

<b>Document Title</b>	SWS_TTCANInterface: Complete Change Documentation 4.3.0 - 4.3.1
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	695

<b>Document Status</b>	Final
<b>Part of AUTOSAR Standard</b>	Classic Platform
<b>Part of Standard Release</b>	4.3.1

## Table of Contents

1	SWS_TTCANInterface	3
1.1	Specification Item SWS_TtCanIf_00071 . . . . .	3

# 1 SWS\_TTCANInterface

## 1.1 Specification Item SWS\_TtCanIf\_00071

### Trace References:

none

### Content:

If development error detection for TtcanIf is enabled: The function CanIf\_TTGetErrorLevel() shall raise the error CAN\_E\_PARAM\_POINTER and shall return CANE\_NOT\_OK if the parameter CanIf\_TTErrorLevel is a NULL pointer.

### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77952: [Can][CanIf] Incompatible return types of Can and CanTrcv

#### Problem description:

While CanTrcv only uses Std\_ReturnType, Can uses Can\_ReturnType in many places, even when only CAN\_OK and CAN\_NOT\_OK are available.

This leads to complicated code in CanIf, because it needs to implement separate checks for return values from CanTrcv and Can and cannot just combine the results.

#### Agreed solution:

=== CanDrv ===

Change of SWS\_Can\_00039 Can\_ReturnType:

- \* type change from enumeration to extra\_literal
- \* Remove range element CAN\_OK
- \* Remove range element CAN\_NOT\_OK
- \* Assign value "0x02" to range element "CAN\_BUSY"
- \* Description: Overlaid return value of Std\_ReturnType for CAN driver API Can\_Write().

~SWS\_Can\_00230 Can\_SetControllerMode

Syntax: Std\_ReturnType Can\_SetControllerMode( uint8 Controller, Can\_StateTransitionType Transition )

Return value:

Std\_ReturnType

E\_OK: request accepted

E\_NOT\_OK: request not accepted, a development error occurred

~SWS\_Can\_00360 Can\_CheckWakeup  
 Syntax: Std\_ReturnType Can\_CheckWakeup( uint8 Controller )  
 Return value:  
 Std\_ReturnType  
 E\_OK: API call has been accepted  
 E\_NOT\_OK: API call has not been accepted

~SWS\_Can\_00233 Can\_Write  
 Syntax: Std\_ReturnType Can\_Write( Can\_HwHandleType Hth, const  
 Can\_PduType\* PduInfo )  
 Return value:  
 Std\_ReturnType  
 E\_OK: Write command has been accepted  
 E\_NOT\_OK: development error occurred  
 CAN\_BUSY: No TX hardware buffer available or pre-emptive call of Can\_Write that  
 can't be implemented re-entrant (see Can\_ReturnType)

Rename CAN\_OK to E\_OK and CAN\_NOT\_OK to E\_NOT\_OK:

- ~SWS\_Can\_00048
- ~SWS\_Can\_00089
- 7.11.5 Return Values
- ~SWS\_Can\_00198
- ~SWS\_Can\_00199
- ~SWS\_Can\_00200
- ~SWS\_Can\_00216
- ~SWS\_Can\_00217
- ~SWS\_Can\_00218
- ~SWS\_CAN\_00219
- ~SWS\_CAN\_00505
- ~SWS\_CAN\_00506
- ~SWS\_Can\_00212

=== CanIf ===

Adapt API Can\_Write() to new signature:

- \* Figure 7.10 "Transmission request with multiple CAN Drivers - simplified"
- \* Figure 9.1 "Transmission request with a single CAN Driver"
- \* Figure 9.2 "Transmission request with multiple CAN Drivers"
- \* Figure 9.5 "Transmit confirmation with buffering"
- \* Figure 9.6 "Transmit Cancelation"
- \* Figure 9.7 "Trigger Transmit Request"

Adapt API Can\_SetControllerMode() to new signature:

- \* Figure 9.11: Start CAN network
- \* Figure 9.13: BusOff recovery

Figure 9.13: Change typo "Cnange" to "Change"

~SWS\_CANIF\_00678: If all calls of Can\_CheckWakeup() or CanTrcv\_CheckWakeup() return E\_NOT\_OK to CanIf, then CanIf\_CheckWakeup() shall return E\_NOT\_OK.

~SWS\_CANIF\_00720: If at least one function call of Can\_CheckWakeup() or CanTrcv\_CheckWakeup() returns E\_OK to CanIf, then CanIf\_CheckWakeup() shall return E\_OK.

Rename CAN\_OK to E\_OK and CAN\_NOT\_OK to E\_NOT\_OK:  
Note between SWS\_CANIF\_00162 and SWS\_CANIF\_00319  
Table in chapter 9.7 Trigger Transmit Request  
Table in chapter 9.11 Start CAN network

=== CanTrcv ===

Adapt API Can\_SetControllerMode() to new signature:

- \* 9.3 De-Initialization (SPI Synchronous)
- \* 9.4 De-Initialization (SPI Asynchronous)

=== EcuSM ===

Adapt API Can\_CheckWakeup() to new signature:

- \* Figure 42 CAN controller wake up by interrupt
- \* Figure 43 CAN controller or transceiver wake up by polling

=== TTCanIf ===

Adapt API Can\_Write() to new signature:

- \* Figure 9.1: CAN Interface Time Triggered transmission with Job List

Correct API Can\_TTReceive() which has return void instead of Can\_ReturnType indeed:

- \* Figure 9.2: CAN Interface Time Triggered reception with Job List

Rename CAN\_OK to E\_OK and CAN\_NOT\_OK to E\_NOT\_OK:  
~SWS\_TtCanIf\_00071

=== TTCanDrv ===

Rename CAN\_OK to E\_OK and CAN\_NOT\_OK to E\_NOT\_OK:

- ~SWS\_TtCan\_00014
- ~SWS\_TtCan\_00018

~SWS\_TtCan\_00022  
~SWS\_TtCan\_00026  
~SWS\_TtCan\_00059  
~SWS\_TtCan\_00078  
~SWS\_TtCan\_00112

=== XCP ===

Adapt API Can\_Write() to new signature:

\* Figure 5: Xcp on Can Transmit

–Last change on issue 77952 comment 22–

**BW-C-Level:**

Application	Specification	Bus
1	4	1