

TRAVEO™ T2G family AUTOSAR MCAL ADC release notes

SRN223354 version 1.14

About this document

Scope and purpose

Thank you for your interest in the TRAVEO™ T2G family AUTOSAR MCAL ADC driver version 1.14. This document lists the installation requirements, software changes, limitations, and known issues.

Intended audience

This document is intended for anyone who uses the analog digital converter (ADC) 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	7
1.6 Note on "_Bswmd.arxml"	8
1.7 Release details.....	8
2 Installation	9
3 Deviations from AUTOSAR.....	10
4 Limitations	18
5 Known defects	41
6 Documentation	42
7 Technical support.....	43
8 Version history	44
8.1 Module SW-Version 1.3.....	44
8.2 Module SW-Version 1.4.....	44
8.3 Module SW-Version 1.5.....	46
8.4 Module SW-Version 1.6.....	49
8.5 Module SW-Version 1.7.....	50
8.6 Module SW-Version 1.8.....	50
8.7 Module SW-Version 1.9.....	51
8.8 Module SW-Version 1.10.....	52

Table of contents

8.9	Module SW-Version 1.11.....	53
8.10	Module SW-Version 1.12.....	53
8.11	Module SW-Version 1.13.....	54
8.12	Module SW-Version 1.14.....	54

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	<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>

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	-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

1.4 Memory consumption

GHS (Adc_lib) section	Size (in bytes)
.text	4528
.bss	1
Combined	4529

GHS (Adc_src) section	Size (in bytes)
.text	4414
.bss	220
.rodata	1036
Combined	5670

System requirements and recommendations

IAR (Adc_lib) section	Size (in bytes)
.text	3804
.bss	1
Combined	3805

IAR (Adc_src) section	Size (in bytes)
.text	3944
.bss	220
.rodata	1036
Combined	5200

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)
Adc_Init	100
Adc_DeInit	80
Adc_StartGroupConversion	64
Adc_StopGroupConversion	52
Adc_ReadGroup	96
Adc_EnableHardwareTrigger	52
Adc_DisableHardwareTrigger	52
Adc_EnableGroupNotification	8
Adc_DisableGroupNotification	8
Adc_GetGroupStatus	28
Adc_GetVersionInfo	4
Adc_GetStreamLastPointer	92
Adc_SetupResultBuffer	32
Adc_ChangeSamplingTime	44
Adc_GetCurrentPowerState	16

System requirements and recommendations

Function	Max stack usage (in bytes)
Adc_GetTargetPowerState	12
Adc_Main_PowerTransitionManager	0
Adc_PreparePowerState	16
Adc_SetPowerState	52
Adc_ChangeCalibrationChannel	36
Adc_SetCalibrationValue	40
Adc_GetCalibrationAlternateValue	36
Adc_GetCalibrationValue	36
Adc_DmaDone_247_Cat1	84
Adc_DmaDone_247_Cat2	84
Adc_DmaDone_240_Cat1	84
Adc_DmaDone_240_Cat2	84
Adc_IsrConversionDone_143_Cat1	84
Adc_IsrConversionDone_143_Cat2	84
Adc_DmaDone_248_Cat1	84
Adc_DmaDone_248_Cat2	84
Adc_DmaDone_245_Cat1	84
Adc_DmaDone_245_Cat2	84

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.*

System requirements and recommendations

1.5.2 IAR Embedded Workbench

Function	Max stack usage (in bytes)
Adc_Init	80
Adc_DeInit	64
Adc_StartGroupConversion	64
Adc_StopGroupConversion	64
Adc_ReadGroup	48
Adc_EnableHardwareTrigger	64
Adc_DisableHardwareTrigger	64
Adc_EnableGroupNotification	8
Adc_DisableGroupNotification	8
Adc_GetGroupStatus	32
Adc_GetVersionInfo	8
Adc_GetStreamLastPointer	56
Adc_SetupResultBuffer	32
Adc_ChangeSamplingTime	40
Adc_GetCurrentPowerState	24
Adc_GetTargetPowerState	16
Adc_Main_PowerTransitionManager	0
Adc_PreparePowerState	24
Adc_SetPowerState	40
Adc_ChangeCalibrationChannel	32
Adc_SetCalibrationValue	44
Adc_GetCalibrationAlternateValue	44
Adc_GetCalibrationValue	44
Adc_DmaDone_247_Cat1	32
Adc_DmaDone_247_Cat2	32
Adc_DmaDone_240_Cat1	32
Adc_DmaDone_240_Cat2	32
Adc_IsrConversionDone_143_Cat1	32
Adc_IsrConversionDone_143_Cat2	32
Adc_DmaDone_248_Cat1	32
Adc_DmaDone_248_Cat2	32
Adc_DmaDone_245_Cat1	32
Adc_DmaDone_245_Cat2	32

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.

System requirements and recommendations

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

1.14.x

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

AUTOSAR specification version (ASR)

4.2.2

Target

MXS40

MCAL configuration settings	Supported derivatives
See the resource release notes	See the resource release notes

Corresponding Adc_MemMap.h stub file version

1.0.1

Installation

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-11015 - [SWS_Adc_00315] General behavior: Static configuration option for priority mechanism

Title: [SWS_Adc_00315] General behavior: Static configuration option for priority mechanism

Description: [SWS_Adc_00315] [The ADC module shall support the static configuration option to disable the priority mechanism.] ()

Reason for rejection: Hardware priority mechanism cannot be disabled. It is always ON in hardware.

T2MC-11214 - [SWS_Adc_00321] Function definitions: `Adc_EnableHardwareTrigger` behavior: Raise the error `ADC_E_BUSY` (priority mechanism is disabled and queuing disabled, not in state `ADC_IDLE`)

Title: [SWS_Adc_00321] Function definitions: `Adc_EnableHardwareTrigger` behavior: Raise the error `ADC_E_BUSY` (priority mechanism is disabled and queuing disabled, not in state `ADC_IDLE`)

Description: [SWS_Adc_00321] [If development error detection is enabled for the ADC driver and if the priority mechanism is disabled and queuing disabled: when called while any group with trigger source SW is not in state `ADC_IDLE`, the function `Adc_EnableHardwareTrigger` shall raise development error `ADC_E_BUSY` and return without any action.] ()

Reason for rejection: The priority mechanism is implemented in hardware and cannot be disabled; therefore, this requirement does not apply.

T2MC-11021 - [SWS_Adc_00333] General behavior: Channel group conversion order when priority mechanism is not active

Title: [SWS_Adc_00333] General behavior: Channel group conversion order when priority mechanism is not active

Description: [SWS_Adc_00333] [If the priority mechanism is not active and if the static configuration parameter `AdcEnableQueuing` is set to ON, the ADC module shall support a queuing of conversion requests and shall service the software groups in a 'first come first served' order.] ()

Reason for rejection: Hardware queuing is always ON and is more efficient than software queuing; therefore, software queuing is not required. In addition, "first come first served" ordering is not possible in only hardware.

T2MC-11022 - [SWS_Adc_00335] General behavior: Software conversion request per channel group in the software queue

Title: [SWS_Adc_00335] General behavior: Software conversion request per channel group in the software queue

Description: [SWS_Adc_00335] [If the queuing mechanism is active (priority mechanism active or queuing explicitly activated), the ADC module shall store each software conversion request per channel group at most one time in the software queue.] ()

Reason for rejection: Hardware queuing is always ON and is more efficient than software queuing; therefore, software queuing is not required.

Deviations from AUTOSAR

T2MC-11016 - [SWS_Adc_00340] General behavior: Static configuration option for priority mechanism ADC_PRIORITY_HW_SW, using both hardware and software prioritization mechanism

Title: [SWS_Adc_00340] General behavior: Static configuration option for priority mechanism ADC_PRIORITY_HW_SW, using both hardware and software prioritization mechanism

Description: [SWS_Adc_00340] [The ADC module shall support the static configuration option to enable the priority mechanism ADC_PRIORITY_HW_SW, using both hardware and software prioritization mechanism. If the hardware does not provide the hardware prioritization mechanism a pure software prioritization mechanism shall be implemented.] (SRS_Adc_12820)

Reason for rejection: The priority mechanism is implemented in HW with better performance than SW implementation. Therefore, no additional software implementation is available.

T2MC-11134 - [SWS_Adc_00342] Function definitions: Adc_Init behavior: Initialize the ADC hardware units and driver in case of variant PC

Title: [SWS_Adc_00342] Function definitions: Adc_Init behavior: Initialize the ADC hardware units and driver in case of variant PC

Description: [SWS_Adc_00342] [In case of Variant PC: The function Adc_Init shall initialize the ADC hardware units and driver according to the pre-compile configuration set. The configuration pointer which is passed to Adc_Init shall be a NULL pointer. The pointer is only evaluated, if development error detection is enabled (see SWS_Adc_00344).] (SRS_BSW_00345, SRS_BSW_00414)

Reason for rejection: Only variant PB is supported.

T2MC-11144 - [SWS_Adc_00344] Function definitions: Adc_Init behavior: Raise the error ADC_E_PARAM_CONFIG in case of variant PC

Title: [SWS_Adc_00344] Function definitions: Adc_Init behavior: Raise the error ADC_E_PARAM_CONFIG in case of variant PC

Description: [SWS_Adc_00344] [In case of Variant PC and if development error detection for the ADC module is enabled: if called without a NULL_PTR as configuration parameter, the function Adc_Init shall raise development error ADC_E_PARAM_CONFIG and return without any action.] ()

Reason for rejection: Only variant PB is supported.

T2MC-11172 - [SWS_Adc_00346] Function definitions: Adc_StartGroupConversion behavior: Raise the error ADC_E_BUSY(priority mechanism disabled, queuing disabled, groups cannot be implicitly stopped, not in state ADC_IDLE)

Title: [SWS_Adc_00346] Function definitions: Adc_StartGroupConversion behavior: Raise the error ADC_E_BUSY(priority mechanism disabled, queuing disabled, groups cannot be implicitly stopped, not in state ADC_IDLE)

Description: [SWS_Adc_00346] [If development error detection for the ADC module is enabled and the priority mechanism is disabled and the queuing is disabled : when called while any of the groups, which cannot be implicitly stopped, is not in state ADC_IDLE , the function Adc_StartGroupConversion shall raise development error ADC_E_BUSY and return without any action.] ()

Reason for rejection: The priority mechanism is implemented in hardware and cannot be disabled; therefore, this requirement does not apply.

Deviations from AUTOSAR

T2MC-11176 - [SWS_Adc_00351] Function definitions: `Adc_StartGroupConversion` behavior: Raise the error `ADC_E_BUSY` (priority mechanism disabled, queuing enabled, groups cannot be implicitly stopped, not in state `ADC_IDLE`)

Title: [SWS_Adc_00351] Function definitions: `Adc_StartGroupConversion` behavior: Raise the error `ADC_E_BUSY` (priority mechanism disabled, queuing enabled, groups cannot be implicitly stopped, not in state `ADC_IDLE`)

Description: [SWS_Adc_00351] [If development error detection for the ADC module is enabled and the priority mechanism is disabled and the queuing is enabled: when called while a group, which cannot be implicitly stopped, is not in state `ADC_IDLE`, the function `Adc_StartGroupConversion` shall raise development error `ADC_E_BUSY` and return without any action.] ()

Reason for rejection: The priority mechanism is implemented in hardware and cannot be disabled; therefore, this requirement does not apply.

T2MC-11216 - [SWS_Adc_00353] Function definitions: `Adc_EnableHardwareTrigger` behavior: Raise the error `ADC_E_BUSY` (maximum number of available hardware triggers is already enabled)

Title: [SWS_Adc_00353] Function definitions: `Adc_EnableHardwareTrigger` behavior: Raise the error `ADC_E_BUSY` (maximum number of available hardware triggers is already enabled)

Description: [SWS_Adc_00353] [If development error detection for the ADC module is enabled: if the maximum number of available hardware triggers is already enabled (device and implementation specific), the function `Adc_EnableHardwareTrigger` shall raise development error `ADC_E_BUSY` and return without any action.] ()

Reason for rejection: It is possible to start all configured groups in the hardware.

T2MC-11020 - [SWS_Adc_00417] General behavior: Channel group conversion order with the same priority level

Title: [SWS_Adc_00417] General behavior: Channel group conversion order with the same priority level

Description: [SWS_Adc_00417] [If the priority mechanism is active, the ADC module shall handle channel group conversion requests for groups with the same priority level, in a 'first come first served' order.] ()

Reason for rejection: If the same priority level is set for more than one channel, the one with a lower channel index will be given a higher priority among these channels during arbitration. Groups with the same priority cannot preempt each other.

T2MC-11173 - [SWS_Adc_00426] Function definitions: `Adc_StartGroupConversion` behavior: Raise the error `ADC_E_BUSY` (priority mechanism disabled, queuing disabled, groups can be implicitly stopped, not in state `ADC_IDLE/ADC_STREAM_COMPLETED`)

Title: [SWS_Adc_00426] Function definitions: `Adc_StartGroupConversion` behavior: Raise the error `ADC_E_BUSY` (priority mechanism disabled, queuing disabled, groups can be implicitly stopped, not in state `ADC_IDLE/ADC_STREAM_COMPLETED`)

Description: [SWS_Adc_00426] [If development error detection for the ADC module is enabled and the priority mechanism is disabled and the queuing is disabled: when called while any of the groups, which can be implicitly stopped, is not in state `ADC_IDLE` and not in state `ADC_STREAM_COMPLETED`, the function `Adc_StartGroupConversion` shall raise development error `ADC_E_BUSY` and return without any action.] ()

Deviations from AUTOSAR

Reason for rejection: The priority mechanism is implemented in hardware and cannot be disabled; therefore, this requirement does not apply.

T2MC-11177 - [SWS_Adc_00428] Function definitions: `Adc_StartGroupConversion` behavior: Raise the error `ADC_E_BUSY` (priority mechanism disabled, queuing enabled, groups can be implicitly stopped, not in state `ADC_IDLE/ADC_STREAM_COMPLETED`)

Title: [SWS_Adc_00428] Function definitions: `Adc_StartGroupConversion` behavior: Raise the error `ADC_E_BUSY` (priority mechanism disabled, queuing enabled, groups can be implicitly stopped, not in state `ADC_IDLE/ADC_STREAM_COMPLETED`)

Description: [SWS_Adc_00428] [If development error detection for the ADC module is enabled and the priority mechanism is disabled and the queuing is enabled: when called while a group, which can be implicitly stopped, is not in state `ADC_IDLE` and not in state `ADC_STREAM_COMPLETED`, the function `Adc_StartGroupConversion` shall raise development error `ADC_E_BUSY` and return without any action.] ()

Reason for rejection: The priority mechanism is implemented in hardware and cannot be disabled; therefore, this requirement does not apply.

T2MC-11409 - [SWS_Adc_00460] Not applicable requirements

Title: [SWS_Adc_00460] Not applicable requirements

Description: [SWS_Adc_00460] [These requirements are not applicable to this specification.] (SRS_BSW_00344, SRS_BSW_00167, SRS_BSW_00170, SRS_BSW_00387, SRS_BSW_00398, SRS_BSW_00375, SRS_BSW_00416, SRS_BSW_00168, SRS_BSW_00423, SRS_BSW_00424, SRS_BSW_00425, SRS_BSW_00426, SRS_BSW_00427, SRS_BSW_00428, SRS_BSW_00429, BSW00431, SRS_BSW_00432, SRS_BSW_00433, BSW00434, SRS_BSW_00417, SRS_BSW_00161, SRS_BSW_00162, SRS_BSW_00005, SRS_BSW_00164, SRS_BSW_00325, SRS_BSW_00326, SRS_BSW_00342, SRS_BSW_00343, SRS_BSW_00160, SRS_BSW_00007, SRS_BSW_00413, SRS_BSW_00347, SRS_BSW_00307, SRS_BSW_00373, SRS_BSW_00301, SRS_BSW_00302, SRS_BSW_00328, SRS_BSW_00312, SRS_BSW_00006, SRS_BSW_00357, SRS_BSW_00355, SRS_BSW_00306, SRS_BSW_00308, SRS_BSW_00371, SRS_BSW_00376, SRS_BSW_00329, SRS_BSW_00330, SRS_BSW_00009, SRS_BSW_00010, SRS_BSW_00341, SRS_BSW_00334, SRS_SPAL_12267, SRS_SPAL_12463, SRS_SPAL_12068, SRS_SPAL_12069, SRS_SPAL_12169, SRS_SPAL_12064, SRS_SPAL_12067, SRS_SPAL_12077, SRS_SPAL_12078, SRS_SPAL_12092, SRS_SPAL_12265)

Reason for rejection: Named RQMs are not applicable.

T2MC-11068 - [SWS_Adc_00473] General behavior: Behavior of preparation process and the setting process when `AdcPowerStateAsynchTransitionMode` is `TRUE`

Title: [SWS_Adc_00473] General behavior: Behavior of preparation process and the setting process when `AdcPowerStateAsynchTransitionMode` is `TRUE`

Description: SWS_Adc_00473 In case the configuration parameter `AdcPowerStateAsynchTransitionMode` is set to `TRUE`, the preparation process shall continue in background after the relative API returns and its completion shall be notified by means of the configured callback.

Reason for rejection: Hardware allows two modes (OFF and ON). No preparation period is required by the hardware. ON and OFF can be manual. Therefore, asynchronous power state transition mode is not supported.

Deviations from AUTOSAR

T2MC-11327 - [SWS_Adc_00480] Configurable interfaces: IoHwAb_Adc_NotifyReadyForPowerState<#Mode> Syntax

Title: [SWS_Adc_00480] Configurable interfaces: IoHwAb_Adc_NotifyReadyForPowerState<#Mode> Syntax

Description: [SWS_Adc_00480] [

Service Name:	IoHwAb_Adc_NotifyReadyForPowerState<#Mode>
Syntax:	void IoHwAb_Adc_NotifyReadyForPowerState<#Mode> (void)
Service ID[hex]:	0x70
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	None
Parameters (inout):	None
Parameters (out):	None
Return Value:	None
Description:	The API shall be invoked by the ADC Driver when the requested power state preparation for mode <#Mode> is completed.

] ()

Reason for rejection: Because there is no preparation period in hardware, asynchronous power state transition mode is not supported. Therefore, notification is not supported.

T2MC-11282 - [SWS_Adc_00483] Function definitions: Adc_SetPowerState behavior: Refer to the necessary parameters in case the normal power state is requested

Title: [SWS_Adc_00483] Function definitions: Adc_SetPowerState behavior: Refer to the necessary parameters in case the normal power state is requested

Description: [SWS_Adc_00483]

[In case the normal Power State is requested, the API shall refer to the necessary parameters contained in the same containers used by Adc_Init.] ()

Reason for rejection: Because there is no information except for setting of OFF power and full power, this function does not refer to the information in the container.

T2MC-11284 - [SWS_Adc_00485] Function definitions: Adc_SetPowerState behavior: Refer to the configuration container

Title: [SWS_Adc_00485] Function definitions: Adc_SetPowerState behavior: Refer to the configuration container

Description: [SWS_Adc_00485]

[The API shall refer to the configuration container related to the required Power State in order to derive some specific features of the state (e.g support of Power States).] ()

Deviations from AUTOSAR

Reason for rejection: There is no information except for setting OFF power and full power; this function does not refer to the information in the container.

T2MC-11288 - [SWS_Adc_00489] Function definitions: `Adc_SetPowerState` behavior: Raise the error `ADC_E_TRANSITION_NOT_POSSIBLE`

Title: [SWS_Adc_00489] Function definitions: `Adc_SetPowerState` behavior: Raise the error `ADC_E_TRANSITION_NOT_POSSIBLE`

Description: [SWS_Adc_00489]

[In case development error reporting is activated:

The API shall report the DET error **ADC_E_TRANSITION_NOT_POSSIBLE** in case the requested power state cannot be directly reached from the current power state.] ()

Reason for rejection: In the hardware feature, the requested power state can be directly reached from the current power state. Therefore, the `ADC_E_TRANSITION_NOT_POSSIBLE` error is not supported.

T2MC-11289 - [SWS_Adc_00490] Function definitions: `Adc_SetPowerState` behavior: Raise the error `ADC_E_PERIPHERAL_NOT_PREPARED`

Title: [SWS_Adc_00490] Function definitions: `Adc_SetPowerState` behavior: Raise the error `ADC_E_PERIPHERAL_NOT_PREPARED`

Description: [SWS_Adc_00490]

[In case development error reporting is activated:

The API shall report the DET error **ADC_E_PERIPHERAL_NOT_PREPARED** in case the HW unit has not been previously prepared for the target power state by use of the API `Adc_PreparePowerState()`.] ()

Reason for rejection: In the hardware feature, there is no preparation period. Therefore, the `ADC_E_PERIPHERAL_NOT_PREPARED` error is not supported.

T2MC-11303 - [SWS_Adc_00498] Function definitions: `Adc_PreparePowerState` behavior: Raise the error `ADC_E_TRANSITION_NOT_POSSIBLE`

Title: [SWS_Adc_00498] Function definitions: `Adc_PreparePowerState` behavior: Raise the error `ADC_E_TRANSITION_NOT_POSSIBLE`

Description: [SWS_Adc_00498]

[In case development error reporting is activated:

The API shall report the DET error **ADC_E_TRANSITION_NOT_POSSIBLE** in case the requested power state cannot be directly reached from the current power state.

All asynchronous operation needed to reach the target power state can be executed in background in the context of `Adc_Main_PowerTransitionManager`.] ()

Reason for rejection: In the hardware feature, the requested power state can be directly reached from the current power state. Therefore, the `ADC_E_TRANSITION_NOT_POSSIBLE` error is not supported.

Deviations from AUTOSAR

T2MC-11308 - [SWS_Adc_00499] Function definitions: `Adc_Main_PowerTransitionManager` behavior: Executes any non-immediate action needed to finalize a power state transition requested by `Adc_PreparePowerState()`

Title: [SWS_Adc_00499] Function definitions: `Adc_Main_PowerTransitionManager` behavior: Executes any non-immediate action needed to finalize a power state transition requested by `Adc_PreparePowerState()`

Description: [SWS_Adc_00499]

[This API executes any non-immediate action needed to finalize a power state transition requested by `Adc_PreparePowerState().()`]

Reason for rejection: Because there is no preparation period in hardware, asynchronous power state transition mode is not supported. Therefore, this main function is not supported.

T2MC-11309 - [SWS_Adc_00500] Function definitions: `Adc_Main_PowerTransitionManager` behavior: The rate of scheduling shall be defined by `Adc MainSchedulePeriod` and shall be variable

Title: [SWS_Adc_00500] Function definitions: `Adc_Main_PowerTransitionManager` behavior: The rate of scheduling shall be defined by `Adc MainSchedulePeriod` and shall be variable

Description: [SWS_Adc_00500]

[The rate of scheduling shall be defined by `Adc MainSchedulePeriod` and shall be variable, as the function only needs to be called if a transition has been requested.]

Reason for rejection: Because there is no preparation period in hardware, asynchronous power state transition mode is not supported. Therefore, this main function is not supported.

T2MC-11310 - [SWS_Adc_00501] Function definitions: `Adc_Main_PowerTransitionManager` behavior: Issue callback notifications

Title: [SWS_Adc_00501] Function definitions: `Adc_Main_PowerTransitionManager` behavior: Issue callback notifications

Description: [SWS_Adc_00501]

[This API shall also issue callback notifications to the eventually registered users (`IoHwAbs`) as configured, only in case the asynch mode is chosen.]

Reason for rejection: Because there is no preparation period in hardware, asynchronous power state transition mode is not supported. Therefore, this main function is not supported.

T2MC-11311 - [SWS_Adc_00502] Function definitions: `Adc_Main_PowerTransitionManager` behavior: Simply return in case the ADC module is not initialized

Title: [SWS_Adc_00502] Function definitions: `Adc_Main_PowerTransitionManager` behavior: Simply return in case the ADC module is not initialized

Description: [SWS_Adc_00502]

[In case the ADC module is not initialized, this function shall simply return without any further elaboration. This is needed to avoid to elaborate uninitialized variables. No DET error shall be entered, because this condition can easily be verified during the startup phase (tasks started before the initialization is complete). Rationale: during the startup phase it can happen that the OS already schedules tasks, which call main functions, while some modules are not initialised yet. This is no real error condition, although need handling, i.e. returning without execution.

Although the transition state monitoring functionality is mandatory, the implementation of this API is

Deviations from AUTOSAR

optional, meaning that if the HW allows for other ways to deliver notification and watch the transition state the implementation of this function can be skipped.] ()

Reason for rejection: Because there is no preparation period in hardware, asynchronous power state transition mode is not supported. Therefore, this main function is not supported.

Limitations

4 Limitations

T2MC-11369 - [ECUC_Adc_00011] Configuration specification: AdcChannel AdcChannelConvTime

Title: [ECUC_Adc_00011] Configuration specification: AdcChannel AdcChannelConvTime

Description:

SWS Item	ECUC_Adc_00011 :		
Name	AdcChannelConvTime		
Description	Configuration of conversion time, i.e. the time during which the analogue value is converted into digital representation, (in clock cycles) for each channel, if supported by hardware. ImplementationType: Adc_ConversionTimeType		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default value	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Limitation: Valid range is only 14.

T2MC-11377 - [ECUC_Adc_00019] Configuration specification: AdcChannel AdcChannelResolution

Title: [ECUC_Adc_00019] Configuration specification: AdcChannel AdcChannelResolution

Description:

SWS Item	ECUC_Adc_00019 :		
Name	AdcChannelResolution		
Description	Channel resolution in bits. ImplementationType: Adc_ResolutionType		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 63		
Default value	--		
Post-Build Variant Multiplicity	true		

Limitations

Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: AdcMaxChannelResolution: The actual resolution has to be less or equal than the maximum resolution.		

Limitation: Valid value is only 12.

T2MC-11376 - [ECUC_Adc_00023] Configuration specification: AdcChannel AdcChannelRefVoltsrcLow

Title: [ECUC_Adc_00023] Configuration specification: AdcChannel AdcChannelRefVoltsrcLow

Description:

SWS Item	ECUC_Adc_00023 :		
Name	AdcChannelRefVoltsrcLow		
Description	Lower reference voltage source for each channel. Enumeration literals are defined vendor specific.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Limitation: This parameter is not used by the ADC driver and therefore not being evaluated.

Limitations

T2MC-11396 - [ECUC_Adc_00087] Configuration specification: AdcHwUnit AdcClockSource

Title: [ECUC_Adc_00087] Configuration specification: AdcHwUnit AdcClockSource

Description:

SWS Item	ECUC_Adc_00087 :		
Name	AdcClockSource		
Description	The ADC module specific clock input for the conversion unit can statically be configured to select different clock sources if provided by hardware. Enumeration literals are defined vendor specific.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Limitation: This parameter is not used by the ADC driver because the clock source input for the ADC is fixed and therefore, not being evaluated.

T2MC-11398 - [ECUC_Adc_00088] Configuration specification: AdcHwUnit AdcPrescale

Title: [ECUC_Adc_00088] Configuration specification: AdcHwUnit AdcPrescale

Description:

SWS Item	ECUC_Adc_00088 :		
Name	AdcPrescale		
Description	Optional ADC module specific clock prescale factor, if supported by hardware. ImplementationType: Adc_PrescaleType		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
	Pre-compile time	X	VARIANT-PRE-COMPILE

Limitations

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

Limitation: This parameter is not used by the ADC driver because hardware controlled by ADC module does not support it and therefore not being evaluated.

T2MC-11375 - [ECUC_Adc_00089] Configuration specification: AdcChannel AdcChannelRefVoltsrcHigh

Title: [ECUC_Adc_00089] Configuration specification: AdcChannel AdcChannelRefVoltsrcHigh

Description:

SWS Item	ECUC_Adc_00089 :		
Name	AdcChannelRefVoltsrcHigh		
Description	Upper reference voltage source for each channel. Enumeration literals are defined vendor specific.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Limitation: This parameter is not used by the ADC driver and therefore not being evaluated.

T2MC-11384 - [ECUC_Adc_00287] Configuration specification: AdcGroup AdcGroupPriority

Title: [ECUC_Adc_00287] Configuration specification: AdcGroup AdcGroupPriority

Description:

SWS Item	ECUC_Adc_00287 :		
Name	AdcGroupPriority		
Description	Priority level of the AdcGroup. ImplementationType: Adc_GroupPriorityType		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		

Limitations

Default value	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: ADC_PRIORITY_IMPLEMENTATION		

Limitation: Valid range is 0...7.

T2MC-11378 - [ECUC_Adc_00290] Configuration specification: AdcChannel AdcChannelSampTime

Title: [ECUC_Adc_00290] Configuration specification: AdcChannel AdcChannelSampTime

Description:

SWS Item	ECUC_Adc_00290 :		
Name	AdcChannelSampTime		
Description	Configuration of sampling time, i.e. the time during which the value is sampled, (in clock cycles) for each channel, if supported by hardware. ImplementationType: Adc_SamplingTimeType		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default value	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Limitation: Valid range is 1...4095.

Limitations

T2MC-11351 - [ECUC_Adc_00391] Configuration specification: AdcGeneral AdcEnableQueuing

Title: [ECUC_Adc_00391] Configuration specification: AdcGeneral AdcEnableQueuing

Description:

SWS Item	ECUC_Adc_00391 :		
Name	AdcEnableQueuing		
Description	<p>Determines, if the queuing mechanism is active in case of priority mechanism disabled.</p> <p>Note: If priority mechanism is enabled, queuing mechanism is always active and the parameter ADC_ENABLE_QUEUING is not evaluated. true: Enabled. false: Disabled.</p>		
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	<p>scope: local</p> <p>dependency: AdcPriorityImplementation: parameter is only evaluated for priority implementation ADC_PRIORITY_NONE.</p>		

Limitation: This parameter is not evaluated by the ADC driver because the prioritization mechanism ADC_PRIORITY_NONE is not supported.

T2MC-11357 - [ECUC_Adc_00393] Configuration specification: AdcGeneral AdcPriorityImplementation

Title: [ECUC_Adc_00393] Configuration specification: AdcGeneral AdcPriorityImplementation

Description:

SWS Item	ECUC_Adc_00393 :		
Name	AdcPriorityImplementation		
Description	<p>Determines whether a priority mechanism is available for prioritization of the conversion requests and if available, the type of prioritization mechanism. The selection applies for groups with trigger source software and trigger source hardware. Two types of prioritization mechanism can be selected. The hardware prioritization mechanism (AdcPriorityHw) uses the ADC hardware features for prioritization of the software conversion requests and hardware trigger signals for groups with trigger source hardware. The mixed hardware and software prioritization mechanism (AdcPriorityHwSw) uses the ADC hardware features for prioritization of ADC hardware trigger for groups with trigger source hardware and a software implemented prioritization mechanism for groups with trigger source software. The group priorities for software triggered groups are typically configured with lower priority levels than the group priorities for hardware triggered groups.</p> <p>ImplementationType: Adc_PriorityImplementationType</p>		

Limitations

Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	ADC_PRIORITY_HW	Hardware priority mechanism is available only	
	ADC_PRIORITY_HW_SW	Hardware and software priority mechanism is available	
	ADC_PRIORITY_NONE	priority mechanism is not available	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
		-	
	Post-build time	-	
		-	
		-	
Scope/ Dependency	scope: local		

Limitation: Because HW priority mechanism has better performance than SW implementation, only ADC_PRIORITY_HW is supported and HW priority mechanism is used for all groups (including SW triggered).

T2MC-11387 - [ECUC_Adc_00400] Configuration specification: AdcGroup AdcHwTrigSignal

Title: [ECUC_Adc_00400] Configuration specification: AdcGroup AdcHwTrigSignal

Description:

SWS Item	ECUC_Adc_00400 :		
Name	AdcHwTrigSignal		
Description	Configures on which edge of the hardware trigger signal the driver should react, i.e. start the conversion (only if supported by the ADC hardware). ImplementationType: Adc_HwTriggerSignalType		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	ADC_HW_TRIG_BOTH_EDGES	React on both edges of the hardware trigger signal (only if supported by the ADC hardware).	
	ADC_HW_TRIG_FALLING_EDGE	React on the falling edge of the hardware trigger signal (only if supported by the ADC hardware).	
	ADC_HW_TRIG_RISING_EDGE	React on the rising edge of the hardware trigger signal (only if supported by the ADC hardware).	
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
	Pre-compile time	X	VARIANT-PRE-COMPILE

Limitations

Multiplicity Configuration Class	Link time	-	
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	-	
	Post-build time	X	VARIANT-POST-BUILD
Scope/ Dependency	scope: local dependency: AdcTriggSrcHw: Valid only if the group is configured to be triggered by a hardware event.		

Limitation: This parameter is not used by the ADC driver because the hardware controlled by the ADC module does not have this feature and therefore not being evaluated. A similar feature is supported by other modules (e.g., PORT driver).

T2MC-11388 - [ECUC_Adc_00401] Configuration specification: AdcGroup AdcHwTrigTimer

Title: [ECUC_Adc_00401] Configuration specification: AdcGroup AdcHwTrigTimer

Description:

SWS Item	ECUC_Adc_00401 :		
Name	AdcHwTrigTimer		
Description	Reload value of the ADC module embedded timer (only if supported by ADC hardware). ImplementationType: Adc_HwTriggerTimerType		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default value	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: AdcTriggSrcHw: Valid only if the group is configured to be triggered by a hardware event.		

Limitation: This parameter is not used by the ADC driver because the hardware controlled by the ADC module does not have this feature and therefore not being evaluated. A similar feature is supported by other modules (e.g., PWM driver).

Limitations

T2MC-11349 - [ECUC_Adc_00405] Configuration specification: AdcGeneral AdcDevErrorDetect

Title: [ECUC_Adc_00405] Configuration specification: AdcGeneral AdcDevErrorDetect

Description:

SWS Item	ECUC_Adc_00405 :		
Name	AdcDevErrorDetect		
Description	Switches the Default Error Tracer (Det) detection and notification ON or OFF. - true: enabled (ON). - false: disabled (OFF).		
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	scope: local		

Limitation: Setting this parameter to false will disable the notification of development errors via DET. However, in contrast to AUTOSAR specification, detection of development errors is still enabled and errors will be reported via AdcErrorCalloutFunction.

T2MC-11403 - [ECUC_Adc_00410] Configuration specification: AdcPublishedInformation
AdcChannelValueSigned

Title: [ECUC_Adc_00410] Configuration specification: AdcPublishedInformation AdcChannelValueSigned

Description:

SWS Item	ECUC_Adc_00410 :		
Name	AdcChannelValueSigned		
Description	Information whether the result value of the ADC driver has sign information (true) or not (false). If the result shall be interpreted as signed value it shall apply to C-language rules.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Published Information	X	All Variants
Scope / Dependency	scope: local		

Limitation: This parameter is fixed to true.

Limitations

T2MC-11404 - [ECUC_Adc_00411] Configuration specification: AdcPublishedInformation
AdcGroupFirstChannelFixed

Title: [ECUC_Adc_00411] Configuration specification: AdcPublishedInformation AdcGroupFirstChannelFixed

Description:

SWS Item	ECUC_Adc_00411 :		
Name	AdcGroupFirstChannelFixed		
Description	Information whether the first channel of an ADC Channel group can be configured (false) or is fixed (true) to a value determined by the ADC HW Unit.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Published Information	X	All Variants
Scope / Dependency	scope: local		

Limitation: This parameter is fixed to 'false'.

T2MC-11405 - [ECUC_Adc_00412] Configuration specification: AdcPublishedInformation
AdcMaxChannelResolution

Title: [ECUC_Adc_00412] Configuration specification: AdcPublishedInformation AdcMaxChannelResolution

Description:

SWS Item	ECUC_Adc_00412 :		
Name	AdcMaxChannelResolution		
Description	Maximum Channel resolution in bits (does not specify accuracy).		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 63		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Published Information	X	All Variants
Scope / Dependency	scope: local		

Limitation: This parameter is fixed to 12.

T2MC-11385 - [ECUC_Adc_00435] Configuration specification: AdcGroup AdcGroupReplacement

Title: [ECUC_Adc_00435] Configuration specification: AdcGroup AdcGroupReplacement

Description:

SWS Item	ECUC_Adc_00435 :		
Name	AdcGroupReplacement		

Limitations

Description	Replacement mechanism, which is used on ADC group level, if a group conversion is interrupted by a group which has a higher priority. ImplementationType: Adc_GroupReplacementType		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	ADC_GROUP_REPL_ABORT_RESTART	Abort/Restart mechanism is used on group level, if a group is interrupted by a higher priority group. The complete conversion round of the interrupted group (all group channels) is restarted after the higher priority group conversion is finished. If the group is configured in streaming access mode, only the results of the interrupted conversion round are discarded. Results of previous conversion rounds which are already written to the result buffer are not affected.	
	ADC_GROUP_REPL_SUSPEND_RESUME	Suspend/Resume mechanism is used on group level, if a group is interrupted by a higher priority group. The conversions round (conversion of all group channels) of the interrupted group is completed after the higher priority group conversion is finished. If the group is configured in streaming access mode, only the results of the interrupted conversion round are discarded. Results of previous conversion rounds which are already written to the result buffer are not affected.	
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	- -	
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	- -	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Limitation: Valid range is ADC_GROUP_REPL_ABORT_RESTART, ADC_GROUP_REPL_SUSPEND_RESUME, and ADC_GROUP_REPL_ABORT_RESUME.

Limitations

T2MC-11373 - [ECUC_Adc_00454] Configuration specification: AdcChannel AdcChannelLowLimit

Title: [ECUC_Adc_00454] Configuration specification: AdcChannel AdcChannelLowLimit

Description:

SWS Item	ECUC_Adc_00454 :		
Name	AdcChannelLowLimit		
Description	Low limit - used for limit checking.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default value	--		
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: local dependency: AdcEnableLimitCheck: not available if limit checking is not globally enabled. AdcChannelLimitCheck: not available if channel specific limit check is not enabled. AdcChannelHighLimit: has to be less or equal than AdcChannelHighLimit.		

Limitation: Valid range is 0...65535.

T2MC-11370 - [ECUC_Adc_00455] Configuration specification: AdcChannel AdcChannelHighLimit

Title: [ECUC_Adc_00455] Configuration specification: AdcChannel AdcChannelHighLimit

Description:

SWS Item	ECUC_Adc_00455 :	
Name	AdcChannelHighLimit	
Description	High limit - used for limit checking.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	0 .. 18446744073709551615	
Default value	--	

Limitations

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: local dependency: AdcEnableLimitCheck: not available if limit checking is not globally enabled. AdcChannelLimitCheck: not available if channel specific limit check is not enabled. AdcChannelLowLimit: has to be greater or equal than AdcChannelLowLimit.		

Limitation: Valid range is 0...65535.

T2MC-11356 - [ECUC_Adc_00458] Configuration specification: AdcGeneral

AdcPowerStateAsynchTransitionMode

Title: [ECUC_Adc_00458] Configuration specification: AdcGeneral AdcPowerStateAsynchTransitionMode

Description:

SWS Item	ECUC_Adc_00458 :		
Name	AdcPowerStateAsynchTransitionMode		
Description	Enables / disables support of the ADC Driver to the asynchronous power state transition.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
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: local dependency: This parameter shall only be configured if the parameter AdcLowPowerStatesSupport is set to true.		

Limitations

Limitation: Because there is no preparation period in hardware, only synchronous power state transition mode is supported. Therefore, this parameter is not used by the ADC driver and not being evaluated.

T2MC-11364 - [ECUC_Adc_00460] Configuration specification: AdcPowerStateConfig
AdcPowerStateReadyCbkJRef

Title: [ECUC_Adc_00460] Configuration specification: AdcPowerStateConfig AdcPowerStateReadyCbkJRef

Description:

SWS Item	ECUC_Adc_00460 :		
Name	AdcPowerStateReadyCbkJRef		
Description	Each instance of this parameter contains a reference to a power mode callback defined in a CDD or IoHwAbs component.		
Multiplicity	1		
Type	EcucFunctionNameDef		
Default value	--		
maxLength	--		
minLength	--		
regularExpression	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local dependency: This parameter shall only be configured if the parameter AdcLowPowerStatesSupport is set to true.		

Limitation: Because there is no preparation period in hardware, only synchronous power state transition mode is supported. Therefore, this parameter is not used by the ADC driver and not being evaluated.

T2MC-11363 - [ECUC_Adc_00461] Configuration specification: AdcPowerStateConfig AdcPowerState

Title: [ECUC_Adc_00461] Configuration specification: AdcPowerStateConfig AdcPowerState

Description:

SWS Item	ECUC_Adc_00461 :
Name	AdcPowerState

Limitations

Description	Each instance of this parameter describes a different power state supported by the ADC HW. It should be defined by the HW supplier and used by the ADC Driver to reference specific HW configurations which set the ADC HW module in the referenced power state. At least the power mode corresponding to full power state shall be always configured.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 18446744073709551615		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local dependency: This parameter shall only be configured if the parameter AdcLowPowerStatesSupport is set to true.		

Limitation: Valid range is 0 (ADC_FULL_POWER), 1 (ADC_OFF_POWER).

T2MC-11407 - [SWS_Adc_00099] Configuration specification: Symbolic names

Title: [SWS_Adc_00099] Configuration specification: Symbolic names

Description: [SWS_Adc_00099] [The symbolic names of ADC channels and ADC channel groups for use by the upper layer shall be defined by the configurator. They are to be Defined in the modules configuration header file.] (SRS_Adc_12307, SRS_Adc_12447)

Limitation: Because they are used for additional API functions, symbolic names of the ADC hardware unit also are generated.

T2MC-11344 - [SWS_Adc_00363] Configuration specification variants

Title: [SWS_Adc_00363] Configuration specification variants

Description: [SWS_Adc_00363]

[**VARIANT-POST-BUILD:** Parameters with "Pre-compile time", "Link time" and "Post-build time" are allowed in this variant.] ()

Limitation: Only post-build time variant is supported.

T2MC-10996 - [SWS_Adc_00451] General behavior: Limit checking enabled channel

Title: [SWS_Adc_00451] General behavior: Limit checking enabled channel

Description: [SWS_Adc_00451] [The ADC module's configuration shall be such that an ADC Channel group contains exactly one ADC Channel if the global limit checking feature is enabled and the channel specific limit checking is enabled for the ADC Channel.] ()

Limitation: The restriction to one channel for limit-checking-enabled groups does not apply, but only the last channel of a group may have limit checking enabled.

Limitations

T2MC-11066 - [SWS_Adc_00471] General behavior: Synchronous and asynchronous power state transitions

Title: [SWS_Adc_00471] General behavior: Synchronous and asynchronous power state transitions

Description: **SWS_Adc_00471** The ADC Driver shall support synchronous and asynchronous power state transitions, depending on the value of the configuration parameter `AdcPowerStateAsynchTransitionMode`.

Limitation: Hardware allows two modes (OFF and ON). No preparation period is required by the hardware. ON and OFF can be manual. Therefore, only synchronous power state transition mode is supported.

T2MC-11279 - [SWS_Adc_00475] Function definitions: `Adc_SetPowerState` Syntax

Title: [SWS_Adc_00475] Function definitions: `Adc_SetPowerState` Syntax

Description: [SWS_Adc_00475] [

Service name:	<code>Adc_SetPowerState</code>	
Syntax:	<pre>Std_ReturnType Adc_SetPowerState(Adc_PowerStateRequestResultType* Result)</pre>	
Service ID[hex]:	0x10	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	None	
Parameters (inout):	None	
Parameters (out):	Result	If the API returns <code>E_OK</code> : <code>ADC_SERVICE_ACCEPTED</code> : Power state change executed. If the API returns <code>E_NOT_OK</code> : <code>ADC_NOT_INIT</code> : ADC Module not initialized. <code>ADC_SEQUENCE_ERROR</code> : wrong API call sequence. <code>ADC_HW_FAILURE</code> : the HW module has a failure which prevents it to enter the required power state.
Return value:	<code>Std_ReturnType</code>	<code>E_OK</code> : Power Mode changed <code>E_NOT_OK</code> : request rejected
Description:	This API configures the Adc module so that it enters the already prepared power state, chosen between a predefined set of configured ones.	

]()

Limitation: In order to clarify the error information, `ADC_POWER_STATE_NOT_SUPP` and `ADC_TRANS_NOT_POSSIBLE` are added to result value. In addition, `ADC_SEQUENCE_ERROR` and `ADC_HW_FAILURE` are deleted from the result value according to the hardware feature.

Limitations

T2MC-11298 - [SWS_Adc_00478] Function definitions: Adc_PreparePowerState Syntax

Title: [SWS_Adc_00478] Function definitions: Adc_PreparePowerState Syntax

Description: [SWS_Adc_00478] [

Service name:	Adc_PreparePowerState	
Syntax:	Std_ReturnType Adc_PreparePowerState(Adc_PowerStateType PowerState, Adc_PowerStateRequestResultType* Result)	
Service ID[hex]:	0x13	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	PowerState	The target power state intended to be attained
Parameters (inout):	None	
Parameters (out):	Result	<p>If the API returns E_OK: ADC_SERVICE_ACCEPTED: ADC Module power state preparation was started.</p> <p>If the API returns E_NOT_OK: ADC_NOT_INIT: ADC Module not initialized. ADC_SEQUENCE_ERROR: wrong API call sequence (Current Power State = Target Power State). ADC_POWER_STATE_NOT_SUPP: ADC Module does not support the requested power state. ADC_TRANS_NOT_POSSIBLE: ADC Module cannot transition directly from the current power state to the requested power state or the HW peripheral is still busy.</p>
Return value:	Std_ReturnType	<p>E_OK: Preparation process started</p> <p>E_NOT_OK: Service is rejected</p>
Description:	This API starts the needed process to allow the ADC HW module to enter the requested power state.	

] ()

Limitation: ADC_SEQUENCE_ERROR and ADC_TRANS_NOT_POSSIBLE are deleted from result value according to the hardware feature.

T2MC-11307 - [SWS_Adc_00479] Function definitions: Adc_Main_PowerTransitionManager Syntax

Title: [SWS_Adc_00479] Function definitions: Adc_Main_PowerTransitionManager Syntax

Description: [SWS_Adc_00479] [

Service name:	Adc_Main_PowerTransitionManager	
Syntax:	void Adc_Main_PowerTransitionManager(void)	
Service ID[hex]:	0x14	

Limitations

Description:	This API is cyclically called and supervises the power state transitions, checking for the readiness of the module and issuing the callbacks IoHwAb_Adc_NotifyReadyForPowerState<Mode> (see AdcPowerStateReadyCbRef configuration parameter).
---------------------	---

] ()

Limitation: Because there is no preparation period in hardware, asynchronous power state transition mode is not supported. Therefore, this main function is implemented as a null function.

T2MC-11299 - [SWS_Adc_00494] Function definitions: Adc_PrepowerState behavior: Initiates all actions needed to enable a HW module to enter the target power state

Title: [SWS_Adc_00494] Function definitions: Adc_PrepowerState behavior: Initiates all actions needed to enable a HW module to enter the target power state

Description: [SWS_Adc_00494]

[This API initiates all actions needed to enable a HW module to enter the target power state. The possibility to operate the periphery depends on the power state and the HW features. These properties should be known to the integrator and the decision whether to use the periphery or not is in his responsibility.] ()

Limitation: No preparation for hardware to enter the target power state is necessary.

T2MC-11302 - [SWS_Adc_00497] Function definitions: Adc_PrepowerState behavior: Raise the error ADC_E_POWER_MODE_NOT_SUPPORTED

Title: [SWS_Adc_00497] Function definitions: Adc_PrepowerState behavior: Raise the error ADC_E_POWER_MODE_NOT_SUPPORTED

Description: [SWS_Adc_00497]

[In case development error reporting is activated:

The API shall report the DET error ADC_E_POWER_MODE_NOT_SUPPORTED in case this API is called with an unsupported power state is requested or the peripheral does not support low power states at all.] ()

Limitation: The same DET error has different names within the AUTOSAR specification document, so it needs to be changed from ADC_E_POWER_MODE_NOT_SUPPORTED to ADC_E_POWER_STATE_NOT_SUPPORTED in this requirement.

T2MC-11091 - [SWS_Adc_00509] Type definitions: Adc_PrescaleType

Title: [SWS_Adc_00509] Type definitions: Adc_PrescaleType

Description: [SWS_Adc_00509]

Name:	Adc_PrescaleType		
Type:	uint		
Range:	--	-	The range of this type is μ C specific and has to be described by the supplier.
Description:	Type of clock prescaler factor. (This is not an API type).		

] ()

Limitations

Limitation: Hardware controlled by the ADC module does not need the prescaling of clock; therefore, prescaler is not available. Subsequently this type is not used by the ADC driver. However, this type is implemented as uint32.

T2MC-11093 - [SWS_Adc_00510] Type definitions: Adc_ConversionTimeType

Title: [SWS_Adc_00510] Type definitions: Adc_ConversionTimeType

Description: [SWS_Adc_00510][

Name:	Adc_ConversionTimeType		
Type:	uint		
Range:	--	- -	The range of this type is μ C specific and has to be described by the supplier.
Description:	Type of conversion time, i.e. the time during which the sampled analogue value is converted into digital representation. (This is not an API type).		

] ()

Limitation: In the hardware feature, the conversion cycle is fixed. Therefore, this type is not used by the ADC driver. However, this type is implemented as uint8.

T2MC-11097 - [SWS_Adc_00512] Type definitions: Adc_ResolutionType

Title: [SWS_Adc_00512] Type definitions: Adc_ResolutionType

Description: [SWS_Adc_00512][

Name:	Adc_ResolutionType		
Type:	uint8		
Range:	--	- -	The range of this type is μ C specific and has to be described by the supplier.
Description:	Type of channel resolution in number of bits. (This is not an API type).		

] ()

Limitation: In the hardware feature, resolution bit size is fixed to 12 bits. Therefore, this type is not used by the ADC driver. However, this type is implemented as uint8.

T2MC-11099 - [SWS_Adc_00513] Type definitions: Adc_StatusType

Title: [SWS_Adc_00513] Type definitions: Adc_StatusType

Description: [SWS_Adc_00513][

Name:	Adc_StatusType		
Type:	Enumeration		
Range:	ADC_IDLE	- -	The conversion of the specified group has not been started. No result is available.

Limitations

	ADC_BUSY	<ul style="list-style-type: none"> - The conversion of the specified group has been started and is still going on. - So far no result is available.
	ADC_COMPLETED	<ul style="list-style-type: none"> - A conversion round (which is not the final one) of the specified group has been finished. - A result is available for all channels of the group.
	ADC_STREAM_COMPLETED	<ul style="list-style-type: none"> - The result buffer is completely filled - For each channel of the selected group the number of samples to be acquired is available
Description:	Current status of the conversion of the requested ADC Channel group.	

] ()

Limitation: Because the ADC module manages the status of hardware errors, ADC_ERROR is added to the range field of this type.

T2MC-11107 - [SWS_Adc_00517] Type definitions: Adc_GroupDefType

Title: [SWS_Adc_00517] Type definitions: Adc_GroupDefType**Description:** [SWS_Adc_00517][

Name:	Adc_GroupDefType
Type:	--
Description:	Type for assignment of channels to a channel group (this is not an API type).

] ()

Limitation: The ADC driver does not use this type because the ADC module can control the assignment of channels to a channel group without using this type. However, this type is implemented as uint32.

T2MC-11115 - [SWS_Adc_00520] Type definitions: Adc_HwTriggerSignalType

Title: [SWS_Adc_00520] Type definitions: Adc_HwTriggerSignalType**Description:** [SWS_Adc_00520][

Name:	Adc_HwTriggerSignalType	
Type:	Enumeration	
Range:	ADC_HW_TRIG_RISING_EDGE	React on the rising edge of the hardware trigger signal (only if supported by the ADC hardware).
	ADC_HW_TRIG_FALLING_EDGE	React on the falling edge of the hardware trigger signal (only if supported by the ADC hardware).
	ADC_HW_TRIG_BOTH_EDGES	React on both edges of the hardware trigger signal (only if supported by the ADC hardware).
Description:	Type for configuring on which edge of the hardware trigger signal the driver should react, i.e. start the conversion (only if supported by the ADC hardware).	

] ()

Limitation: The hardware controlled by the ADC module does not support the selection of trigger type. Therefore, this type is not used by the ADC driver.

Limitations

T2MC-11117 - [SWS_Adc_00521] Type definitions: Adc_HwTriggerTimerType

Title: [SWS_Adc_00521] Type definitions: Adc_HwTriggerTimerType

Description: [SWS_Adc_00521][

Name:	Adc_HwTriggerTimerType		
Type:	uint		
Range:	--	-	The range of this type is μ C specific and has to be described by the supplier.
Description:	Type for the reload value of the ADC module embedded timer (only if supported by the ADC hardware).		

] ()

Limitation: Because the hardware controlled by the ADC module does not have this feature, this type is not used by the ADC driver. However, this type is implemented as uint32.

T2MC-11119 - [SWS_Adc_00522] Type definitions: Adc_PriorityImplementationType

Title: [SWS_Adc_00522] Type definitions: Adc_PriorityImplementationType

Description:

[SWS_Adc_00522][

Name:	Adc_PriorityImplementationType		
Type:	Enumeration		
Range:	ADC_PRIORITY_NONE	priority mechanism is not available	
	ADC_PRIORITY_HW	Hardware priority mechanism is available only	
	ADC_PRIORITY_HW_SW	Hardware and software priority mechanism is available	
Description:	Type for configuring the prioritization mechanism.		

] ()

Limitation: This type is not used by the ADC module because only ADC_PRIORITY_HW is supported. However, this type is implemented.

T2MC-11121 - [SWS_Adc_00523] Type definitions: Adc_GroupReplacementType

Title: [SWS_Adc_00523] Type definitions: Adc_GroupReplacementType

Description: [SWS_Adc_00523][

Name:	Adc_GroupReplacementType		
Type:	Enumeration		

Limitations

Range:	ADC_GROUP_REPL_ABORT_RESTART	Abort/Restart mechanism is used on group level, if a group is interrupted by a higher priority group. The complete conversion round of the interrupted group (all group channels) is restarted after the higher priority group conversion is finished. If the group is configured in streaming access mode, only the results of the interrupted conversion round are discarded. Results of previous conversion rounds which are already written to the result buffer are not affected.
	ADC_GROUP_REPL_SUSPEND_RESUME	Suspend/Resume mechanism is used on group level, if a group is interrupted by a higher priority group. The conversion round of the interrupted group is completed after the higher priority group conversion is finished. Results of previous conversion rounds which are already written to the result buffer are not affected.
Description:	Replacement mechanism, which is used on ADC group level, if a group conversion is interrupted by a group which has a higher priority.	

] ()

Limitation: Valid range of this type: ADC_GROUP_REPL_ABORT_RESTART, ADC_GROUP_REPL_SUSPEND_RESUME, and ADC_GROUP_REPL_ABORT_RESUME.

T2MC-11123 - [SWS_Adc_00524] Type definitions: Adc_ChannelRangeSelectType

Title: [SWS_Adc_00524] Type definitions: Adc_ChannelRangeSelectType

Description: [SWS_Adc_00524][

Name:	Adc_ChannelRangeSelectType	
Type:	Enumeration	
Range:	ADC_RANGE_UNDER_LOW	Range below low limit - low limit value included
	ADC_RANGE_BETWEEN	Range between low limit and high limit - high limit value included
	ADC_RANGE_OVER_HIGH	Range above high limit
	ADC_RANGE_ALWAYS	Complete range - independent from channel limit settings
	ADC_RANGE_NOT_UNDER_LOW	Range above low limit
	ADC_RANGE_NOT_BETWEEN	Range above high limit or below low limit - low limit value included
	ADC_RANGE_NOT_OVER_HIGH	Range below high limit - high limit value included

Limitations

Description:	In case of active limit checking: defines which conversion values are taken into account related to the boards defined with AdcChannelLowLimit and AdcChannelHighLimit.
---------------------	---

] ()

Limitation: The ADC driver does not use this type definition in order to generate a value to set directly to the hardware register. However, this type is implemented as an AUTOSAR specification.

T2MC-11125 - [SWS_Adc_00525] Type definitions: Adc_ResultAlignmentType

Title: [SWS_Adc_00525] Type definitions: Adc_ResultAlignmentType

Description: [SWS_Adc_00525][

Name:	Adc_ResultAlignmentType	
Type:	Enumeration	
Range:	ADC_ALIGN_LEFT	left alignment
	ADC_ALIGN_RIGHT	right alignment
Description:	Type for alignment of ADC raw results in ADC result buffer (left/right alignment).	

] ()

Limitation: The ADC driver does not use this type definition in order to generate a value to set directly to the hardware register. However, this type is implemented as an AUTOSAR specification.

T2MC-14834 - ADC: AUTOSAR C implementation rules

Title: ADC: AUTOSAR C implementation rules

Description: The MCAL modules shall fulfill all design and implementation guidelines as described in Specification of C Implementation Rules AUTOSAR_TR_CImplementationRules.pdf.

Limitation: Out of scope: keyword macros 'CONST' and 'VAR' are not required for declaration/definition of the local variable, function parameter, and structure/union fields.

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-38119 - [ADC, SPI] Problems in some *arxml* files

Title: [ADC, SPI] Problems in some *arxml* files

Description: The following code is implemented in the *arxml* file (i.e. *Adc_Bswmd.arxml* and *Spi_Bswmd.arxml*).

However, the code does not work correctly because the relative path is not available in tresos Studio. The absolute path should be used instead.

Adc_Bswmd.arxml:

```
[!INCLUDE "..../Adc_TS_T40D13M1I0R0/generate/include/Adc_Macros.m"!][!//
```

Spi_Bswmd.arxml:

```
[!INCLUDE "..../Spi_TS_T40D13M1I0R0/generate/include/Spi_Macros.m"!][!//
```

```
[!INCLUDE "..../Spi_TS_T40D13M1I0R0/generate/include/Spi_Macros_Der.m"!][!//
```

Workaround:

Replace the relative path with the absolute path below:

Adc_Bswmd.arxml:

```
[!INCLUDE "?concat($pluginPath,\"generate\\include\\Adc_Macros.m\")"!][!//
```

Spi_Bswmd.arxml:

```
[!INCLUDE "?concat($pluginPath,\"generate\\include\\Spi_Macros.m\")"!][!//
```

```
[!INCLUDE "?concat($pluginPath,\"generate\\include\\Spi_Macros_Der.m\")"!][!//
```

T2MC-39515 - [ADC] DET error process of power state control APIs are wrong

Title: [ADC] DET error process of power state control APIs are wrong

Description: The error process of *Adc_SetPowerState* and *Adc_PreparePowerState* is wrong.

In case the target power state is the same as the current one, these APIs report DET error and return with an *E_NOT_OK* result. However, the right operation is "No action is executed and the API returns immediately with an *E_OK* result".

The value stored in the Result of the output parameter of *Adc_SetPowerState* is not clear.

If the target power state is wrong, the value stored in Result is changed from *ADC_SEQUENCE_ERROR* to *ADC_POWER_STATE_NOT_SUPP*.

In case one or more HW channels are in a state different than IDLE, the value stored to Result is changed from *ADC_SEQUENCE_ERROR* to *ADC_TRANS_NOT_POSSIBLE*.

T2MC-38071 - [ADC] Diagnostic enable bit is not set by power management APIs

Title: [ADC] Diagnostic enable bit is not set by power management APIs

Description: Diagnostic enable bit would not be set after *Adc_SetPowerState* is called with *ADC_FULL_POWER* in certain condition.

Diagnostic enable bit needs to be set in each HW Unit by the ADC driver according to the configuration setting.

However, the ADC driver always sets its referencing diagnostic enable setting of the first HW Unit even

Version history

though it is not the first HW Unit.

It means that the diagnostic enable bit in all HW Units is disabled, if it is disabled in the first HW Unit.

Error conditions:

- More than two HW Units are configured.
- Diagnostic reference is not used in the first HW Unit, but in other HW Units.

Workaround:

Diagnostic reference needs to be enabled in the first HW Unit.

It makes other HW Units available to use diagnostic reference.

T2MC-38118 - [ADC] SW priority mechanism missing

Title: [ADC] SW priority mechanism missing

Description: A configuration parameter AdcPriorityImplementation has the following values according to the AUTOSAR specification.

- ADC_PRIORITY_HW
- ADC_PRIORITY_HW_SW
- ADC_PRIORITY_NONE

Only ADC_PRIORITY_HW appears in the configuration parameter because it is only supported by the ADC driver.

All values should be shown irrespective of whether they are supported.

T2MC-39562 - [ADC] Unnecessary ISR is generated even though it is not used.

Title: [ADC] Unnecessary ISR is generated even though it is not used.

Description: If AdcInterruptMode is enabled, the ISR handler is generated. However, the ISR handler is also generated even if AdcInterruptMode is disabled.

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-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-39536 - [ADC] Add the limitation of unused type definition

Title: [ADC] Add the limitation of unused type definition

Description: The following type definition requirements specified by AUTOSAR must add the limitation for user guide and release note because they are not used by the ADC driver.

[SWS_Adc_00525] Adc_ResultAlignmentType

[SWS_Adc_00524] Adc_ChannelRangeSelectType

[SWS_Adc_00522] Adc_PriorityImplementationType

T2MC-45751 - [ADC] Data generated from the configuration is wrong.

Title: [ADC] Data generated from the configuration is wrong.

Description: Following are the problems in the data generated from configuration.

- If AdcInterruptMode is disabled, the ISR handler is not generated. However, management data of interrupt is generated. This data is unnecessary.
 - If two DMA units (not DMA channels) are used, the syntax of the generated data is incorrect. Therefore, a compile error occurs.
-

T2MC-39160 - [ADC] Improvement of limitation on P-DMA usage when limit check

Title: [ADC] Improvement of limitation on P-DMA usage when limit check

Description: For the ADC groups in which limit-checking is enabled, P-DMA can be used only in two groups per hardware unit.

This is due to the hardware restriction.

The trigger to P-DMA corresponding to the range detect event must be connected by two generic triggers per hardware unit.

However, hardware adds the range detect event information for a 1-to-1 trigger connection from each ADC channel to a corresponding P-DMA channel. Therefore, the ADC driver can remove this restriction.

PASS IP of ADC hardware adds TR_DONE_GRP_VIO bit field in POST_CTL register.

ADC driver adds configuration parameter AdcDmaOutputTriggerSelect to choose whether use 1-to-1 trigger or generic trigger.

T2MC-48302 - [ADC] Improvement of unused AdcChannel

Title: [ADC] Improvement of unused AdcChannel

Description: The ADC driver allows the configuration of ADC channels without using them in an ADC group. A warning should be provided to inform the user that there are unassigned channels.

T2MC-43030 - [ADC] Instance number addition of PASS hardware

Title: [ADC] Instance number addition of PASS hardware

Description: PASS information format of the resource properties file is changed by an instance number addition of PASS hardware.

Therefore, the ADC driver must also be changed to support this change.

Version history

Changed enumeration name selected by AdcHwUnitId configuration parameter.
For example, from ADC_SAR_0 to ADC_SAR_0_0 (ADC_SAR_[PASS number]_[SAR number]
PASS number is fixed 0.

T2MC-39877 - [ADC] No implementation of dependencies on some configuration parameters

Title: [ADC] No implementation of dependencies on some configuration parameters

Description: The following configuration parameters do not implement dependencies described in AUTOSAR.

The dependencies need to be implemented to avoid deviation from the AUTOSAR specification.

AdcGroupPriority: This parameter depends on AdcPriorityImplementation (AdcGroupPriority should be set if AdcPriorityImplementation is set to 'ADC_PRIORITY_HW').

AdcNotification: This parameter is only available, if the notification capability is configured as available by AdcGrpNotifCapability.

AdcPowerState: This parameter shall only be configured if the parameter AdcLowPowerStatesSupport is set to true.

T2MC-39161 - [ADC] Support of analog calibration

Title: [ADC] Support of analog calibration

Description: The ADC driver supports the analog calibration feature.

The ADC driver adds the following APIs and configuration parameter.

- Adc_ChangeCalibrationChannel API
 - Adc_SetCalibrationValue API
 - Adc_GetCalibrationAlternateValue API
 - Adc_GetCalibrationValue API
 - AdcUseAlternateCalibration configuration parameter
-

T2MC-39516 - [ADC] Support TRAVEO™ T2G-B-H-8M

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

Description: AUTOSAR MCAL supports the TRAVEO™ T2G-B-H-8M.

The Arm® core of TRAVEO™ T2G-B-H-8M is Cortex®-M7, and this core supports the cache feature and CPUs tightly coupled memories (TCMs).

Therefore, the ADC user guide adds a caution about the cache feature and TCMs of the buffer used by the DMA transfer.

T2MC-43547 - [ADC] There is no critical section in the interrupt handler.

Title: [ADC] There is no critical section in the interrupt handler.

Description: If the APIs are interrupted during the execution of the ADC interrupt handler (for example, when the APIs are called by a high priority interrupt handler), the ADC driver does not work correctly.

Therefore, the ADC driver needs the critical section in the ADC interrupt handler.

Version history

T2MC-41024 - [ADC] There is not enough default setting of hardware registers.

Title: [ADC] There is not enough default setting of hardware registers.

Description: Default setting of the following registers is missing.

- PWRUP_TIME, IDLE_PWRDWN, MSB_STRETCH, and HALF_LSB field of SAR_CTL register
- ANA_CAL, DIG_CAL, ANA_CAL_ALT, and DIG_CAL_ALT registers
- DBG_FREEZE_NE field of PASS_CTL register

These registers should be explicitly set to default values instead of the initial values of the hardware.

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-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.

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.

Version history

8.4 Module SW-Version 1.6

T2MC-59613 - [ADC] Change the ADC user guide to include improvement of hardware trigger output of GPT, PWM, and OCU drivers

Title: [ADC] Change the ADC user guide to include improvement of hardware trigger output of GPT, PWM, and OCU drivers

Description: ADC can be triggered by the TCPWM peripheral.

TCPWM peripheral is controlled by GPT, PWM, and OCU drivers.

Configuration parameter names of the ADC trigger setting are changed due to improvement of hardware trigger output of GPT, PWM, and OCU drivers.

Therefore, the ADC user guide needs to be modified.

The changed configuration parameters are the following:

- GptHardwareTriggeredAdc -> GptHwTriggerOutputLine
 - PwmHardwareTriggeredAdc -> PwmHwTriggerOutputLine
 - PwmAdcTriggerFactor -> PwmHwTriggerOutputFactor
 - PwmAdcTriggerDefault -> PwmHwTriggerOutputDefaultTime
 - OcuHardwareTriggeredAdc -> OcuHwTriggerOutputLine
-

T2MC-59572 - [ADC] Execution condition of Adc_EnableHardwareTrigger is incorrectly described in the ADC user guide

Title: [ADC] Execution condition of Adc_EnableHardwareTrigger is incorrectly described in the ADC user guide

Description: Following is a description in the ADC user guide.

Adc_EnableHardwareTrigger() is not allowed on an ongoing group. However, it can be called in ADC_STREAM_COMPLETED when the channel group is configured for streaming access and linear buffer mode.

However, Adc_EnableHardwareTrigger cannot be called with a streaming linear conversion channel group when the group status is ADC_STREAM_COMPLETED. Therefore, the latter description needs to be removed.

T2MC-59486 - [ADC] Multiplicity definition is missing in the *arxml* and *xdm* files

Title: [ADC] Multiplicity definition is missing in the *arxml* and *xdm* files

Description: The following warning messages are detected by the new AMDC tool version 1.0.17. They should be corrected.

- A container has no 'ECUC-MULTIPLICITY-CONFIGURATION-CLASS' elements. (warning of rule A205)
The following containers are modified:
 - Container AdcDemEventParameterRefs
 - Container AdcHwUnit
 - Container AdcChannel
 - Container AdcGroup
 - Container AdcPowerStateConfig
 - A parameter has no 'ECUC-MULTIPLICITY-CONFIGURATION-CLASS' elements. (warning of rule A205)
The following parameter is modified:
-

Version history

-
- Parameter AdcIncludeFile
-

T2MC-59573 - [ADC] Safety mechanism for buffer used by the ADC module is not described in detail in the safety manual

Title: [ADC] Safety mechanism for buffer used by the ADC module is not described in detail in the safety manual

Description: ADC environment needs to assure correctness of buffer size and layout from the safety point of view.

There is no description regarding the above point in any manuals. It should be described in the safety manual.

8.5 Module SW-Version 1.7

T2MC-66687 - [ADC] Support of EPASS_phase2 hardware and port selection feature

Title: [ADC] Support of EPASS_phase2 hardware and port selection feature

Description: The ADC driver supports EPASS_pahse2 hardware and supports the port selection feature.

The ADC driver adds the following configuration parameters.

- AdcSarMux1ConnectToAdc0
 - AdcSarMux1DiagnoseEnable
 - AdcSarMux1DiagnosticReference
 - AdcSarMux2ConnectToAdc0
 - AdcSarMux2DiagnoseEnable
 - AdcSarMux2DiagnosticReference
 - AdcSarMux3ConnectToAdc0
 - AdcSarMux3DiagnoseEnable
 - AdcSarMux3DiagnosticReference
-

T2MC-77594 - Support IAR compiler

Title: Support IAR compiler

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

8.6 Module SW-Version 1.8

T2MC-91242 - [ADC] Adc.xdm is inconsistent with Adc.arxml

Title: [ADC] Adc.xdm is inconsistent with Adc.arxml

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

Adc.xdm:

- UUID is missing in the TS_T40D13M1I0R0 and ADC_EcuParameterDefinition parameters.
 - POSTBUILDVARIANTMULTIPLICITY, POSTBUILDVARIANTVALUE, or both are missing in some parameters.
-

Version history

Adc.xml:

- ORIGIN is wrong in the AdcGroupReplacement parameter.
- DESC of AdcReadGroupApi is different from *Adc.xdm*.
- POST-BUILD-VARIANT-MULTIPLICITY is wrong in the AdcHwUnit, AdcChannel, and AdcGroup parameters.

Also, there are some minor inconsistencies in the description of the AdcUseAlternateCalibration parameter.

8.7 Module SW-Version 1.9

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.

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.

Version history

T2MC-96771 - Some memory placement restrictions are not mentioned in the user guide

Title: Some memory placement restrictions are not mentioned in the user guide

Description: Modules that work with other bus masters such as DW should have restrictions on memory placement.

However, there is insufficient information on these memory restrictions in the user guide. Add the restriction that tightly coupled memories (TCMs) do not support DMA to the user guide.

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.

T2MC-97128 - Unnecessary exclusive control for a process

Title: Unnecessary exclusive control for a process

Description: Some modules have exclusive control in the section where only variables and registers are written atomically. Exclusive control should not be performed for a process that is clearly not affected by interference.

8.8 Module SW-Version 1.10

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.

Version history

8.9 Module SW-Version 1.11

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.

8.10 Module SW-Version 1.12

T2MC-170549 - [ADC] Interrupt status flag cleared at the end of interrupt processing

Title: [ADC] Interrupt status flag cleared at the end of interrupt processing

Description: The ADC interrupt function clears the interrupt flag after performing interrupt processing. So, if an interrupt of the same factor occurs during interrupt processing, it is not processed. The interrupt function should clear the flag at its beginning to allow handling of factors that occur during ADC interrupts. However, ADC module does not stack when interrupt factor occurs during interrupt processing.

If the same interrupt factor occurs during interrupt processing, it will be missed and the following may occur:

- SW trigger source, continuous conversion, and single access mode:

The result buffer is not updated in the later group conversion, or it is not updated in the preceding group conversion. It depends on the interrupt timing. In addition, one of the notifications is not called (if notification is enabled).

- HW trigger source, one-shot conversion, and single or streaming access mode:

The result buffer is not updated in the later group conversion, or it is not updated in the preceding group conversion. It depends on the interrupt timing. In addition, one of the notifications is not called (if notification is enabled).

Work around:

The trigger cycle must be longer than the interruption time*.

The group conversion time must be longer than the interruption time*.

The DMA transfer time must be longer than the interruption time* (if DMA is enabled).

(Trigger cycle + Group conversion time + DMA transfer time > Interruption time*)

[Note] * Interruption time includes the notification time (if notification is enabled).

T2MC-170774 - [ADC] Need to guarantee the order of register settings between relevant peripherals for robustness

Title: [ADC] Need to guarantee the order of register settings between relevant peripherals for robustness

Description: If a driver controls different peripherals that have different bridges and buffers, the order of register settings must be guaranteed.

It is also necessary to guarantee the order of CPU instruction and peripheral operations.

DW control and some CPU instructions meet the above conditions regarding the ADC driver.

Therefore, it should be added to the register read back process.

Version history

T2MC-170538 - [ADC] Unused structure member

Title: [ADC] Unused structure member

Description: Unused structure member has been found inside MCAL code.

The following structure member is not used:

- ChannelConfigsNumber in Adc_ConfigType

This would not affect any functions and its behavior. However, the unused structure member should be removed because it is redundant.

T2MC-170664 - Limitation on notification is missing in user guide

Title: Limitation on notification is missing in user guide

Description: There is a possibility that notifications are performed (under a particular condition) even if notification is disabled. This unexpected behavior would hardly ever occur; however, it should be described in the user guide with a solution to avoid it.

[Conditions]

- Notification is enabled in advance before the notification is called.
- Notification is changed to be disabled in a few cycles just before the notification is called.

[Workaround]

Notification is disabled in advance before running the service.

8.11 Module SW-Version 1.13

T2MC-178672 - [ADC] To align SW construction with ADCEXMC, change the internal construction of the `Adc_Init()` function.

Title: [ADC] To align SW construction with ADCEXMC, change the internal construction of the `Adc_Init()` function.

Description: Some of code access to the MMIO register has been moved to another function. This change comes from the modification of the ADCEXMC module. The reason of code change is to keep easy maintenance between ADC and ADCEXMC in future. This code change does not affect software behavior and interface.

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.

8.12 Module SW-Version 1.14

T2MC-183985 - [ADC] Support TRAVEO™ T2G-C-2D-6M DDR

Title: [ADC] Support TRAVEO™ T2G-C-2D-6M DDR

Description: AUTOSAR MCAL supports the TRAVEO™ T2G-C-2D-6M DDR.

A part of trigger connections is missing in the device as follows:

- Input trigger: `PASS_0_TR_SAR_CH_IN_m` (m: 28 to 31)
 - Output trigger: `PASS_0_TR_SAR_CH_RANGEVIO_n` (n: 28 to 31)
-

Version history

However, the configuration phase allows selecting missing triggers even though they do not exist. The missing trigger connections must be checked and an error must be issued if they are selected during the configuration phase.

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 2022-07-05

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

002-23354 Rev. *L

IMPORTANT NOTICE

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

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

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.