_CanTpRxIndication()`/`PduR_CanTpTxConfirmation()` is not called.

Rationale:

Due to the decision in the Bugzilla issue http://www.autosar.org/bugzilla/show_bug.cgi?id=52106 the notification result `NTFRSLT_E_CANCELATION_OK` and `NTFRSLT_E_CANCELATION_NOT_OK` shall not be used. Instead a successful cancellation shall be reported with `PduR_CanTpRxIndication()`/`PduR_CanTpTxConfirmation()` using the notification result NTFRSLT_E_NOT_OK.

Requirements:

CANTP244, CANTP255, CANTP263

- No AUTOSAR Debug and Trace support

Description:

CanTp is not instrumented for the usage with AUTOSAR Debug and Trace.

Requirements:

CANTP249, CANTP250, CANTP251, CANTP252, CANTP253

- `CanTpRxDl` and `CanTpTxDl` are not used

Description:

The configuration parameters `CanTpRxDl` and `CanTpTxDl` are not used.

Rationale:

Based on RFC53101 the `CanTpRxDl` and `CanTpTxDl` are deprecated and shall not be used in the future. http://www.autosar.org/bugzilla/show_bug.cgi?id=53101

Requirements:

CANTP280_Conf, CANTP267_Conf

- Det errors `CANTP_E_TX_COM`, `CANTP_E_RX_COM` and `CANTP_E_COM` are not reported

  Description:

  In case that a connection is aborted due to a timeout or other connection related issues, the module does not report a Det error.

  Rationale:

  These Det reports are no real development error information but additional runtime information in case that a connection problem occurs. However, in this case the upper layer is informed about the reason of the aborted connection anyway. The Det report does not provide any additional or relevant information if this happens. Find the discussion to this topic at http://www.autosar.org/bugzilla/show_bug.cgi?id=52569

  Requirements:

  CANTP229, CANTP293

- No consistency check between code files and header files

  Description:

  The inter-module version checks as specified by the SWS are not implemented.

  Rationale:

  - The required compile-time version checks would result in an inflexible basic software stack hardly to integrate.
  - EB tresos AutoCore is an already integrated product.
  - The project handling of EB tresos Studio provides means to enforce that only modules with the same AUTOSAR release version can be added to the project.

  Requirements:

  CANTP307, CANTP308

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.

- Limitation on the number of connection channels

  Description:

The CanTp supports up to 65535 half duplex or 32767 full duplex connection channels. If full and half duplex channels are mixed twice the number of full duplex channels plus the number of half duplex channels must be lower than or equal to 65535.

Rationale:

This limitation allows to use 2 byte to identify channels and therefore reduces the ROM size of the configuration.

- Limitation on the number of N-SDUs

  Description:

  The maximum number of `CanTpRxNSdus` and `CanTpTxNSdus` are implementation dependent and limited to 32767 each.

- Limitation on parameter `CanTpRxWftMax`

  Description:

  The CanTp supports a maximum value of 255 for parameter `CanTpRxWftMax`.

  Rationale:

  This limitation allows to use 1 byte for the `N_WFTmax` parameter and therefore reduces the ROM size of the configuration.

- Maximum data size of segmented frames is limited to 65535

  Description:

  CanTp only supports to transmit N-SDUs and receive I-PDUs which do not exceed 65535.

  Rationale:

  The `PduLengthTypeEnum` in `EcuC` is limited to 16 bit.