

TRAVEO™ T2G family AUTOSAR MCAL CAN release notes

SRN223389 version 1.17

About this document

Scope and purpose

Thank you for your interest in the TRAVEO™ T2G family AUTOSAR MCAL CAN driver version 1.17. This document lists the installation requirements, software changes, limitations, and known issues.

Intended audience

This document is intended for anyone who uses the controller area network (CAN) driver of the TRAVEO™ T2G family.

Table of contents

About this document	1
Table of contents	1
1 System requirements and recommendations.....	3
1.1 Supported compilers	3
1.2 Compiler options.....	3
1.3 Library compiler options.....	4
1.4 Memory consumption	4
1.5 Stack consumption	5
1.5.1 Green Hills Software.....	5
1.5.2 IAR Embedded Workbench	6
1.6 Note on "_Bswmd.arxml"	7
1.7 Release details.....	7
2 Installation	9
3 Deviations from AUTOSAR.....	10
4 Limitations	17
5 Known defects	33
6 Documentation	34
7 Technical support.....	35
8 Version history	36
8.1 Module SW-Version 1.3.....	36
8.2 Module SW-Version 1.4.....	36
8.3 Module SW-Version 1.5.....	39
8.4 Module SW-Version 1.6.....	40
8.5 Module SW-Version 1.7.....	42
8.6 Module SW-Version 1.8.....	43
8.7 Module SW-Version 1.9.....	45
8.8 Module SW-Version 1.10.....	46

Table of contents

8.9	Module SW-Version 1.11.....	47
8.10	Module SW-Version 1.12.....	47
8.11	Module SW-Version 1.13.....	48
8.12	Module SW-Version 1.14.....	48
8.13	Module SW-Version 1.15.....	50
8.14	Module SW-Version 1.16.....	50
8.15	Module SW-Version 1.17.....	51

System requirements and recommendations

1 System requirements and recommendations

Software prerequisites	Supported version
EB tresos Studio package for Infineon	26.2.0

1.1 Supported compilers

Green Hills Software, compiler v2017.1.4

IAR Embedded Workbench 8.0, EWARM FS 8.22.3

1.2 Compiler options

This section summarizes the compiler options used to build and test the module. When changing the compiler options, the module must be considered untested.

Compiler	Option (Cortex®-M4F core)
Green Hills Software, compiler v2017.1.4	-cpu=cortexm4f -thumb -thumb_lib -C99 --short_enum -align4 --no_commons --no_alternative_tokens -asm3g -preprocess_assembly_files -nostartfiles -globalcheck=normal -globalcheck_qualifiers --prototype_errors -Wformat -Wimplicit-int -Wshadow -Wtrigraphs -Wundef -reject_duplicates -c -list -Ospeed -OI -Olink -Ointerproc -Omax -fsingle

Compiler	Option (Cortex®-M7 core)
Green Hills Software, compiler v2017.1.4	-cpu=cortexm7 -thumb -thumb_lib -C99 --short_enum -align4 --no_commons --no_alternative_tokens -asm3g -preprocess_assembly_files -nostartfiles -globalcheck=normal -globalcheck_qualifiers --prototype_errors -Wformat -Wimplicit-int -Wshadow -Wtrigraphs -Wundef -reject_duplicates -c -list -Ospeed -OI -Olink -Ointerproc -Omax -fhard

Compiler	Option (Cortex®-M4F core)
IAR Embedded Workbench 8.0, EWARM FS 8.22.3	--debug --endian=little --cpu=Cortex-M4 -e --fpu=VFPv4_sp -Ohs --no_size_constraints

Compiler	Option (Cortex®-M7 core)
IAR Embedded Workbench 8.0, EWARM FS 8.22.3	--debug --endian=little --cpu=Cortex-M7 -e --fpu=VFPv5_d16 -Ohs --no_size_constraints

System requirements and recommendations

1.3 Library compiler options

If a binary library has been delivered with this module, it has been built using the following options:

Compiler	Option (Cortex®-M4F core)
Green Hills Software, compiler v2017.1.4	<code>-cpu=cortexm4f -thumb -thumb_lib -C99 --short_enum -align4 --no_commons --no_alternative_tokens -asm3g -preprocess_assembly_files -nostartfiles -globalcheck=normal -globalcheck_qualifiers --prototype_errors -Wformat -Wimplicit-int -Wshadow -Wtrigraphs -Wundef -reject_duplicates -c -list -Ospeed -OI -Olink -Ointerproc -Omax -fsingle</code>

Compiler	Option (Cortex®-M7 core)
Green Hills Software, compiler v2017.1.4	<code>-cpu=cortexm7 -thumb -thumb_lib -C99 --short_enum -align4 --no_commons --no_alternative_tokens -asm3g -preprocess_assembly_files -nostartfiles -globalcheck=normal -globalcheck_qualifiers --prototype_errors -Wformat -Wimplicit-int -Wshadow -Wtrigraphs -Wundef -reject_duplicates -c -list -Ospeed -OI -Olink -Ointerproc -Omax -fhard</code>

Compiler	Option (Cortex®-M4F core)
IAR Embedded Workbench 8.0, EWARM FS 8.22.3	<code>--debug --endian=little --cpu=Cortex-M4 -e --fpu=VFPv4_sp -Ohs --no_size_constraints</code>

Compiler	Option (Cortex®-M7 core)
IAR Embedded Workbench 8.0, EWARM FS 8.22.3	<code>--debug --endian=little --cpu=Cortex-M7 -e --fpu=VFPv5_d16 -Ohs --no_size_constraints</code>

1.4 Memory consumption

GHS (Can_lib) section	Size (in bytes)
.text	6734
.rodata	124
.bss	2
Combined	6860

GHS (Can_src) section	Size (in bytes)
.text	6732
.rodata	1871
.bss	216
Combined	8819

System requirements and recommendations

IAR (Can_lib) section	Size (in bytes)
.text	6856
.rodata	0
.bss	2
Combined	6858

IAR (Can_src) section	Size (in bytes)
.text	6304
.rodata	1800
.bss	216
Combined	8320

Note: The memory consumption of *_src.lib depends on the configuration.

Note: The listed memory consumption will vary depending on customer configuration.

Explanatory notes for this section

Section	Description
.text	Program code
.data	Variables with explicitly initialized values
.bss	Variables that are not explicitly initialized
.rodata	Read-only data

1.5 Stack consumption

1.5.1 Green Hills Software

Function	Max stack usage (in bytes)
Can_Init	120
Can_MainFunction_Write	28
Can_SetControllerMode	372
Can_DisableControllerInterrupts	28
Can_EnableControllerInterrupts	28
Can_Write	200
Can_GetVersionInfo	20
Can_MainFunction_Read	352
Can_MainFunction_BusOff	48
Can_MainFunction_Wakeup	48
Can_CheckWakeup	28
Can_MainFunction_Mode	32

System requirements and recommendations

Function	Max stack usage (in bytes)
Can_ChangeBaudrate	52
Can_CheckBaudrate	52
Can_SetBaudrate	52
Can_SetBaudrateInChangedClock	56
Can_DeInit	76
Can_SetIcomConfiguration	60
Can_GetStatus	36
Can_Interrupt_CANFD00_Cat1	352
Can_Interrupt_CANFD00_Cat2	352
Can_Interrupt_CANFD01_Cat1	352
Can_Interrupt_CANFD01_Cat2	352
Can_Interrupt_CANFD10_Cat1	352
Can_Interrupt_CANFD10_Cat2	352
Can_Interrupt_CANFD11_Cat1	352
Can_Interrupt_CANFD11_Cat2	352

Note: Stack consumption has been evaluated using the *gstack* utility program, which is part of the Green Hills release package. To enable the measurement of stack consumption in your project, build the source code according to the instructions given in the "Measuring stack consumption" section of the module's user guide.

Note: The listed stack consumption will vary depending on customer configuration.

Note: The GHS stack consumption listed in the release notes was measured using the additional compile option `"-gs"`. The GHS compiler cannot measure stack consumption for the selected optimization level (see compilation options). Green Hills cannot exclude possible effects of `"-gs"` on optimization and stack consumption. Therefore, Infineon cannot guarantee the accuracy of these values. For more information on measuring GHS stack consumption, see the section *gstack* utility program in *Build_arm.pdf*.

1.5.2 IAR Embedded Workbench

Function	Max stack usage (in bytes)
Can_Init	84
Can_MainFunction_Write	160
Can_SetControllerMode	276
Can_DisableControllerInterrupts	32
Can_EnableControllerInterrupts	32
Can_Write	144
Can_GetVersionInfo	16
Can_MainFunction_Read	284
Can_MainFunction_BusOff	48
Can_MainFunction_Wakeup	48

System requirements and recommendations

Function	Max stack usage (in bytes)
Can_CheckWakeup	32
Can_MainFunction_Mode	40
Can_ChangeBaudrate	48
Can_CheckBaudrate	40
Can_SetBaudrate	48
Can_SetBaudrateInChangedClock	48
Can_DeInit	68
Can_SetIcomConfiguration	64
Can_GetStatus	40
Can_Interrupt_CANFD00_Cat1	276
Can_Interrupt_CANFD00_Cat2	276
Can_Interrupt_CANFD01_Cat1	276
Can_Interrupt_CANFD01_Cat2	276
Can_Interrupt_CANFD10_Cat1	276
Can_Interrupt_CANFD10_Cat2	276
Can_Interrupt_CANFD11_Cat1	276
Can_Interrupt_CANFD11_Cat2	276

Note: To enable the measurement of stack consumption in your project, build the source code with the linker option "`--enable_stack_usage --log call_graph`". See stack usage analysis of the IAR C/C++ development guide for details.

Note: The listed stack consumption will vary depending on customer configuration.

1.6 Note on "*_Bswmd.arxml"

Note that the `<Module>_Bswmd.arxml` files are templates that can be freely modified by the customer or RTE vendor.

These are in the `output\generated\swcd` subfolder of your project folder.

Named files are not tested.

1.7 Release details

Module software version

1.17.x

(x=software patch version; see the delivery notes for details)

AUTOSAR specification version (ASR)

4.2.2

System requirements and recommendations

Target

MXS40

MCAL configuration settings

See the resource release notes

Supported derivatives

See the resource release notes

Corresponding Can_MemMap.h stub file version

1.0.1

2 Installation

See the installation manual for EB tresos Studio for INFINEON AUTOSAR software products and installation manual for MCAL42-TRAVEO.

3 Deviations from AUTOSAR

T2MC-2344 - Can - Can_CheckBaudrate is called by CanIf_CheckBaudrate

Title: Can - Can_CheckBaudrate is called by CanIf_CheckBaudrate

Description: The service Can_CheckBaudrate(Controller, Baudrate) shall be called by CanIf_CheckBaudrate() for the requested CAN controller. ()

Reason for rejection: This requirement is associated with the CAN interface module.

T2MC-2226 - Can - Can_SetControllerMode(CAN_T_WAKEUP) lefts SLEEP mode

Title: Can - Can_SetControllerMode(CAN_T_WAKEUP) lefts SLEEP mode

Description: This logical sleep mode shall left only, if function Can_SetControllerMode(CAN_T_WAKEUP) is called. ()

Reason for rejection: The requirement is obsolete because the hardware does support sleep mode.

T2MC-2419 - Can calls Icu_EnableNotification when external controller transitioned to SLEEP

Title: Can calls Icu_EnableNotification when external controller transitioned to SLEEP

Description: Icu_EnableNotification shall be called when "external" CAN controllers have been transitioned to SLEEP state (CANIF_CS_SLEEP). ()

Reason for rejection: External CAN controllers are not supported.

T2MC-2420 - Can calls Icu_EnableNotification when external controller transitioned to STOPPED

Title: Can calls Icu_EnableNotification when external controller transitioned to STOPPED

Description: Icu_DisableNotification shall be called when "external" Can controllers have been transitioned to STOPPED state (CANIF_CS_STOPPED). ()

Reason for rejection: Off-chip CAN controllers are not supported.

T2MC-2225 - Can controller with logical sleep mode

Title: Can controller with logical sleep mode

Description: If the CAN HW does not support a sleep mode, the function Can_SetControllerMode(CAN_T_SLEEP) shall set the CAN controller to the logical sleep mode. ()

Reason for rejection: The requirement is obsolete because the hardware does support sleep mode.

T2MC-2438 - Can defines empty Can_MainFunction_BusOff when no polling at all

Title: Can defines empty Can_MainFunction_BusOff when no polling at all

Description: The Can module may implement the function Can_MainFunction_BusOff as empty define in case no polling at all is used. ()

Reason for rejection: The function is not implemented as a define.

Deviations from AUTOSAR

T2MC-2432 - Can defines empty Can_MainFunction_Read when no polling at all

Title: Can defines empty Can_MainFunction_Read when no polling at all

Description: The Can module may implement the function Can_MainFunction_Read as empty define in case no polling at all is used. ()

Reason for rejection: The function is not implemented as a define.

T2MC-2443 - Can defines empty Can_MainFunction_Wakeup when no polling at all

Title: Can defines empty Can_MainFunction_Wakeup when no polling at all

Description: The Can module may implement the function Can_MainFunction_Wakeup as empty define in case no polling at all is used. ()

Reason for rejection: The function is not implemented as a define.

T2MC-2426 - Can defines empty Can_MainFunction_Write when no polling at all

Title: Can defines empty Can_MainFunction_Write when no polling at all

Description: The Can module may implement the function Can_MainFunction_Write as empty define in case no polling at all is used. ()

Reason for rejection: The function is not implemented as a define.

T2MC-2191 - Can does not generate two functions with identical name

Title: Can does not generate two functions with identical name

Description: In case several CAN Hardware Units (of same or different vendor) are implemented in one ECU the function names, and global variables of the Can modules shall be implemented such that no two functions with the same name are generated. ()

Reason for rejection: The TRAVEO™ T2G platform implements only one type of CAN HW unit.

T2MC-2201 - Can does not support an emulated logical SLEEP state

Title: Can does not support an emulated logical SLEEP state

Description: When the CAN hardware does not support sleep mode and is triggered to transition into SLEEP state, the Can module shall emulate a logical SLEEP state from which it returns only, when it is triggered by software to transition into STOPPED state. ()

Reason for rejection: The CAN hardware supports sleep mode with wakeup.

T2MC-2181 - Can for off-chip CAN controllers includes Spi.h

Title: Can for off-chip CAN controllers includes Spi.h

Description: Can module implementations for off-chip CAN controllers shall include the header file Spi.h. By this inclusion, the APIs to access an external CAN controller by the SPI module [12] are included.()

Reason for rejection: Off-chip CAN controllers are not supported.

Deviations from AUTOSAR

T2MC-2272 - Can handles L-SDU without global access to CAN hardware

Title: Can handles L-SDU without global access to CAN hardware

Description: The Can module shall copy the L-SDU in a shadow buffer, if the CAN Hardware is not globally accessible. ()

Reason for rejection: A shadow buffer does not exist because a hardware FIFO will always be supported.

T2MC-2538 - Can implements container CanMainFunctionRWPeriods

Title: Can implements container CanMainFunctionRWPeriods

Description:

SWS Item	ECUC_Can_00484		
Name	CanMainFunctionPeriod		
Description	This parameter describes the period for cyclic call to <code>Can_MainFunction_Read</code> or <code>Can_MainFunction_Write</code> depending on the referring item. Unit is seconds. Different poll-cycles will be configurable if more than one CanMainFunctionPeriod is configured. In this case multiple <code>Can_MainFunction_Read()</code> or <code>Can_MainFunction_Write()</code> will be provided by the CAN Driver module.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	0.001 .. 65.535		
Default value	--		
Post-Build Variant Value	False		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

Reason for rejection: The multiplicity of `CanMainFunctionRWPeriods` and `CanMainFunctionRWPeriodRef` containers is incompatible between ASR 4.0.3 and 4.2.2. As a consequence, `CanMainFunctionPeriod` cannot be supported with multiplicity 1. It is redundant to `CanMainFunctionPeriodRead` and `CanMainFunctionPeriodWrite` anyway. Therefore, it was decided to remove `CanMainFunctionPeriod`.

T2MC-2190 - Can implements different hardware units

Title: Can implements different hardware units

Description: For CAN Hardware Units of different type, different Can modules shall be implemented. (SRS_BSW_00347)

Reason for rejection: The TRAVEO™ T2G platform implements only one type of CAN HW unit.

Deviations from AUTOSAR

T2MC-2434 - Can implements pattern Can_MainFunction_Read_X if multiplicity greater 1

Title: Can implements pattern Can_MainFunction_Read_X if multiplicity greater 1

Description: The API name of Can_MainFunction_Read() shall obey the following pattern:

- Can_MainFunction_Read_0()
- Can_MainFunction_Read_1()
- Can_MainFunction_Read_2()
- Can_MainFunction_Read_3()

... and so on, if more than one period (see ECUC_Can_00358) is supported. ()

Reason for rejection: Only one such MainFunction is implemented.

T2MC-2428 - Can implements pattern Can_MainFunction_Write_X if multiplicity greater 1

Title: Can implements pattern Can_MainFunction_Write_X if multiplicity greater 1

Description: The API name of Can_MainFunction_Write() shall obey the following pattern:

- Can_MainFunction_Write_0()
- Can_MainFunction_Write_1()
- Can_MainFunction_Write_2()
- Can_MainFunction_Write_3()

... and so on, if more than one period (see ECUC_Can_00356) is supported. ()

Reason for rejection: Only one such MainFunction is implemented.

T2MC-2317 - Can includes Can_GeneralTypes.h

Title: Can includes Can_GeneralTypes.h

Description: If different CAN drivers are used, only one instance of this file has to be included in the source tree. For implementation all Can_GeneralTypes.h related types in the documents mentioned before shall be considered. ()

Reason for rejection: TRAVEO™ T2G supports only one type of CAN controller hardware per device.

T2MC-2304 - Can modules environment indicates errors by return value

Title: Can modules environment indicates errors by return value

Description: The Can module's environment shall indicate Default errors only in the return values of a function of the Can module when DET is switched on and the function provides a return value. The returned value is CAN_NOT_OK. (SRS_BSW_00369, SRS_BSW_00386, SRS_SPAL_12448)

Reason for rejection: The return values of all CAN API functions are independent of the setting of CanDevErrorDetect.

Deviations from AUTOSAR

T2MC-2565 - Can not applicable requirements

Title: Can not applicable requirements

Description: These requirements are not applicable to this specification.

(SRS_BSW_00170, SRS_BSW_00383, SRS_BSW_00395, SRS_BSW_00397, SRS_BSW_00398, SRS_BSW_00399, SRS_BSW_00400, SRS_BSW_00168, SRS_BSW_00423, SRS_BSW_00424, SRS_BSW_00425, SRS_BSW_00426, SRS_BSW_00427, SRS_BSW_00429, SRS_BSW_00433, SRS_BSW_00336, SRS_BSW_00422, SRS_BSW_00417, SRS_BSW_00409, SRS_BSW_00455, SRS_BSW_00162, SRS_BSW_00415, SRS_BSW_00325, SRS_BSW_00326, SRS_BSW_00342, SRS_BSW_00453, SRS_BSW_00413, SRS_BSW_00307, SRS_BSW_00447, SRS_BSW_00353, SRS_BSW_00361, SRS_BSW_00439, SRS_BSW_00449, SRS_BSW_00378, SRS_BSW_00359, SRS_BSW_00440, BSW00443, BSW00444, BSW00445, BSW00446, SRS_SPAL_12163, SRS_SPAL_12462, SRS_SPAL_12068, SRS_SPAL_12064, SRS_Can_01125, SRS_Can_01126)

Reason for rejection: Named RQMs are not applicable.

T2MC-2251 - Can provides an adapted SDU-buffer

Title: Can provides an adapted SDU-buffer

Description: If the presentation inside the CAN Hardware buffer differs from AUTOSAR definition, the Can module must provide an adapted SDU-Buffer for the upper layers. ()

Reason for rejection: The hardware buffer matches the AUTOSAR definition.

T2MC-2418 - Can provides enabling and disabling of wakeup notification

Title: Can provides enabling and disabling of wakeup notification

Description: Can driver shall use the following APIs provided by Icu driver, to enable and disable the wakeup event notification:

- Icu_EnableNotification
- Icu_DisableNotification

()

Reason for rejection: External CAN controllers are not supported.

T2MC-2228 - Can returns from logical sleep mode with no effect to controller state

Title: Can returns from logical sleep mode with no effect to controller state

Description: If the CAN HW does not support a sleep mode, the function

Can_SetControllerMode(CAN_T_WAKEUP) shall return from the logical sleep mode, but have no effect to the CAN controller state (as the controller is already in stopped state). ()

Reason for rejection: The requirement is obsolete because the hardware does support sleep mode.

T2MC-2202 - Can shall remain in STOPPED state, while logical SLEEP state is active

Title: Can shall remain in STOPPED state, while logical SLEEP state is active

Description: The CAN hardware shall remain in state STOPPED, while the logical SLEEP state is active.()

Reason for rejection: The CAN hardware supports sleep mode with wakeup.

Deviations from AUTOSAR

T2MC-2296 - Can supports autonomous sending of messages in pretended networking mode

Title: Can supports autonomous sending of messages in pretended networking mode

Description: Autonomous sending of messages in Pretended Networking mode shall be supported only if additional ICOM hardware is available. A configuration parameter defines if there is hardware support or not (Refer to CanIcomVariant). ()

Reason for rejection: The Icom module is not installed in hardware.

T2MC-2270 - Can supports shadow buffer mechanism instead of hw FIFO

Title: Can supports shadow buffer mechanism instead of hw FIFO

Description: Controllers that do not support a hardware FIFO often provide the capabilities to implement a shadow buffer mechanism, where additional hardware objects take over when the primary hardware object is busy. The number of hardware objects is configured via "CanHwObjectCount". ()

Reason for rejection: Hardware FIFO is supported.

T2MC-2192 - Can uses naming conventions for different CAN controllers

Title: Can uses naming conventions for different CAN controllers

Description: The naming conventions shall be used only in that case, if multiple different CAN controller types on one ECU have to be supported. ()

Reason for rejection: The TRAVEO™ T2G platform implements only one type of CAN HW unit.

T2MC-2271 - Can uses of shadow buffer after reception

Title: Can uses of shadow buffer after reception

Description: The Can module shall copy the L-SDU in a shadow buffer after reception, if the RX buffer cannot be protected (locked) by CAN Hardware against overwriting by a newly received message. ()

Reason for rejection: A shadow buffer does not exist because a hardware FIFO will always be supported.

T2MC-2167 - Can uses synchronous APIs of MCAL drivers

Title: Can uses synchronous APIs of MCAL drivers

Description: The Can module shall use the synchronous APIs of the underlying MCAL drivers and shall not provide callback functions that can be called by the MCAL drivers.()

Reason for rejection: Off-chip CAN controllers are not supported.

T2MC-2470 - Can VARIANT-PRE-COMPILE

Title: Can VARIANT-PRE-COMPILE

Description: VARIANT-PRE-COMPILE: Only pre-compile configuration parameters.()

Reason for rejection: Support post build only

Deviations from AUTOSAR

T2MC-2166 - Can with off-chip controller use other driver

Title: Can with off-chip controller use other driver

Description: If an off-chip CAN controller is used (3), the Can module shall use services of other MCAL drivers (e.g. SPI). (SRS_BSW_00005)

Reason for rejection: Off-chip CAN controllers are not supported.

Limitations

4 Limitations

T2MC-2392 - Can - Can_CheckWakeup is configurable by CanWakeupFunctionalityAPI

Title: Can - Can_CheckWakeup is configurable by CanWakeupFunctionalityAPI

Description: The function Can_CheckWakeup shall be pre compile time configurable On/Off by the configuration parameter: CanWakeupFunctionalityAPI. ()

Limitation: Support post build only

T2MC-2459 - Can - Can_SetIcomConfiguration compiler switch CAN_PUBLIC_ICOM_SUPPORT

Title: Can - Can_SetIcomConfiguration compiler switch CAN_PUBLIC_ICOM_SUPPORT

Description: Can_SetIcomConfiguration() shall be pre compile time configurable ON/OFF by the configuration parameter CAN_PUBLIC_ICOM_SUPPORT. ()

Limitation: Support post build only

T2MC-2638 - Can - Can_Write performs cancellation for L-PDU with identical priority

Title: Can - Can_Write performs cancellation for L-PDU with identical priority

Description: The function Can_Write shall perform following actions if the hardware transmit object is busy with another transmit request for an L-PDU that has identical priority than that for the current request:

- The transmission of the L-PDU with identical priority shall be cancelled (asynchronously) in case CanIdenticalIdCancellation is enabled. Compare to chapter 7.5.1.2.
- The transmission of the L-PDU with identical priority shall not be cancelled in case CanIdenticalIdCancellation is disabled and the function Can_Write is left without any actions.
- The function Can_Write shall return CAN_BUSY

()

Limitation: Cancellation of transmit messages is possible only for HW objects using a dedicated buffer. Cancellation for HW objects using the transmit FIFO is not supported.

T2MC-2637 - Can - Can_Write performs cancellation for L-PDU with lower priority

Title: Can - Can_Write performs cancellation for L-PDU with lower priority

Description: The function Can_Write shall perform following actions if the hardware transmit object is busy with another transmit request for an L-PDU that has lower priority than that for the current request:

- The transmission of the L-PDU with lower priority shall be cancelled (asynchronously) in case transmit cancellation functionality is enabled. Compare to chapter 7.5.1.2.
- The function Can_Write shall return CAN_BUSY

()

Limitation: Cancellation of transmit messages is possible only for HW objects using a dedicated buffer. Cancellation for HW objects using the transmit FIFO is not supported.

Limitations

T2MC-2399 - Can - Can_Write performs no actions if hardware transmit object is busy

Title: Can - Can_Write performs no actions if hardware transmit object is busy

Description: The function Can_Write shall perform no actions if the hardware transmit object is busy with another transmit request for an L-PDU:

1. The transmission of the other L-PDU shall not be cancelled and the function Can_Write is left without any actions
2. The function Can_Write shall return CAN_BUSY

(SRS_Can_01049)

Limitation: This requirement applies only if cancelation is disabled.

T2MC-2498 - Can - CanController contains CanWakeupSupport

Title: Can - CanController contains CanWakeupSupport

Description:

SWS Item	ECUC_Can_00330		
Name	CanWakeupSupport		
Description	CAN driver support for wakeup over CAN Bus.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency			

Limitation: Wake-up support cannot be disabled. The CAN driver will wake up on any non-filtered message as long as the hardware is powered and clocked.

T2MC-2506 - Can - CanControllerBaudrateConfig contains CanControllerPropSeg

Title: Can - CanControllerBaudrateConfig contains CanControllerPropSeg

Description:

SWS Item	ECUC_Can_00073		
Name	CanControllerPropSeg		
Description	Specifies propagation delay in time quantas.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
Post-Build Variant Value	true		

Limitations

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Limitation: The parameter is fixed to 0 because of a hardware limitation.

T2MC-2507 - Can - CanControllerBaudrateConfig contains CanControllerSeg1

Title: Can - CanControllerBaudrateConfig contains CanControllerSeg1

Description:

SWS Item	ECUC_Can_00074		
Name	CanControllerSeg1		
Description	Specifies phase segment 1 in time quantas.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Limitation: CAN controller hardware supports only the range 2..255.

T2MC-2508 - Can - CanControllerBaudrateConfig contains CanControllerSeg2

Title: Can - CanControllerBaudrateConfig contains CanControllerSeg2

Description:

SWS Item	ECUC_Can_00075		
Name	CanControllerSeg2		
Description	Specifies phase segment 2 in time quantas.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Limitations

Limitation: CAN controller hardware supports only the range 2..128.

T2MC-2509 - Can - CanControllerBaudrateConfig contains CanControllerSyncJumpWidth

Title: Can - CanControllerBaudrateConfig contains CanControllerSyncJumpWidth

Description:

SWS Item	ECUC_Can_00383		
Name	CanControllerSyncJumpWidth		
Description	Specifies the synchronization jump width for the controller in time quantas.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
Post-Build Variant Value	true		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Limitation: CAN controller hardware supports only the range 1..128.

T2MC-2513 - Can - CanControllerFdBaudrateConfig contains CanControllerPropSeg

Title: Can - CanControllerFdBaudrateConfig contains CanControllerPropSeg

Description:

SWS Item	ECUC_Can_00476		
Name	CanControllerPropSeg		
Description	Specifies propagation delay in time quantas.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
Post-Build Variant Value	true		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Limitation: The parameter is fixed to 0 because of a hardware limitation.

Limitations

T2MC-2514 - Can - CanControllerFdBaudrateConfig contains CanControllerSeg1

Title: Can - CanControllerFdBaudrateConfig contains CanControllerSeg1

Description:

SWS Item	ECUC_Can_00477		
Name	CanControllerSeg1		
Description	Specifies phase segment 1 in time quantas.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
Post-Build Variant Value	true		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Limitation: CAN controller hardware supports only the range 1..32.

T2MC-2515 - Can - CanControllerFdBaudrateConfig contains CanControllerSeg2

Title: Can - CanControllerFdBaudrateConfig contains CanControllerSeg2

Description:

SWS Item	ECUC_Can_00478		
Name	CanControllerSeg2		
Description	Specifies phase segment 2 in time quantas.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
Post-Build Variant Value	true		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Limitation: CAN controller hardware supports only the range 1..16.

Limitations

T2MC-2516 - Can - CanControllerFdBaudrateConfig contains CanControllerSyncJumpWidth

Title: Can - CanControllerFdBaudrateConfig contains CanControllerSyncJumpWidth

Description:

SWS Item	ECUC_Can_00479		
Name	CanControllerSyncJumpWidth		
Description	Specifies the synchronization jump width for the controller in time quantas.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
Post-Build Variant Value	true		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Limitation: CAN controller hardware supports only the range 1..16.

T2MC-2475 - Can - CanGeneral contains CanDevErrorDetection

Title: Can - CanGeneral contains CanDevErrorDetection

Description:

SWS Item	ECUC_Can_00064 :		
Name	CanDevErrorDetection		
Description	Switches the Development Error Detection and Notification ON or OFF. true: enable (ON). false: disable (OFF).		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	False		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

Limitation: CanDevErrorDetection switches only the development error notification ON or OFF.

Limitations

T2MC-2487 - Can - CanGeneral contains CanSupportTTCANRef

Title: Can - CanGeneral contains CanSupportTTCANRef

Description:

SWS Item	ECUC_CAN_00430 :		
Name	CanSupportTTCANRef		
Description	The parameter refers to CanIfSupportTTCAN parameter in the CAN Interface Module configuration. The CanIfSupportTTCAN parameter defines whether TTCAN is supported.		
Multiplicity	1		
Type	Reference to [CanIfPrivateCfg]		
Post-Build Variant Value	false		
Value ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

Limitation: Lower multiplicity of 0 is allowed, since TTCAN is not supported.

T2MC-2544 - Can - CanIcomConfig contains CanIcomWakeOnBusOff

Title: Can - CanIcomConfig contains CanIcomWakeOnBusOff

Description:

SWS Item	ECUC_Can_00442		
Name	CanIcomWakeOnBusOff		
Description	This parameter defines that the MCU shall wake if the bus off is detected or not.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	true		
Post-Build Variant Value	false		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

Limitation: Depending on the hardware implementation, the ICOM is also able to send messages. But TRAVEO™ T2G does not implement Icom with hardware. Therefore, Icom implements it by software, there is no transmission in pretended networking mode. Since it does not transmit, bus-off never occurs.

Limitations

T2MC-2547 - Can - CanIcomGeneral contains CanIcomLevel

Title: Can - CanIcomGeneral contains CanIcomLevel

Description:

SWS Item	ECUC_Can_00445		
Name	CanIcomLevel		
Description	Defines the level of Pretended Networking. This parameter is reserved for future implementations (Pretended Networking level 2).		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	CAN_ICOM_LEVEL_ONE	--	
	CAN_ICOM_LEVEL_TWO	--	
Default value	CAN_ICOM_LEVEL_ONE		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

Limitation: All ON/OFF for pretended networking mode are controlled with CanPublicIcomSupport only. Therefore, CanIcomGeneral is not supported.

T2MC-2548 - Can - CanIcomGeneral contains CanIcomVariant

Title: Can - CanIcomGeneral contains CanIcomVariant

Description:

SWS Item	ECUC_Can_00446	
Name	CanIcomVariant	
Description	Defines the variant, which is supported by this CanController	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	CAN_ICOM_VARIANT_HW	--
	CAN_ICOM_VARIANT_NONE	--
	CAN_ICOM_VARIANT_SW	--
Default value	CAN_ICOM_VARIANT_NONE	
Post-Build Variant Value	false	

Limitations

Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

Limitation: All ON/OFF for pretended networking mode are controlled with CanPublicComSupport only. Therefore, CanComGeneral is not supported.

T2MC-2552 - Can - CanComRxMessage contains CanComMessageId

Title: Can - CanComRxMessage contains CanComMessageId

Description:

SWS Item	ECUC_Can_00449		
Name	CanComMessageId		
Description	This parameter defines the message ID the wakeup causes of this CanComRxMessage are configured for. In addition, a mask (CanComMessageIdMask) can be defined, in that case it is possible to define a range of rx messages, which can create a wakeup condition.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 536870912		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	--	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

Limitation: Range: set to the following range by CanIdType: CanIdType=STANDARD: 0 ... 2047, CanIdType=EXTENDED: 0 ... 536870911

T2MC-2553 - Can - CanComRxMessage contains CanComMessageIdMask

Title: Can - CanComRxMessage contains CanComMessageIdMask

Description:

SWS Item	ECUC_Can_00465		
Name	CanComMessageIdMask		
Description	Describes a mask for filtering of CAN identifiers. The CAN identifiers of incoming messages are masked with this CanComMessageIdMask. If the masked identifier matches the masked value of CanComMessageId, it can		

Limitations

	create a wakeup condition for this CanIcomRxMessage. Bits holding a 0 mean don't care, i.e. do not compare the message's identifier in the respective bit position. The mask shall be built by filling with leading 0.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 536870912		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	--	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU dependency: CanIcomMessageIdMask and CanIcomRxMessageSignalConfig shall not be defined together.		

Limitation: Range: set to the following range by CanIdType: CanIdType=STANDARD: 0 ... 2047,
CanIdType=EXTENDED: 0 ... 536870911

T2MC-2561 - Can - CanIcomRxMessageSignalConfig contains CanIcomSignalRef

Title: Can - CanIcomRxMessageSignalConfig contains CanIcomSignalRef

Description:

SWS Item	ECUC_Can_00456		
Name	CanIcomSignalRef		
Description	This parameter defines a reference to the signal which shall be checked additional to the message id (CanIcomMessageId). This reference is used for documentation to define which ComSignal originates this filter setting. All signals being referred by this reference shall point to the same PDU.		
Multiplicity	0..1		
Type	Reference to [ComSignal]		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	

Limitations

	Post-build time	--	
Scope / Dependency	scope: ECU dependency: The signal referenced by CanIcomSignalRef shall be included in a ComIPdu which matches with the current CAN Controller and the CAN Identifier (CanIcomMessageId) configured for this CanIcomRxMessage.		

Limitation: The reference destination of this parameter is ComSignal/ComFilter. However, this function is not used because it can replace the mask setting with the parameter of CanIcomMessageSignal.

T2MC-2630 - Can cancellation of pending L-PDUs is configurable

Title: Can cancellation of pending L-PDUs is configurable

Description: The Can module shall allow that the cancellation of pending L-PDUs with identical priority is statically configurable at pre-compile time by parameter CanIdenticalIdCancellation.()

Limitation: Support post build only

T2MC-2323 - Can defines Can_IdType

Title: Can defines Can_IdType

Description:

Name:	Can_IdType		
Type:	uint16, uint32		
Range:	Standard32Bit	--	0..0x400007FF
	Standard16Bit	--	0..0x47FF
	Extended32Bit	--	0..0xDFFFFFFF
Description:	Represents the Identifier of an L-PDU. The two most significant bits specify the frame type: 00 CAN message with Standard CAN ID 01 CAN FD frame with Standard CAN ID 10 CAN message with Extended CAN ID 11 CAN FD frame with Extended CAN ID		

()

Limitation: "uint16" and "Standard16Bit" are not supported by the TRAVEO™ T2G CAN driver.

T2MC-2415 - Can defines L-PDU-Callout API

Title: Can defines L-PDU-Callout API

Description: The L-PDU-Callout API shall be defined as:

FUNC(boolean, COM_APPL_CODE) <LPDU_CalloutName> (

uint8 Hr,

Can_IdType CanId,

uint8 CanDlc,

Limitations

```
const uint8  *CanSduPtr
```

```
)
```

where <LPDU_CalloutName> has to be substituted with the concrete L-PDU callout name which is configurable, see SWS_Can_00434_Conf.

Limitation: *The argument Hrh is Can_HwHandleType.*

T2MC-2218 - Can detects invalid state transitions

Title: Can detects invalid state transitions

Description: When the function `Can_SetControllerMode(CAN_T_START)` is entered and the CAN controller is not in state STOPPED it shall detect an invalid state transition (Compare to SWS_Can_00200). ()

Limitation: The transition STARTED to STARTED is allowed in addition, according to Figure 7-2 in specification of CAN driver AUTOSAR 4.2.2".

T2MC-2188 - Can fulfills implementation rules

Title: Can fulfills implementation rules

Description: The Can module shall fulfill all design and implementation guidelines described in [11]. (SRS_BSW_00007, SRS_BSW_00306, SRS_BSW_00308, SRS_BSW_00309, SRS_BSW_00330)

Limitation: *AUTOSAR_TR_CImplementationRules.pdf* does not exist in AUTOSAR 4.2.2. Therefore, MCAL modules shall fulfill all design and implementation guidelines as described in specification of C implementation rules *AUTOSAR_TR_CImplementationRules.pdf* in AUTOSAR 4.0.3. Out of scope: keyword macros 'CONST' and 'VAR' are not required for declaration/definition of the local variable, function parameter, and structure/union fields.

Limitations

T2MC-2175 - Can header file structure

Title: Can header file structure

Description:

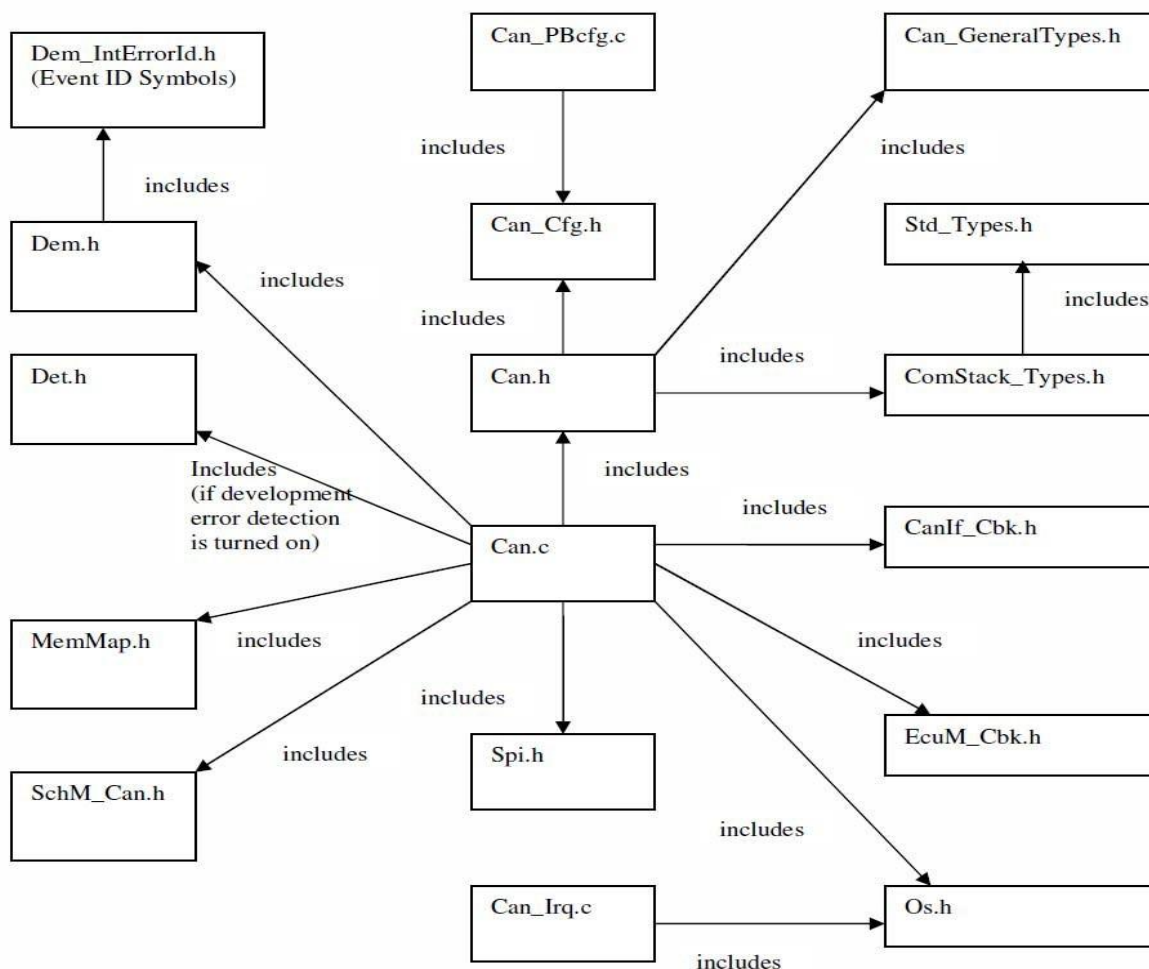


Figure 5-1: File structure for the Can module

(SRS_BSW_00381, SRS_BSW_00412, SRS_BSW_00346, SRS_BSW_00158, SRS_BSW_00435, SRS_BSW_00436, SRS_BSW_00348, SRS_BSW_00301)

Limitation: *Can.c* is a placeholder for all .c files in the *src* folder. *MemMap.h* renames to *Can_MemMap.h* and owns CAN module alone. The CAN module does not include *Spi.h*.

T2MC-2343 - Can implements API function Can_CheckBaudrate

Title: CAN implements the Can_CheckBaudrate API function.

Description:

Service name:	Can_CheckBaudrate
Syntax:	Std_ReturnType Can_CheckBaudrate(uint8 Controller, uint16 Baudrate)
Service ID[hex]:	0x0e
Sync/Async:	Synchronous

Limitations

Reentrancy:	Reentrant
Parameters (in):	Controller CAN Controller to check for the support of a certain baudrate Baudrate Baudrate to check in kbps
Parameters (inout):	None
Parameters (out):	None
Return value:	Std_ReturnType E_OK: Baudrate supported by the CAN Controller E_NOT_OK: Baudrate not supported / invalid CAN controller
Description:	This service shall check, if a certain CAN controller supports a requested baudrate Please note that this API is deprecated and is kept only for backward compatibility reasons. In the next major release this API will be deleted.

Limitation: The type of the baudrate parameter is `const uint16` to allow backward compatibility and consistency to `Can_ChangeBaudrate`.

T2MC-2546 - Can implements the CanIcomGeneral container

Title: Can implements the CanIcomGeneral container

Description:

SWS Item	ECUC_Can_00444
Container Name	CanIcomGeneral
Description	This container contains the general configuration parameters of the ICOM Configuration.
Configuration Parameters	

Limitation: All ON/OFF for pretended networking mode are controlled with `CanPublicIcomSupport` only. Therefore, `CanIcomGeneral` is not supported.

T2MC-2629 - Can implements transmit cancellation

Title: Can implements transmit cancellation

Description: The Can module shall allow that the functionality “Transmit Cancellation” is statically configurable (ON | OFF) at pre-compile time.(BSW01133)

Limitation: Support post build only

T2MC-2633 - Can initiates cancellation of L-PDU with the lowest priority

Title: Can initiates cancellation of L-PDU with the lowest priority

Description: The Can module shall initiate a cancellation of the L-PDU with the lowest priority, when all hardware transmit objects assigned by the HTH are busy and an L-PDU with a higher priority is requested to be transmitted.(BSW01133)

Limitations

Limitation: Cancellation of transmit messages is possible only for HW objects using a dedicated buffer. Cancellation for HW objects using the transmit FIFO is not supported.

T2MC-2631 - Can initiates cancellation, when an L-PDU with higher priority is requested

Title: Can initiates cancellation, when an L-PDU with higher priority is requested

Description: The Can module shall initiate a cancellation, when the hardware transmit object assigned by a HTH is busy and an L-PDU with higher priority is requested to be transmitted.(BSW01133)

Limitation: Cancellation of transmit messages is possible only for HW objects using a dedicated buffer. Cancellation for HW objects using the transmit FIFO is not supported.

T2MC-2634 - Can initiates cancellation, when L-PDU with identical priority is requested

Title: Can initiates cancellation, when L-PDU with identical priority is requested

Description: The Can module shall initiate a cancellation, when one of the hardware transmit objects assigned by the HTH is busy, an L-PDU with identical priority is requested to be transmitted and CanIdenticalIdCancellation is enabled.(BSW01133)

Limitation: Cancellation of transmit messages is possible only for HW objects using a dedicated buffer. Cancellation for HW objects using the transmit FIFO is not supported.

T2MC-2632 - Can initiates cancellation, when transmit object is busy

Title: Can initiates cancellation, when transmit object is busy

Description: The Can module shall initiate a cancellation when the hardware transmit object assigned by a HTH is busy, an L-PDU with identical priority is requested to be transmitted and CanIdenticalIdCancellation is enabled.()

Limitation: Cancellation of transmit messages is possible only for HW objects using a dedicated buffer. Cancellation for HW objects using the transmit FIFO is not supported.

T2MC-2256 - Can provides a pre-compile switch for multiplexed transmission

Title: Can provides a pre-compile switch for multiplexed transmission

Description: The Can module shall allow that the functionality “Multiplexed Transmission” is statically configurable (ON | OFF) at pre-compile time. (SRS_Can_01134)

Limitation: Support post build only

T2MC-2274 - Can raises error CAN_E_DATA_LOST in case of overwrite or overrun detected

Title: Can raises error CAN_E_DATA_LOST in case of overwrite or overrun detected

Description: If the default error detection for the Can module is enabled, the Can module shall raise the error CAN_E_DATA_LOST in case of “overwrite” or “overrun” event detection. ()

Limitation: The hardware allows detection of the message lost event only for FIFOs, not for dedicated buffers. Therefore, the filter setting for the FIFOs should be set to cover all dedicated message buffers also so that lost messages can be detected for all messages. Note that the dedicated message buffers take precedence in case of receptions; i.e., only if the dedicated message buffer is full, the message will be stored to FIFO and only if the FIFO is full, the message lost error takes effect.

Limitations

T2MC-2200 - Can supports sleep mode with state SLEEP

Title: Can supports sleep mode with state SLEEP

Description: When the CAN hardware supports sleep mode and is triggered to transition into SLEEP state, the Can module shall set the controller to the SLEEP state from which the hardware can be woken over CAN Bus. (SRS_SPAL_12067)

Limitation: The power domain and clock of the CAN controller hardware may not be disabled during sleep, otherwise wakeup is not possible. The driver does not check if the hardware is powered and clocked.

T2MC-2170 - Can uses system OsCounter to prevent endless loops

Title: Can uses system OsCounter to prevent endless loops

Description: The Can module shall use the OsCounter provided by the system service for timeout detection in case the hardware does not react in the expected time (hardware malfunction) to prevent endless loops. ()

Limitation: Endless loops are considered in function calls only. Endless loops across several main function calls are not considered.

T2MC-2471 - Can VARIANT-POST-BUILD

Title: Can VARIANT-POST-BUILD

Description: VARIANT-POST-BUILD: Mix of pre compile- and post build time configuration parameters.()

Limitation: Support post build only

Known defects

5 Known defects

The listed issues were known at the day this release note was generated. Further problems may have been discovered in the meantime. For an up-to-date list of known issues, contact your Infineon sales representative.

This release has no known issues at the time of release.

6 Documentation

All user guides for MCAL drivers are in the `\doc` subdirectory of the *installation* directory. The default location is:

`C:\INFINEON_ESDB\Tresos26_2_0\doc`

7 Technical support

If you have questions related to the driver, contact the local support application engineer.

Version history

8 Version history

8.1 Module SW-Version 1.3

Initial module setup.

8.2 Module SW-Version 1.4

T2MC-39176 - [All] Correcting vendor-specific module definition

Title: [All] Correcting vendor-specific module definition

Description: The following rules should be followed in the vendor-specific module definition.

- The multiplicity of each AUTOSAR parameter, reference and container is not correctly derived.
 - The DEFAULT-VALUE of each parameter is not valid.
 - If the target of DESTINATION-REF is not the standard AUTOSAR container, the reference should not start with '/AUTOSAR/EcucDefs/'.
-

T2MC-38100 - [CAN] By calling Can_Write in pretended networking mode, it doesn't work

Title: [CAN] By calling Can_Write in pretended networking mode, it doesn't work

Description: The expected behavior is to simply return the return value of BUSY even if Can_Write is called in "Pretended Networking Mode".

But, when CAN module calls Can_Write in "Pretended Networking Mode", it requests transmission cancellation to hardware.

It is not expected behavior to issue a transmission cancellation request.

When the CAN module requests transmission cancellation to the hardware, the transmission cancellation completion flag is set afterwards.

The transmission cancellation completion flag is cleared by transmission processing.

However, when "Pretended Networking Mode" is enabled, transmission processing will not be executed.

Therefore, since the transmission cancellation completion flag is not cleared, if the transmission interrupt is valid, a transmission cancel completion interrupt occurs indefinitely.

This problem does not occur when transmission cancellation is invalid in the configuration setting.

Also, this defect does not occur unless Can_Write is called for "Pretended Networking Mode".

T2MC-39633 - [CAN] MCDC coverage does not reach 100%

Title: [CAN] MCDC coverage does not reach 100%

Description: MCDC coverage for the following functions has not reached 100%.

Can_Write

Can_CheckBaudrateInternal

Can_Write_WriteRequest

Refer to "UT_CAN_CAN_function_call_report.html"

Version history

T2MC-39640 - [CAN] Miss acquisition of message RAM size parameter

Title: [CAN] Miss acquisition of message RAM size parameter

Description: According to "23.4 Message RAM" of TRM (No. 002-19314 Rev.*B), memory size was described as 24 KB.

On the other hand, in tresos, when acquiring CanMessageRamSize from ResourcePropertiesFile, use the parameter of MEMORY.CANxMRAM.size (x: Instance number).

In TRAVEO™ T2G-1M, "MEMORY.CAN 0 MRAM.size:0x10000" is described in ResourcePropertiesFile.

Therefore, clearly get the wrong parameter.

It is correct to get message RAM size from CANFD.x.MRAM_SIZE (x: Instance number).

Since "CANFD.0.MRAM_SIZE:24" means 24 KB, it matches the description of TRM.

T2MC-39639 - [CAN] Mistake the calculation of the number of Rx message filters

Title: [CAN] Mistake the calculation of the number of Rx message filters.

Description: The maximum value of the message filter is standard ID=128 elements, extended ID=64 elements (per CanController).

The message filter sets with CanHwFilter in tresos.

There are two defects that become a problem as follows.

1. The RxHandler dedicated to Icom invalidates the filter setting for CanHwFilter. But, editing is enabled and it is possible to add a filter..

However, setting CanHwFilter does not affect Generate.

2. Number of CanHwFilter can be set indefinitely.
-

T2MC-38099 - [CAN] Reject of BusOff requirement in "Pretended Networking Mode"

Title: [CAN] Reject of BusOff requirement in "Pretended Networking Mode"

Description: Depending on the hardware implementation, the ICOM is also able to send messages (Refer to AUTOSAR 4.2.2 section 7.10). But TRAVEO™ T2G does not implement Icom with hardware. Therefore, Icom implements it by software, there is no transmission in pretended networking mode. Since it does not transmit, bus-off never occurs. For this reason, it is necessary to delete the function to wake-up when bus-off occurs in pretended networking mode.

T2MC-39410 - [CAN] Delete CanFilterMask and CanFilterMaskRef in tresos

Title: [CAN] Delete CanFilterMask and CanFilterMaskRef in tresos

Description: CDT 305848:[TRAVEO™ T2G] CAN: usage CanHardwareObject/CanFilterMaskRef field

Delete CanFilterMask and CanFilterMaskRef by the proposal from Continental below.

Then, in case CanRxBufferSelection is CAN_RX_FIFO0/1, change CanHwFilterMask from automatic to manual input.

CDT 305848 [TRAVEO™ T2G] CAN: usage CanHardwareObject/CanFilterMaskRef field

Version history

In Tresos, the configuration schema is checking that, in a CanHardwareObject, the CanFilterMaskRef/CanFilterMaskValue is matching the first CanHwFilter/CanHwFilterMask.

From Can.xdm, this is the related test:

```
expr="(./CanObjectType = 'RECEIVE') and  
(as:ref(./CanFilterMaskValue != ../CanHwFilter/*[1]/CanHwFilterMask))"
```

```
true="CanFilterMaskRef referenced to CanFilterMaskValue must be the same as CanHwFilterMask of first  
configured CanHwFilter."/>
```

But CanFilterMaskRef/CanFilterMaskValue is completely useless and not used anywhere during the generation of .h/.c files.

I don't see the added value to enforce users to configure / update some parameters that are not used at the end.

I propose to document CanFilterMaskRef as "not used" and remove the check on its content.

T2MC-38111 - Base module in conflict with EB Base module

Title: Base module in conflict with EB Base module

Description: Base module delivered is containing files in conflict with the Base module delivered by EB.

Base module follows these rules:

- Keep the standard files in Base module
- Move controller/compiler specific headers in platform plugin

This could be achieved with moving files from CYT2 folder into platform plugin.

T2MC-38074 - File extension should be changed from .bmd to .arxml

Title: File extension should be changed from .bmd to .arxml

Description: The file extension should be changed from *.bmd to *.arxml.

Each module still has an autosar/<module>.bmd file.

Version history

8.3 Module SW-Version 1.5

T2MC-39747 - [All] Checking for valid C function name and including filename in configuration parameters

Title: [All] Checking for valid C function name and including filename in configuration parameters

Description: Checking for valid C function name:

Check all configuration parameters related to the function name to see if it is a valid C function name.

A part of parameters are not checked.

If an invalid function name is set, a compile error will occur during the build process, which is inconvenient for users.

Therefore, it is better to check whether the configured function names are valid C function names in advance (i.e. during configuration phase).

Checking for valid filename:

Check all configuration parameters related to the file name to see if it is valid.

A part of parameters cannot check the fact that empty file name (i.e. ".h") is wrong.

If an invalid file name is set, a compile error will occur during the build process, which is inconvenient for users.

Therefore, it is better, to check in advance, whether the configured file names are valid.

This CR is intended to solve the inconvenience.

T2MC-43523 - [CAN, BASE] CanObjectId does not correspond to a value of 256 or more

Title: [CAN, BASE] CanObjectId does not correspond to a value of 256 or more

Description: The RxHwHandle member of the Can_RxHandleMappingType table that stores CanObjectId is uint8.

When a value of 256 or more is entered, overflow occurs.

T2MC-43939 - [CAN] Support TRAVEO™ T2G-B-H-8M

Title: [CAN] Support TRAVEO™ T2G-B-H-8M

Description: Add 8M series document to "Hardware Documentation" section on user guide.

T2MC-41850 - [General] <CODE-DESCRIPTORS> Node should be added to the *arxml* files of all modules

Title: [General] <CODE-DESCRIPTORS> Node should be added to the *arxml* files of all modules

Description: For all modules, the <CODE-DESCRIPTORS> Node needs to be added for the RTE within the BSWMD *arxml* file.

Version history

8.4 Module SW-Version 1.6

T2MC-50714 - [CAN] BaudRate duplicated error can't detected (CanChangeBaudrate=true)

Title: [CAN] BaudRate duplicated error can't detected (CanChangeBaudrate=true)

Description: In Tresos, if CanChangeBaudrateApi is valid, unless the CanControllerBaudRate is unique in each CanController, it will cause an error.

However, this error cannot be detected.

T2MC-50717 - [CAN] CanId and mask of lcom are not subject to CanIdType restriction.

Title: [CAN] CanId and mask of lcom are not subject to CanIdType restriction.

Description: In Tresos, CanlcomMessageld and CanlcomMessageldMask can set 0x20000000 as the maximum irrespective of CanID regardless of STANDARD / EXTENDED.

The maximum value of CanlcomMessageld and CanlcomMessageldMask must be set to the maximum value according to the setting of CanIdType in CanHardwareObject

That is, the maximum value must be set according to the CanIdType condition as follows.

- CanIdType=STANDARD: CanlcomMessageld=0x0 to 0x7FF, CanlcomMessageldMask=0x0 to 0x7FF
- CanIdType=EXTENDED: CanlcomMessageld=0x0 to 0x1FFFFFFF, CanlcomMessageldMask=0x0 to 0x1FFFFFFF

T2MC-50713 - [CAN] CanMessageRamSize allows input of negative values

Title: [CAN] CanMessageRamSize allows input of negative values

Description: Even if a negative value is input to CanMessageRamSize in Tresos, the expected error message is not output.

The contents of the Can.xdm file are shown below.

```
<a:tst expr="../../CanHardwareObject/*" true="Negative values are not allowed for
CanMessageRamSize."/>
```

This code checks whether at least one handler is defined in the container of CanHardwareObject and it has nothing to do with checking whether it is a negative value or not.

T2MC-50716 - [CAN] Expected error is not output when CanlcomRxMessageDedicated=true

Title: [CAN] Expected error is not output when CanlcomRxMessageDedicated=true

Description: In Tresos, the following parameter condition in CanHardwareObject is an error.

CanlcomRxMessageDedicated=true

CanIdValue is enabled

At this time, the error message output as the parameter of CanIdValue is different from the expected message.

Expect message

If CanlcomRxMessageDedicated is enabled, CanIdValue must be invalidated.

Version history

Actual message

Invalid XPath-expression for attribute "INVALID" of node "APath": (2085) Cannot parse value "" to an int-value

T2MC-56133 - [CAN] Inconsistency of Can.arxml and Can.xdm

Title: [CAN] Inconsistency of Can.arxml and Can.xdm

Description: Several inconsistencies between .arxml and .xdm were detected in the unit test.

.arxml and .xdm should be consistent.

- Can.xdm

1) CanControllerBaseAddress

There are not define of maximum value and minimum value.

2) CanControllerId

There are not define of maximum value and minimum value.

3) CanMessageRamBaseAddress

There are not define of maximum value and minimum value.

4) CanMessageRamSize

There is not define of maximum value.

5) CanControllerBaudRate

There is not define of maximum value.

Minimum value is different.

6) CanObjectId

Maximum value is different.

7) CanIdValue

Maximum value is different.

8) CanHwFilterCode

Maximum value is different.

9) CanHwFilterMask

Maximum value is different.

T2MC-50715 - [CAN] RxFIFO0 exceeds value of 64 and Config file can be generated

Title: [CAN] RxFIFO0 exceeds value of 64 and Config file can be generated

Description: In Tresos, setting RxFIFO0 for Icom in the condition that the total value of Rx FIFO0 of each CanController is 64 does not cause an error.

In the error check of CanHwObjectCount, RxFIFO0 for Icom is not added to the total number of RxFIFO0.

Version history

Therefore, even if the total value of Rx FIFO0 exceeds 64, a configuration file is generated without causing an error.

T2MC-50612 - [General] Delete device-dependent information from the user guide

Title: [General] Delete device-dependent information from the user guide

Description: Any device-dependent information should not be included in the user guide.

Therefore, delete the datasheet name from the related documentation in the user guide.

T2MC-50519 - [General] Export issue with MCAL ES10_20180308

Title: [General] Export issue with MCAL ES10_20180308

Description: An example of the issue is described below.

The configuration exported from Tresos does not correspond to the real configuration shown in Tresos. See the attached example.

The issue concerns other modules too, not only the port described in attached pdf file.

8.5 Module SW-Version 1.7

T2MC-59571 - [CAN, BASE] Can_HwHandleType supports both uint8 and uint16

Title: [CAN, BASE] Can_HwHandleType supports both uint8 and uint16

Description: Can_HwHandleType must support both uint8 and uint16.

Can_GeneralTypes.h is provided by the integrator and must be compatible with MCAL CAN and BSW CAN (CANIF, PDUR etc).

T2MC-59650 - [CAN] Warning message with AMDC 1.0.17

Title: [CAN] Warning message with AMDC 1.0.17

Description: The following warning message was displayed after updating AMDC to version 1.0.17.

Warning;Can_Merged.xml;nobody;Rule A205: Parameter 'Can/CanGeneral/CanIncludeFile' has no 'ECUC-MULTIPLICITY-CONFIGURATION-CLASS' elements.;

Version history

8.6 Module SW-Version 1.8

T2MC-67027 - [CAN] CAN driver can't detect error frame in pretended mode

Title: [CAN] CAN driver can't detect error frame in pretended mode

Description: In pretended networking mode, CAN driver cannot wakeup when detecting an error frame on the CAN bus.

(Only when configuration of CanIcomPayloadLengthError is valid)

T2MC-72714 - [CAN] CanController configuration generated from tresos points to wrong other configuration data

Title: [CAN] CanController configuration generated from tresos points to wrong other configuration data

Description: The CAN module creates the CanController list in Tresos in the following order. If a config file is generated, it causes defects [1] and [2].

Name CanControllerIdCanControllerPhysicalChannel

CntL_00 CANFD01

CntL_11 CANFD00

[1] The pointer member of the Can_ControllerConfigType table points to an unintended position.

CanControllerId: ascending order

CanControllerPhysicalChannel: descending order

The following is a pointer member of Can_ControllerConfigType that is a defect.

BaudratesCfgPtr, DefaultBaudratePtr, RxBuffersFiltersPtr, RxFifo0FiltersPtr, RxFifo1FiltersPtr

[2] When an interrupt occurs and Can_InterruptHandler is called from Can_Interrupt_CANFDxx_Cat1 or Can_Interrupt_CANFDxx_Cat2, the configuration data of CanController in which the interrupt is not generated is used.

WorkAround

Set CanControllerId and CanControllerPhysicalChannel in CanController in ascending order.

Name CanControllerIdCanControllerPhysicalChannel

CntL_00 CANFD00

CntL_11 CANFD01

T2MC-67324 - [CAN] Config variant can select VariantPreCompile in tresos

Title: [CAN] Config variant can select VariantPreCompile in tresos

Description: According to the following requirements, only Post-Build needs to be supported, not Pre-Compile.

Version history

T2MC-10476 - MCAL_Gen_14: Support PB config (in SWRSCUSTOM)

Description: Each MCAL modules shall support the post-build selection as defined by the AUTOSAR module specification.

However, Config variant in tresos can select "VariantPreCompile".

T2MC-67018 - [CAN] Defect in the range check of CANFD's data prescaler

Title: [CAN] Defect in the range check of CANFD's data prescaler

Description: The range check of CANFD's data prescaler is different from the check when generating Can_PBcfg.h from tresos and the check in the Can_SetBaudrateInChangedClock () API.

- Range check of data prescaler when generating Can_PBcfg.h from tresos:

Check range: $1 \leq \text{DataPrescaler} \leq 32$

- Range check of data prescaler in Can_SetBaudrateInChangedClock() API:

Check range: $1 \leq \text{DataPrescaler} \leq 2$

The range check of data prescaler in Can_SetBaudrateInChangedClock () was implemented according to ISO 11898-1. However, according to ISO 11898-1, data prescaler must set to 1 or 2 when transmitter delay compensation is enabled. That is, both tresos and Can_SetBaudrateInChangedClock must be checked for the following range conforming to ISO-11898-1.

- Transmitter delay compensation is enabled: $1 \leq \text{DataPrescaler} \leq 2$
 - Transmitter delay compensation is disabled: $1 \leq \text{DataPrescaler} \leq 32$
-

T2MC-65907 - [CAN] If CanIf_TriggerTransmit returns NOT_OK, unintended DET occurs

Title: [CAN] If CanIf_TriggerTransmit returns NOT_OK, unintended DET occurs

Description: When the return value of CanIf_TriggerTransmit is E_NOT_OK, a DET error of CAN_E_PARAM_POINTER occurs.

However, after that, a DET error of CAN_E_PARAM_DLC will occur unintentionally

T2MC-65906 - [CAN] Received Msg stored in RxFIFO may not be transferred to CanIf

Title: [CAN] Received Msg stored in RxFIFO may not be transferred to CanIf

Description: When RxFIFO0/1 for normal mode and RxFIFO0 for pretended networking mode are mixed in the configuration setting,

the following unexpected operation is performed with RxFIFO0/1.

CAN module transfers the wrong RxHandler to the CanIf_RxIndication

CAN module does not call CanIf_RxIndication

T2MC-65908 - [CAN] Some definitions are missing in BSWMD file

Title: [CAN] Some definitions are missing in BSWMD file

Description: The following APIs do not implement exclusive control. However, CAN-ENTER-EXCLUSIVE-AREA-REFS is defined for these APIs in the BSWBD file.

Version history

Can_GetVersionInfo

Can_CheckWakeup

Can_MainFunction_Read

Can_MainFunction_Write

T2MC-77594 - Support IAR compiler

Title: Support IAR compiler

Description: Support IAR compiler (IAR EWARM FS 8.22.3.15992).

8.7 Module SW-Version 1.9

T2MC-91585 - [CAN] *Can.xdm* is inconsistent with *Can.arxml*

Title: [CAN] *Can.xdm* is inconsistent with *Can.arxml*

Description: There are some inconsistencies between *Can.xdm* and *Can.arxml* in the following definitions:

- The CanFdPaddingValue, CanMainFunctionRWPeriodRef parameters in *Can.arxml* must have POST-BUILD-VARIANT-MULTIPLICITY set to true.
 - The LOWER-MULTIPLICITY=0 and MULTIPLICITY-CONFIG-CLASSES parameters in *Can.arxml* must define CanWakeupSourceRef, CanFdPaddingValue, and CanMainFunctionRWPeriodRef.
 - - The CanRxBufferSelection parameter must have LOWER-MULTIPLICITY=0 because there is an "ENABLE" tag in the *xdm* file (*Can.arxml* and *Can.xdm* files need modification).
 - For the CanHwFilter parameter is UPPER-MULTIPLICITY is greater than LOWER-MULTIPLICITY, so POST-BUILD-VARIANT-MULTIPLICITY must be true (*Can.arxml* and *Can.xdm* files need modification).
 - The CanSupportTTCANRef parameter does not require "Multiplicity Configuration Class" in *Can.xdm*.
 - The CanMainFunctionPeriod parameter in *Can.arxml* must define MULTIPLICITY-CONFIG-CLASSES.
-

T2MC-90691 - [CAN] Insufficient description of limitations in release notes

Title: [CAN] Insufficient description of limitations in release notes

Description: The following requirements are limited in function, but are not described in the limitation section of the release note.

T2MC-2546 - Can implements container CanIcomGeneral

T2MC-2547 - Can - CanIcomGeneral contains CanIcomLevel

T2MC-2548 - Can - CanIcomGeneral contains CanIcomVariant

Version history

8.8 Module SW-Version 1.10

T2MC-97131 - Different macros are used for setting and checking the value

Title: Different macros are used for setting and checking the value

Description: Some modules differ in the macro names defined and the macro names used in the processing.

For example, when the macro set to TRUE is judged as STD_ON, the definition value is 1 for both, but the same macro must be used.

```
#define MACRO_DEFINE (TRUE)
-
#if MACRO_DEFINE == STD_ON
xxx
#endif
```

In Platform_Types.h of the base module

```
#define TRUE 1U
#define FALSE 0U
```

In Std_Types.h of the base module

```
#define STD_ON 0x01U
#define STD_OFF 0x00U
```

T2MC-97382 - Macro definition at variable declaration is missing and the limitation is not mentioned in release notes

Title: Macro definition at variable declaration is missing and the limitation is not mentioned in release notes

Description: Macro definitions are not used when declaring some variables and pointers (in FLS, MCU, PORT, SPI, and WDG).

According to AUTOSAR specification:

[SWS_COMPILER_00026]

```
#define VAR(vartype, memclass)
```

True:

```
volatile P2VAR(Spi_DmaChannelRegsType, AUTOMATIC, REGSPACE) retPtr;
```

False:

```
volatile Spi_DmaChannelRegsType * retPtr;
```

This issue is present in the following cases:

- All types of pointer declaration/definition are defined without macros.
These contain the function parameter/global variable/local variable/structure field/union field.
- All types of function declaration/definition are defined without macros.
- When there is nested macro usage in function macros.
- Raw pointer is used in the function macro:
e.g., FUNC(int *, memclass) function(void);
- Global variable or static variable in the function is not defined with macros.

To fully comply with the above cases, change variable and function definitions in FLS, MCU, PORT, SPI, and WDG.

Version history

In requirements, keyword macros 'CONST' and 'VAR' are not required for declaration/definition of the local variable, function parameter, and structure/union fields.
The information must be described in all release notes.

T2MC-39519 - Support EB tresos V26.2.0

Title: Support EB tresos V26.2.0

Description: Support EB tresos V26.2.0

[Impact]

Strict AUTOSAR specification and check for parameter configuration errors are implemented in EB tresos V26.2.0.

In addition, handling of reference paths (relative paths) such as system description file (ARXML) is changed in EB tresos V26.2.0.

Therefore, if the current ECUC configuration definitions XML file contains deviations or errors, you may find errors during import to tresos26. In that case, the ECUC configuration definitions XML file must be modified appropriately.

In addition, if the current ARXML file contains unresolvable paths, you may find errors during import to tresos26. In that case, ARXML file must be modified.

The SW has been tested; no risks except for the low-level risk listed above were found.

8.9 Module SW-Version 1.11

T2MC-164778 - Support MISRA C:2012 coding rule

Title: Support MISRA C:2012 coding rule

Description: Support MISRA C:2012 coding rule.

The MISRA C:2012 coding rule checks the source code.

If a deviation from the rules is required, add the deviation comment to the code and report the result.

If a deviation is for MISRA-C:2004 only, remove the deviation comment.

8.10 Module SW-Version 1.12

T2MC-164831 - [ALL] Misleading comment in Module_MemMap.h

Title: [ALL] Misleading comment in Module_MemMap.h

Description: *{Mip}_MemMap.h* files are provided as sample template files. But, the file header comment cannot be modified, which is a contradiction. To resolve this contradiction, change the file header comment to allow user modification.

Also, to make sure that the file is not a part of the commercial product, move the *{Mip}_MemMap.h* files to the *MemMap* stub folder.

Version history

8.11 Module SW-Version 1.13

T2MC-170662 - [CAN] Some parameters are inconsistent between XDM and ARXM

Title: [CAN] Some parameters are inconsistent between XDM and ARXM

Description: When the value set in CanControllerInstance is generated, a warning message is output and the set value is not applied.

[impact]

Change CanControllerInstance to the following value in Can.xdm and Can.arxml.

0 -> INSTANCE0, 1-> INSTANCE1

If an existing (old) project is applied for this version, you will find some errors in the "Problems View" window during import to EB tresos. In that case, modify the CanControllerInstance parameter (0-> INSTANCE0, 1-> INSTANCE1).

T2MC-170537 - [CAN] Unnecessary "extern" definition in Can_RegisterAbstraction.c

Title: [CAN] Unnecessary "extern" definition in Can_RegisterAbstraction.c

Description: "Extern" is defined in the Can_RaSetNormalMode () function in Can_RegisterAbstraction.c. This definition is verbose and should be removed.

Following is supported in release V1.10.0.

T2MC-172517 - Add a description on DeepSleep in the user guide

Title: Add a description on DeepSleep in the user guide

Description: Add a note on DeepSleep mode in the user guide.

8.12 Module SW-Version 1.14

T2MC-179497 - [CAN] TX cancel does not work due to CAN controller mode state change

Title: [CAN] TX cancel does not work due to CAN controller mode state change

Description: If CAN TX cancellation is enabled and the CAN controller mode is transitioned as follows, the CAN TX cancellation cannot be performed; the `Can_Write()` API always returns busy.

The state of the CAN controller transitions from CAN_T_START to CAN_T_STOP and then to CAN_T_START again.

T2MC-179195 - [CAN] State mismatch can occur between software and hardware if an ECC error is detected in SLEEP state

Title: [CAN] State mismatch can occur between software and hardware if an ECC error is detected in SLEEP state

Description: In extremely rare cases, if an ECC error is detected in SLEEP state, a state mismatch can occur between software and hardware, where hardware=STOPPED and software=SLEEP. When this issue occurs, the state will not match; CAN communication will not be possible. This defect cannot be resolved by fixing it according to [T2MC-178860](#).

Version history

T2MC-178861 - [CAN] Improved performance with Xpath fixes in xdm files

Title: [CAN] Improved performance with Xpath fixes in xdm files

Description: In Can.xdm, the following code changes can improve the performance of EB tresos Studio.

[Before] `//CanGeneral/CanChangeBaudrateApi=...`

[After] `../(omit)/CanGeneral/CanChangeBaudrateApi=... (Relative path)`

If specified with a double slash in XPath, all element nodes will be searched from the descendant nodes of the root node. By not using double slashes, search time is reduced and performance is improved.

T2MC-179485 - [CAN] Memory allocation to unintended sections

Title: [CAN] Memory allocation to unintended sections

Description: For the `Can_ConfigSetPtr` static pointer variable in `Can_Data.c`, the type definition of the memory section for this variable was not correct: instead of `VAR_INIT`, it was `CONST`. Therefore, this variable has a risk of unexpected memory allocation.

[Current definition]

```
#define CAN_START_SEC_CONST_ASIL_B_UNSPECIFIED
static P2CONST(Can_ConfigType, AUTOMATIC, AUTOMATIC) Can_ConfigSetPtr =
NULL_PTR;
#define CAN_END_SEC_CONST_ASIL_B_UNSPECIFIED
```

[Correct definition]

```
#define CAN_START_SEC_VAR_INIT_ASIL_B_UNSPECIFIED
static P2CONST(Can_ConfigType, AUTOMATIC, AUTOMATIC) Can_ConfigSetPtr =
NULL_PTR;
#define CAN_END_SEC_VAR_INIT_ASIL_B_UNSPECIFIED
```

T2MC-178860 - [CAN] In very rare cases, CAN communication is not possible due to state transition conflict

Title: [CAN] In very rare cases, CAN communication is not possible due to state transition conflict

Description: In very rare cases, when multiple state transitions interfere with each other, the transition states of software and hardware will not match; CAN communication will not be possible from that point on. When this happens, the entire CAN driver must be reset by using the `Can_DeInit()` API.

There is a very short time during which the interrupts remain enabled in the `Can_MainFunction_Mode()` API. The state conflict occurs if an event occurs during this short interval.

This issue occurs in the following conditions:

1. **Conflict between the `Can_MainFunction_Mode()` API and the BusOff event:** If the BusOff event occurs within this API, a transition state mismatch will occur. When this occurs, CAN communication is not be possible.
 2. **Conflict between the `Can_MainFunction_Mode()` API and the Wakeup event:** If a Wakeup event occurs in this API, a transition state mismatch will occur. When this occurs, CAN drive cannot wake from Sleep.
 3. **Conflict between the `Can_MainFunction_Mode()` API and an ECC error event:** If an ECC error event occurs within this API, a transition state mismatch will occur. When this occurs, CAN communication will not be possible.
-

Version history

T2MC-179196 - [CAN] Use the same definitions for the const pointer of the argument in all functions

Title: [CAN] Use the same definitions for the const pointer of the argument in all functions

Description: Only the following functions have the const pointer definition of the argument differ from other functions. Therefore, the same definitions for the const pointer of the argument should be used.

```
FUNC(void, AUTOMATIC) Can_TxSetTxCancelRequest
( CONST(Can_ControllerConfigType*, AUTOMATIC) ControllerCfgPtr,
:
)
```

other module

e.g.

```
FUNC(void, AUTOMATIC) Can_TxWrite
(
P2CONST(Can_ControllerConfigType, AUTOMATIC, AUTOMATIC) ControllerCfgPtr,
:
)
```

T2MC-178688 - Addition of the notice for Arm® errata and workaround in the user guide

Title: Addition of the notice for Arm® errata and workaround in the user guide

Description: Add a notice for Arm® Cortex®-M4 errata 838869 and software workaround in the user guide.

8.13 Module SW-Version 1.15

T2MC-179766 - [CAN] Out-of-range access for 4-ch CAN FD unit during Init when using the last channel

Title: [CAN] Out-of-range access for 4-ch CAN FD unit during Init when using the last channel

Description: In *Can_RegisterAbstraction.c*, the maximum number of "CH" members in the *stc_CAN_t* structure is 3.

```
typedef struct stc_CAN{
    stc_CAN_CH_t CH[3];
    uint32 au32Reserved[640];
:
} stc_CAN_t;
```

Out-of-range access will occur for variants equipped with four or more CAN channels. However, in this case, the reserved area of "au32Reserved" is accessed, so it does not cause a fatal exception error. It does not cause any functional problems, but the "CH" member range should be extended from the viewpoint of source code quality.

8.14 Module SW-Version 1.16

T2MC-181017 - [CAN] Avoid the possible state mismatch in the CAN driver

Title: T2MC-181017 - [CAN] Avoid the possible state mismatch in the CAN driver

Description: When *Can_SetControllerMode(CAN_T_WAKEUP)* is called while the CAN driver is in the STOPPED state, the driver returns a State mismatch condition via Error notification.

Version history

This is because LCY CAN driver assumes that `Can_SetControllerMode(CAN_T_WAKEUP)` is called only in its SLEEP state. On the other hand, the Vector CAN driver allows this function call in STOP mode.

The mismatch condition occurs if the state is changed from both sides (from UserInt and EcuM). This mismatch can be prevented by keeping the SLEEP state of the CAN driver or removing one of the sides.

T2MC-183050 - [CAN] Read data corruption may occur if RX is in polling mode and TX is in interrupt mode

Title: T2MC-183050 - [CAN] Read data corruption may occur if RX is in polling mode and TX is in interrupt mode

Description: When the CanController configuration settings are as follows, the read data may be corrupted under limited timing conditions.

- CanRxProcessing: Polling mode
- CanTxProcessing: Interrupt mode

If a TX interrupt occurs while `Can_MainFunction_Read()` is running, the interrupt handler may process the RX read operation incorrectly. The interrupt handler must have guard processing to handle the data according to the mode.

8.15 Module SW-Version 1.17

T2MC-183982 - [CAN] Improvement of interrupt handler to prevent spurious interrupts

Title: [CAN] Improvement of interrupt handler to prevent spurious interrupts

Description: INTR is recommended to be read back to ensure the update in the register immediately. (This recommendation is from application note AN224432 *E, which was recently updated.) This is because there is a possibility that a spurious interrupt occurs when leaving an interrupt handler after clearing the INTR without register readback.

Currently, there is no INTR readback operation in the CAN interrupt category 2 handler. If a spurious interrupt occurs, the CAN module ignores this interrupt and exits.

This can be resolved by adding a readback operation to ensure that the write buffer is drained.

T2MC-183983 - Update copyright notice and disclaimer statement

Title: Update copyright notice and disclaimer statement

Description: Copyright notice and disclaimer statement in the file header comment are updated to follow the up-to-date specifications.

Trademarks

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

Edition 2022-06-24

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2022 Infineon Technologies AG.

All Rights Reserved.

Do you have a question about this document?

Go to www.infineon.com/support

Document reference

002-23389 Rev. *Q

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffheitsgarantie").

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.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

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.